Note for Alan's Class

Zhangzhang Si[1], Haifeng Gong[1,2], Song-Chun Zhu[1,2], and Ying Nian Wu[1]

[1] Department of Statistics, University of California, Los Angeles
[2] Lotus Hill Research Institute, Ezhou, China
{zzsi, hfgong, sczhu, ywu}@stat.ucla.edu

February 8, 2009

1 Active basis model: an algorithmic tour

The active basis model is a natural generalization of the wavelet regression model. In this section, we first explain the background and motivation for the active basis model. Then we work through a series of variable selection algorithms for wavelet regression, where the active basis model emerges naturally.

1.1 From wavelet regression to active basis

1.1.1 $p > n$ regression and variable section

Wavelets have proven to be immensely useful for signal analysis and representation [7]. Various dictionaries of wavelets have been designed for different types of signals or function spaces [3, 13]. Two key factors underlying the successes of wavelets are the sparsity of the representation and the efficiency of the analysis. Specifically, a signal can typically be represented by a linear superposition of a small number of wavelet elements selected from an appropriate dictionary. The selection can be accomplished by efficient algorithms such as matching pursuit [10] and basis pursuit [4].

From a linear regression perspective, a signal can be considered a response variable, and the wavelet elements in the dictionary can be considered the predictor variables or regressors. The number of elements in a dictionary can often be much greater than the dimensionality of the signal, so this is the so-called "$p > n$" problem. The selection of the wavelet elements is the variable selection problem in linear regression. The matching pursuit algorithm [10] is the forward selection method, and the basis pursuit [4] is the lasso method [14].

1.1.2 Gabor wavelets and simple V1 cells

Interestingly, wavelet sparse coding also appears to be employed by the biological visual system for representing natural images. By assuming the sparsity of the linear representation, Olshausen and Field [11] were able to learn from natural images a dictionary of localized, elongate, and oriented basis functions that resemble the Gabor wavelets. Similar wavelets were also obtained by independent component analysis of natural images [1]. From a linear regression perspective,

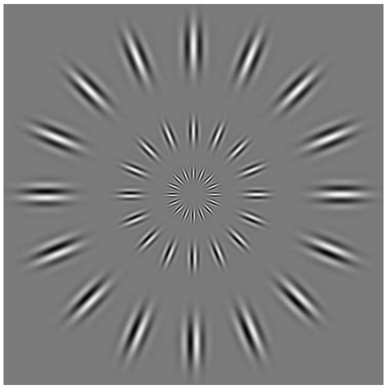

Figure 1: A collection of Gabor wavelets at different locations, orientations, and scales. Each Gabor wavelet element is a sine or cosine wave multiplied by an elongate and oriented Gaussian function. The wave propagates along the shorter axis of the Gaussian function.

Olshausen and Field essentially asked the following question: Given a sample of response vectors (i.e., natural images), can we find a dictionary of predictor vectors or regressors (i.e., basis functions or basis elements), so that each response vector can be represented as a linear combination of a small number of regressors selected from the dictionary? Of course, for different response vectors, different sets of regressors may be selected from the dictionary.

Figure (1) displays a collection of Gabor wavelet elements at different locations, orientations and scales. These are sine and cosine waves multiplied by elongate and oriented Gaussian functions, where the waves propagate along the shorter axes of the Gaussian functions. Such Gabor wavelets have been proposed as mathematical models for the receptive fields of the simple cells of the primary visual cortex or V1 [6].

The dictionary of all the Gabor wavelet elements can be very large, because at each pixel of the image domain, there can be many Gabor wavelet elements at different scales and orientations. According to Olshausen and Field [11], the biological visual system represents a natural image by a linear superposition of a small number of Gabor wavelet elements selected from such a dictionary.

1.1.3 From generic classes to specific categories

Wavelets are designed for generic function classes or learned from generic ensembles such as natural images, under the generic principle of sparsity. While such generality offers enormous scope for the applicability of wavelets, sparsity alone is clearly inadequate for modeling specific patterns. Recently, we have developed an active basis model for images of various object classes [15]. The model is a natural consequence of seeking a common wavelet representation simultaneously for multiple training images from the same object category.

Figure (2) illustrates the basic idea. In the first row, there are 8 images of deers. The images are of the same size of 122 × 120 pixels. The deers appear at the same location, scale, and pose in these

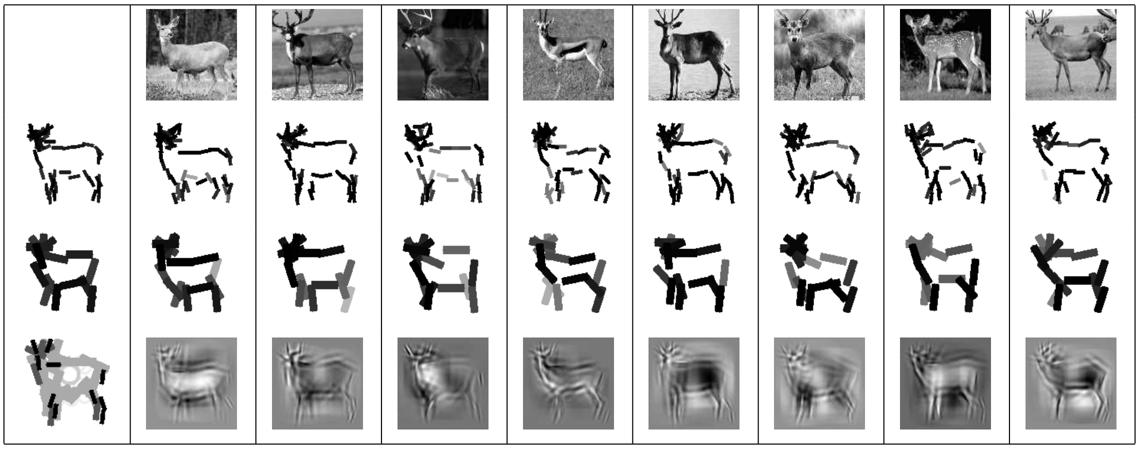

Figure 2: Active basis templates. Each Gabor wavelet element is illustrated by a bar of the same length and at the same location and orientation as the corresponding element. The first row displays the training images. The second row displays the templates composed of Gabor wavelet elements at a fixed scale, where the first template is the common deformable template, and the other templates are deformed templates for coding the corresponding images. The third row displays the templates composed of Gabor wavelet elements at a larger scale than that in the second row. In the last row, the template is composed of Gabor wavelets at multiple scales, where larger elements are illustrated by bars of lighter shades. The rest of the images are reconstructed by linear superpositions of the wavelet elements of the deformed templates.

images. For these very similar images, we want to seek a common wavelet representation, instead of coding each image individually. Specifically, we want these images to be represented by similar sets of wavelet elements, with similar coefficients. We can achieve this by selecting a common set of wavelet elements, while allowing these wavelet elements to locally perturb their locations and orientations before they are linearly combined to code each individual image. The perturbations are introduced to account for shape deformations in the deers. The linear basis formed by such perturbable wavelet elements is called an active basis.

This is illustrated by the second and third rows of Figure (2). In each row, the first plot displays the common set of Gabor wavelet elements selected from a dictionary. The dictionary consists of Gabor wavelets at all the locations and orientations, but at a fixed scale. Each Gabor wavelet element is symbolically illustrated by a bar at the same location and orientation and with the same length as the corresponding Gabor wavelet. So the active basis formed by the selected Gabor wavelet elements can be interpreted as a template, as if each element is a stroke for sketching the template. The templates in the second and third rows are learned using dictionaries of Gabor wavelets at two different scales, with the scale of the third row about twice as large as the scale of the second row. The number of Gabor wavelet elements of the template in the second row is 50, while the number of elements of the template in the third row is 15. Currently we treat this number as a tuning parameter, although they can be determined in a more principled way.

Within each of the second and third rows, and for each training image, we plot the Gabor wavelet elements that are actually used to represent the corresponding image. These elements are perturbed versions of the corresponding elements in the first column. So the templates in the first column are deformable templates, and the templates in the remaining columns are deformed templates. Thus the goal of seeking a common wavelet representation for images from the same object category leads us to formulate the active basis, which is a deformable template for the images from the object category.

In the last row of Figure (2), the common template is learned by selecting from a dictionary that consists of Gabor wavelet elements at multiple scales instead of a fixed scale. In addition to Gabor wavelet elements, we also include the center-surround difference of Gaussian wavelet elements in the dictionary. Such isotropic wavelet elements are of large scales, and they mainly capture the regional contrasts in the images. In the template in the last row, larger Gabor wavelet elements are illustrated by bars of lighter shades. The difference of Gaussian elements are illustrated by circles. The remaining images are reconstructed by such multi-scale wavelet representations, where each image is a linear superposition of the Gabor and difference of Gaussian wavelet elements of the corresponding deformed templates.

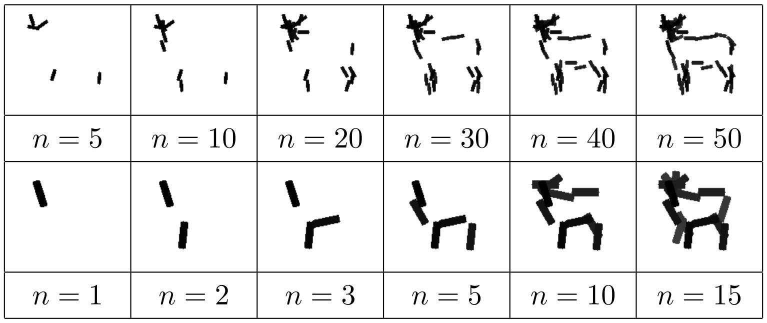

Figure 3: Shared sketch process for learning the active basis templates at two different scales.

The active basis can be learned by the shared sketch algorithm that we recently developed [15]. This algorithm can be considered a paralleled version of the matching pursuit algorithm [10]. It can also be considered a modification of the projection pursuit algorithm [8]. The algorithm selects the wavelet elements sequentially from the dictionary. Each time when an element is selected, it is shared by all the training images in the sense that a perturbed version of this element is included in the linear representation of each image. Figure (3) illustrates the shared sketch process for obtaining the templates displayed in the second and third rows of Figure (2).

While selecting the wavelet elements of the active basis, we also estimate the distributions of their coefficients from the training images. This gives us a statistical model for the images. After learning this model, we can then use it to recognize the same type of objects in testing images. In machine learning and computer vision literature, detecting or classifying objects using the learned model is often called inference. The inference algorithm is often a part of the learning algorithm. For active basis model, both the learning and inference can be formulated as maximum likelihood estimation problems.

1.1.4 Local maximum pooling and complex V1 cells

Besides wavelet sparse coding, another inspiration to the active basis model comes from neuroscience. Riesenhuber and Poggio [12] observed that the complex cells of the primary visual cortex or V1 appear to perform local maximum pooling of the responses from simple cells. From the perspective of active basis model, this corresponds to estimating the perturbations of the wavelet elements of the active basis template, so that the template is deformed to match the observed image. Therefore, if we are to believe Olshausen and Field's theory on wavelet sparse coding [11] and Riesenhuber and Poggio's theory on local maximum pooling, then the active basis model seems to be a very natural logical consequence.

In the following subsections, we shall describe wavelet sparse coding, the active basis model, and the learning and inference algorithms in detail.

1.2 An overcomplete dictionary of Gabor wavelets

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. The Gaussian function is elongate 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 domain, so that it has a well defined length and width.

We then translate, rotate, and dilate $G(x_1, x_2)$ to obtain a general form of the Gabor wavelets:

$$B_{x_1,x_2,s,\alpha}(x_1', x_2') = G(\tilde{x}_1/s, \tilde{x}_2/s)/s^2,$$

where

$$\tilde{x_1} = (x_1' - x_1)\cos\alpha - (x_2' - x_2)\sin\alpha,$$
$$\tilde{x_2} = (x_1' - x_1)\sin\alpha + (x_2' - x_2)\cos\alpha.$$

Writing $x = (x_1, x_2)$, each $B_{x,s,\alpha}$ is a localized function, where $x = (x_1, x_2)$ is the central location, s is the scale parameter, and α is the orientation. The frequency of the wave propagation in $B_{x,s,\alpha}$ is $\omega = 1/s$. $B_{x,s,\alpha} = (B_{x,s,\alpha,0}, B_{x,s,\alpha,1})$, 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 each $B_{x,s,\alpha}$, the pair $B_{x,s,\alpha,0}$ and $B_{x,s,\alpha,1}$ are orthogonal to each other.

The dictionary of Gabor wavelets is

$$\Omega = \{B_{x,s,\alpha}, \forall (x, s, \alpha)\}.$$

We can discretize the orientation so that $\alpha \in \{o\pi/A, o = 0, ..., O-1\}$, i.e., O equally spaced orientations (the default value of O is 15 in our experiments). In this article, we mostly learn the active basis template at a fixed scale s. The dictionary Ω is called "overcomplete" because the number of wavelet elements in Ω is larger than the number of pixels in the image domain, since at each pixel, there can be many wavelet elements tuned to different orientations and scales.

For an image $\mathbf{I}(x)$, with $x \in D$, where D is a set of pixels, such as a rectangular grid, 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

$$\langle \mathbf{I}, B_{x,s,\alpha,\eta} \rangle = \sum_{x'} \mathbf{I}(x') B_{x,s,\alpha,\eta}(x').$$

The summation is over the finite support of $B_{x,s,\alpha,\eta}$. We write $\langle \mathbf{I}, B_{x,s,\alpha} \rangle = (\langle \mathbf{I}, B_{x,s,\alpha,0} \rangle, \langle \mathbf{I}, B_{x,y,s,\alpha,1} \rangle)$. The local energy is

$$|\langle \mathbf{I}, B_{x,s,\alpha} \rangle|^2 = \langle \mathbf{I}, B_{x,s,\alpha,0} \rangle^2 + \langle \mathbf{I}, B_{x,s,\alpha,1} \rangle^2.$$

$|\langle \mathbf{I}, B_{x,s,\alpha} \rangle|^2$ is the local spectrum or the magnitude of the local wave in image $\mathbf{I}$ at (x, s, α).

Let

$$\sigma_s^2 = \frac{1}{|D|O} \sum_{\alpha} \sum_{x \in D} |\langle \mathbf{I}, B_{x,s,\alpha} \rangle|^2,$$

where $|D|$ is the number of pixels in $\mathbf{I}$, and O is the total number of orientations. For each image $\mathbf{I}$, we normalize it to $\mathbf{I} \leftarrow \mathbf{I}/\sigma_s$, so that different images are comparable.

1.3 Matching pursuit algorithm

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

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

where $(B_{x_i,s,\alpha_i}, i = 1, ..., n) \subset \Omega$ is a set of Gabor wavelet elements selected from the dictionary Ω, c_i is the coefficient, and U is the unexplained residual image. Recall that each B_{x_i,s,α_i} is a pair of Gabor cosine and sine components. So $B_{x_i,s,\alpha_i} = (B_{x_i,s,\alpha_i,0}, B_{x_i,s,\alpha_i,1})$ and $c_i = (c_{i,0}, c_{i,1})$, and $c_i B_{x_i,s,\alpha_i} = c_{i,0} B_{x_i,s,\alpha_i,0} + c_{i,1} B_{x_i,s,\alpha_i,1}$.

In the representation (1), n is often assumed to be small, e.g., $n = 50$. So the representation (1) is called sparse representation or sparse coding. This representation translates a raw intensity image with a huge number of pixels into a sketch with only a small number of strokes represented by $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$. Because of the sparsity, $\mathbf{B}$ captures the most visually meaningful elements in the image. The set of wavelet elements $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$ can be selected from Ω by the matching pursuit algorithm [10], which seeks to minimize $\|\mathbf{I} - \sum_{i=1}^{n} c_i B_{x_i,s,\alpha_i}\|^2$ by a greedy scheme.

Algorithm 0: Matching pursuit algorithm

```
0 Let $i \leftarrow 0$. $U \leftarrow \mathbf{I}$.
1 $i \leftarrow i + 1$. Let $(x_i, \alpha_i) = \arg\max_{x,\alpha} |\langle U, B_{x,s,\alpha}\rangle|^2$.
2 Let $c_i = \langle U, B_{x_i,s,\alpha_i}\rangle$. Let $U \leftarrow U - c_i B_{x_i,s,\alpha_i}$.
3 Stop if $i = n$, else go back to 1.
```

In the above algorithm, it is possible that a wavelet element is selected more than once, but this is extremely rare for real images. As to the choice of n or the stopping criterion, we can stop the algorithm if $|c_i|$ is below a threshold.

Readers who are familiar with the so-called "large p and small n" problem in linear regression may have recognized that wavelet sparse coding is a special case of this problem, where $\mathbf{I}$ is the response vector, and each $B_{x,s,\alpha} \in \Omega$ is a predictor vector. The matching pursuit algorithm is actually the forward selection procedure for variable selection.

The forward selection algorithm in general can be too greedy. But for image representation, each Gabor wavelet element only explains away a small part of the image data, and we usually pursue the elements at a fixed scale, so such a forward selection procedure is not very greedy.

1.4 Matching pursuit for multiple images

Let $\{\mathbf{I}_m, m = 1, ..., M\}$ be a set of training images defined on a common rectangle lattice D, and let us suppose that these images come from the same object category, where the objects appear at the same pose, location, and scale in these images. We can model these images by a common set of Gabor wavelet elements,

$$\mathbf{I}_m = \sum_{i=1}^{n} c_{m,i} B_{x_i,s,\alpha_i} + U_m, \quad m = 1, ..., M. \tag{2}$$

$\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$ can be considered a common template for these training images.

We can select these elements by applying the matching pursuit algorithm on these multiple images simultaneously, which seeks to minimize $\sum_{m=1}^{M} \|\mathbf{I} - \sum_{i=1}^{n} c_i B_{x_i,s,\alpha_i}\|^2$ by a greedy scheme.

Algorithm 1: Matching pursuit on multiple images

```
0 Initialize $i \leftarrow 0$. For $m = 1, ..., M$, initialize $U_m \leftarrow \mathbf{I}_m$.
1 $i \leftarrow i + 1$. Select
      $(x_i, \alpha_i) = \arg\max_{x,\alpha} \sum_{m=1}^{M} |\langle U_m, B_{x,\alpha} \rangle|^2.$
2 For $m = 1, ..., M$, let $c_{m,i} = \langle U_m, B_{x_i,s,\alpha_i} \rangle$, and update $U_m \leftarrow U_m - c_{m,i} B_{x_i,s,\alpha_i}$.
3 Stop if $i = n$, else go back to 1.
```

Algorithm 1 is similar to algorithm 0. The difference is that, in Step 1, (x_i, α_i) is selected by maximizing the sum of the squared responses.

1.5 Active basis and local maximum pooling

The objects in the training images share similar shapes, but there can still be considerable variations in their shapes. In order to account for the shape deformations, we introduce the perturbations to the common template, and the model becomes

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

Again, $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$ can be considered a common template for the training images, but this time, this template is deformable. Specifically, 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. $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$ can be considered the deformed template for coding image $\mathbf{I}_m$. We call the basis formed by $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$ the active basis, and we call $(\Delta x_{m,i}, \Delta\alpha_{m,i}, i = 1, ..., n)$ the activities or perturbations of the basis elements for image m.

Figure (2) illustrates three examples of active basis templates. In the second and third rows, the templates in the first column are $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$. The scale parameter s in the second row is smaller than the s in the third row. For each row, the templates in the remain columns are the deformed templates $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$. The template in the last row should be more precisely represented by $\mathbf{B} = (B_{x_i,s_i,\alpha_i}, i = 1, ..., n)$, where each element has its own s_i automatically selected together with (x_i, α_i). In this article, we focus on the situation where we fix s (default length of the wavelet element is 17 pixels).

For the activity or perturbation of a wavelet element $B_{x,s,\alpha}$, we assume that $\Delta x = (\Delta x_1, \Delta x_2)$, where $\Delta x_1 = d \sin\alpha$, $\Delta x_2 = d \cos\alpha$, with $d \in [-b_1, b_1]$. We also assume $\Delta\alpha \in [-b_2, b_2]$. b_1 and b_2 are the bounds for the allowed displacements in location and orientation (default values: $b_1 = 6$ pixels, and $b_2 = \pi/15$). We define

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

be the set of all possible activities for a basis element tuned to orientation α.

We can continue to apply the matching pursuit algorithm to the multiple training images, the only difference is that we add a local maximum pooling operation in Steps 1 and 2. The following algorithm is a greedy procedure to minimize the least squares criterion

$$\sum_{m=1}^{M} \|\mathbf{I}_m - \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i}, s, \alpha_i+\Delta\alpha_{m,i}}\|^2. \tag{4}$$

Algorithm 2: Matching pursuit with local maximum pooling

```
0 Initialize $i \leftarrow 0$. For $m = 1, ..., M$, initialize $U_m \leftarrow \mathbf{I}_m$.

1 $i \leftarrow i + 1$. Select
    $$(x_i, \alpha_i) = \arg\max_{x,\alpha} \sum_{m=1}^{M} \max_{(\Delta x, \Delta\alpha) \in A(\alpha)} |\langle U_m, B_{x+\Delta x, s, \alpha+\Delta\alpha}\rangle|^2.$$

2 For $m = 1, ..., M$, retrieve
    $$(\Delta x_{m,i}, \Delta\alpha_{m,i}) = \arg \max_{(\Delta x, \Delta\alpha) \in A(\alpha_i)} |\langle U_m, B_{x_i+\Delta x, s, \alpha_i+\Delta\alpha}\rangle|^2.$$
  Let $c_{m,i} \leftarrow \langle U_m, B_{x_i+\Delta x_{m,i}, s, \alpha_i+\Delta\alpha_{m,i}}\rangle$, and update $U_m \leftarrow U_m - c_{m,i} B_{x_i+\Delta x_{m,i}, s, \alpha_i+\Delta\alpha_{m,i}}$.

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

Algorithm 2 is similar to Algorithm 1. The difference is that we add an extra local maximization operation in Step 1: $\max_{(\Delta x, \Delta\alpha) \in A(\alpha)} |\langle U_m, B_{x+\Delta x, s, \alpha+\Delta\alpha}\rangle|^2$. With (x_i, α_i) selected in Step 1, Step 2 retrieves the corresponding maximal $(\Delta x, \Delta\alpha)$ for each image.

We can rewrite Algorithm 2 by defining $R_m(x, \alpha) = \langle U_m, B_{x,s,\alpha}\rangle$. Then instead of updating the residual image U_m in Step 2, we can update the responses $R_m(x, \alpha)$.

Algorithm 2.1: Matching pursuit with local maximum pooling

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

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

2 For $m = 1, ..., M$, 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
    $$R_m(x, \alpha) \leftarrow R_m(x, \alpha) - c_{m,i} \langle B_{x,s,\alpha}, B_{x_i+\Delta x_{m,i}, s, \alpha_i+\Delta\alpha_{m,i}}\rangle.$$

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

1.6 Shared sketch algorithm

Finally, we come to the shared sketch algorithm that we actually used in the experiments in this paper. The algorithm involves two modifications to Algorithm 2.1.

Algorithm 3: Shared sketch algorithm

```
0 Initialize $i \leftarrow 0$. For $m = 1, ..., M$, initialize $R_m(x, \alpha) \leftarrow \langle \mathbf{I}_m, B_{x,s,\alpha} \rangle$ for all $(x, \alpha)$.
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 $m = 1, ..., M$, 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
      $\mathrm{corr}(B_{x,s,\alpha}, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}) > 0.$
3 Stop if $i = n$, else go back to 1.
```

The two modifications are:

(1) In Step 1, we change $|R_m(x + \Delta x, \alpha + \Delta\alpha)|^2$ to $h(|R_m(x + \Delta x, \alpha + \Delta\alpha)|^2)$ where $h()$ is a sigmoid function, which increases from 0 to a saturation level ξ (default: $\xi = 6$),

$$h(r) = \xi \left[\frac{2}{1 + e^{-2r/\xi}} - 1 \right]. \tag{5}$$

Intuitively, $\sum_{m=1}^{M} \max_{(\Delta x, \Delta\alpha) \in A(\alpha)} h(|R_m(x + \Delta x, \alpha + \Delta\alpha)|^2)$ can be considered a vote from all the images for the location and orientation (x, α), where each image contributes $\max_{(\Delta x, \Delta\alpha) \in A(\alpha)} h(|R_m(x + \Delta x, \alpha + \Delta\alpha)|^2)$. The sigmoid transformation prevents a small number of images from contributing very large values. As a result, the selection of (x, α) is a more "democratic" choice, and the selected element tends to sketch the edges shared by all the training images. In the next section, we shall formally justify the use of sigmoid transformation by a statistical model.

(2) In Step 3, we update $R_m(x, \alpha) \leftarrow 0$ if $B_{x,s,\alpha}$ is not orthogonal to $B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}$. That is, we enforce the orthogonality of the basis $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$ for each training image m. Our experience with matching pursuit is that it usually selects elements that have little overlap with each other. So for computational convenience, we simply enforce that the selected elements are orthogonal to each other. For two Gabor wavelets B_1 and B_2, we define their correlation as $\mathrm{corr}(B_1, B_2) = \sum_{\eta_1=0}^{1} \sum_{\eta_2=0}^{1} |\langle B_{1,\eta_1}, B_{2,\eta_2} \rangle|^2$, i.e., the sum of squared inner products between the sine and cosine components of B_1 and B_2. In practical implementation, we allow small correlation between selected elements, i.e., we update $R_m(x, \alpha) \leftarrow 0$ if $\mathrm{corr}(B_{x,s,\alpha}, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}) > \epsilon$ (the default value of $\epsilon = .1$).

1.7 Statistical modeling of images

In this subsection, we develop a statistical model for $\mathbf{I}_m$. A statistical model is not only important for justifying Algorithm 3 for learning the active basis template, it also enables us to use the learned template to recognize the objects in testing images, because we can use the log-likelihood to score the matching between the learned template and image data.

The statistical model is based on the decomposition $\mathbf{I}_m = \sum_{i=1}^m c_{m,i} B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}} + U_m$, where $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$ is orthogonal, and $c_{m,i} = \langle \mathbf{I}_m, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}} \rangle$, so U_m lives in the subspace that is orthogonal to $\mathbf{B}_m$. In order to specify a statistical model for $\mathbf{I}_m$ given $\mathbf{B}_m$, we only need to specify the distribution of $(c_{m,i}, i = 1, ..., n)$ and the conditional distribution of U_m given $(c_{m,i}, i = 1, ..., n)$.

The least squares criterion (4) that drives Algorithm 2 implicitly assumes that U_m is a white noise distribution, and $c_{m,i}$ follows a diffused prior distribution. This assumption is wrong. Because there can be occasional strong edges in the background, and a white noise U_m cannot account for strong edges. Moreover, the distribution of $c_{m,i}$ should be estimated from training images.

In this work, we choose to estimate the distribution of $c_{m,i}$ from training images by fitting an exponential family model to the sample $\{c_{m,i}, m = 1, ..., M\}$ obtained from the training images, and we assume that the conditional distribution of U_m given $(c_{m,i}, i = 1, ..., n)$ is the same as the corresponding conditional distribution in natural images. Such a conditional distribution can account for occasional strong edges in the background, and it is the use of such a conditional distribution of U_m as well as the exponential family model for $c_{m,i}$ that leads to the sigmoid transformation in Algorithm 3. Intuitively, a large response $|R_m(x + \Delta x, \alpha + \Delta\alpha)|^2$ indicates that there can be an edge at $(x + \Delta x, \alpha + \Delta\alpha)$. Because an edge can also be accounted for by the distribution of U_m in natural images, a large response should not be taken at its face value for selecting the basis elements. Instead, it should be discounted by a transformation such as $h()$ in Algorithm 3.

1.8 Density substitution and projection pursuit

We adopt the density substitution scheme of projection pursuit [8] to construct a statistical model. We start from a reference distribution $q(\mathbf{I})$. In this article, we assume that $q(\mathbf{I})$ is the distribution of all the natural images. We do not need to know $q(\mathbf{I})$ explicitly beyond the marginal distribution $q(c)$ of $c = \langle \mathbf{I}, B_{x,s,\alpha} \rangle$ under $q(\mathbf{I})$. Because $q(\mathbf{I})$ is stationary and isotropic, $q(c)$ is the same for different (x, α). $q(c)$ is a heavy tail distribution because there are edges in natural images. $q(c)$ can be estimated from natural images by pooling a histogram of $\{\langle \mathbf{I}, B_{x,s,\alpha} \rangle, \forall \mathbf{I}, \forall (x, \alpha)\}$ where $\{\mathbf{I}\}$ is a sample of natural images.

Given $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$, we modify the reference distribution $q(\mathbf{I}_m)$ to a new distribution $p(\mathbf{I}_m)$ by changing the distributions of $c_{m,i}$. Let $p_i(c)$ be the distribution of $c_{m,i}$ pooled from $\{c_{m,i}, m = 1, ..., M\}$, which are obtained from training images $\{\mathbf{I}_m, m = 1, ..., M\}$.

Then we change the distribution of $c_{m,i}$ from $q(c)$ to $p_i(c)$, for each $i = 1, ..., n$, while keeping the conditional distribution of U_m given $(c_{m,i}, i = 1, ..., n)$ unchanged. This leads us to

$$p(\mathbf{I}_m \mid \mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)) = q(\mathbf{I}_m) \prod_{i=1}^{n} \frac{p_i(c_{m,i})}{q(c_{m,i})}, \tag{6}$$

where we assume that $(c_{m,i}, i = 1, ..., n)$ are independent under both $p(\mathbf{I}_m)$ and $q(\mathbf{I}_m \mid \mathbf{B}_m)$, for orthogonal $\mathbf{B}_m$. The conditional distributions of U_m given $(c_{m,i}, i = 1, ..., n)$ under $p(\mathbf{I}_m \mid \mathbf{B}_m)$ and $q(\mathbf{I}_m)$ are canceled out in $p(\mathbf{I}_m \mid \mathbf{B}_m)/q(\mathbf{I}_m)$ because they are the same. The Jacobians are also the same and are canceled out. So $p(\mathbf{I}_m \mid \mathbf{B}_m)/q(\mathbf{I}_m) = \prod_{i=1}^{n} p_i(c_{m,i})/q(c_{m,i})$.

The following are three perspectives to view model (6):

(1) Classification: we may consider $q(\mathbf{I})$ as representing the negative examples, and $\{\mathbf{I}_m\}$ are positive examples. We want to find the basis elements $(B_{x_i,s,\alpha_i}, i = 1, ..., n)$ so that the projections $c_{m,i} = \langle \mathbf{I}_m, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}} \rangle$ for $i = 1, ..., n$ best distinguish the positive examples from the negative examples.

(2) Hypothesis testing: we may consider $q(\mathbf{I})$ as representing the null hypothesis, and the observed histograms of $c_{m,i}, i = 1, ..., n$ are the test statistics that are used to reject the null hypothesis.

(3) Coding: we choose to code $c_{m,i}$ by $p_i(c)$ instead of $q(c)$, while continue to code U_m by the conditional distribution of U_m given $(c_{m,i}, i = 1, ..., n)$ under $q(\mathbf{I})$.

For all the three perspectives, we need to choose B_{x_i,s,α_i} so that there is big contrast between $p_i(c)$ and $q(c)$. The shared sketch process can be considered as sequentially flipping dimensions of $q(\mathbf{I}_m)$ from $q(c)$ to $p_i(c)$ to fit the observed images. It is essentially a projection pursuit procedure, with an additional local maximization step for estimating the activities of the basis elements.

1.9 Exponential tilting and saturation transformation

While $p_i(c)$ can be estimated from $\{c_{m,i}, m = 1, ..., M\}$ by pooling a histogram, we choose to parametrize $p_i(c)$ with a single parameter so that it can be estimated from even a single image.

We assume $p_i(c)$ to be the following exponential family model:

$$p(c; \lambda) = \frac{1}{Z(\lambda)} \exp\{\lambda h(r)\} q(c),$$

where $\lambda > 0$ is the parameter. For $c = (c_0, c_1)$, $r = |c|^2 = c_0^2 + c_1^2$.

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

is the normalizing constant. $h(r)$ is a monotone increasing function. We assume $p_i(c) = p(c; \lambda_i)$, which accounts for the fact that the squared responses $\{|c_{m,i}|^2 = |\langle \mathbf{I}_m, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}} \rangle|^2, m = 1, ..., M\}$ in the positive examples are in general larger than those in natural images, because $B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}$ tends to sketch a local edge segment in each $\mathbf{I}_m$. As mentioned before, $q(c)$ is estimated by pooling a histogram from natural images.

We argue that $h(r)$ should be a saturation transformation in the sense that as $r \to \infty$, $h(r)$ approaches a finite number. The sigmoid transformation in (5) is such a transformation. The reason for such a transformation is as follows. Let $q(r)$ be the distribution of $r = |c|^2 = |\langle \mathbf{I}, B\rangle|^2$ under $q(c)$ where $\mathbf{I} \sim q(\mathbf{I})$. We may implicitly model $q(c)$ as a mixture of $p_{\rm on}(r)$ and $p_{\rm off}(r)$, where $p_{\rm on}$ is the distribution of r when B is on an edge in $\mathbf{I}$, and $p_{\rm off}$ is the distribution of r when B is off an edge in $\mathbf{I}$. Let $q(r) = (1-\rho_0)p_{\rm off}(r)+\rho_0 p_{\rm on}$, where ρ_0 is the proportion of edges in natural images. Similarly, let $p_i(r)$ be the distribution of $r = |c|^2$ under $p_i(c)$. We can model $p_i(r) = (1-\rho_i)p_{\rm off}(r) + \rho_i p_{\rm on}$, where $\rho_i > \rho_0$, i.e., the proportion of edges sketched by the selected basis element is higher than the proportion of edges in natural images. Then, as $r \to \infty$, $p_i(r)/q(r) \to \rho_i/\rho_0$, which is a constant. Therefore, $h(r)$ should saturate as $r \to \infty$.

1.10 Maximum likelihood learning and pursuit index

Now we can justify the shared sketch algorithm as a greedy scheme for maximizing the log-likelihood. With parametrization (7) for the statistical model (6), the log-likelihood is

$$\sum_{m=1}^{M}\sum_{i=1}^{n} \log \frac{p_i(c_{m,i})}{q(c_{m,i})} = \sum_{i=1}^{n} \lambda_i \sum_{m=1}^{M} h(|\langle \mathbf{I}_m, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}\rangle|^2) - M \log Z(\lambda_i). \tag{7}$$

We want to estimate the locations and orientations of the elements of the active basis, $(x_i, \alpha_i, i = 1, ..., n)$, the activities of these elements, $(\Delta x_{m,i}, \Delta\alpha_{m,i}, i = 1, ..., n)$, and the weights $(\lambda_i, i = 1, ..., n)$, by maximizing the log-likelihood (7), subject to the constraints that $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$ is orthogonal for each m.

First, we consider the problem of estimating the weight λ_i given $\mathbf{B}_m$ and $c_{m,i}$. To maximize the log-likelihood (7) over λ_i, we only need to maximize

$$l_i(\lambda_i) = \lambda_i \sum_{m=1}^{M} h(|\langle \mathbf{I}_m, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}\rangle|^2) - M \log Z(\lambda_i).$$

By setting $l_i'(\lambda_i) = 0$, we get the well-known form of the estimating equation for exponential family model:

$$\mu(\lambda_i) = \frac{1}{M}\sum_{m=1}^{M} h(c_{m,i}|^2), \tag{8}$$

where the mean parameter $\mu(\lambda)$ of the exponential family model is

$$\mu(\lambda) = \mathrm{E}_\lambda[h(r)] = \frac{1}{Z(\lambda)} \int h(r) \exp\{\lambda h(r)\} q(r) dr. \tag{9}$$

The estimating equation (8) can be solved easily, because $\mu(\lambda)$ is a one-dimensional function. We can simply store this monotone function over a one-dimensional grid. Then we solve this equation by looking up the stored values, with the help of nearest neighbor linear interpolation for the values between the grid points. For each grid point of λ, $\mu(\lambda)$ can be computed by one-dimensional integration. Thanks to the independence assumption, we only need to deal with such one-dimensional functions, which relieves us from time consuming MCMC computations such as those in our previous work [16].

Next let us consider the problem of selecting (x_i, α_i) and estimating the activity $(\Delta x_{m,i}, \Delta \alpha_{m,i})$ for each image $\mathbf{I}_m$. Let $\hat{\lambda}_i$ be the solution to the estimating equation (9). $l_i(\hat{\lambda}_i)$ is monotone in $\sum_{m=1}^M h(|c_{m,i}|^2)/M$. Therefore, we need to find (x_i, α_i), and $(\Delta x_{m,i}, \Delta \alpha_{m,i})$, by maximizing $\sum_{m=1}^M h(|c_{m,i}|^2)$. This justifies Step 1 of Algorithm 3, where $\sum_{m=1}^M h(|R_m(x + \Delta x, \alpha + \Delta\alpha)|^2)$ serves as the pursuit index.

1.11 SUM-MAX maps for template matching

After learning the active basis model, in particular, the basis elements $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$ and the weights $(\lambda_i, i = 1, ..., n)$, we can use the learned model to find the object in a testing image $\mathbf{I}$. The testing image may not be defined on the same lattice as the training images. For example, the testing image may be larger than the training images. We assume that there is one object in the testing image, but we do not know the location of the object in the testing image. In order to detect the object, we scan the template over the testing image, and at each location x, we can deform the template and match it to the image patch around x. This gives us a log-likelihood score at each location x. Then we can find the maximum likelihood location $\hat{x}$ that achieves the maximum of the log-likelihood score among all the x. After computing $\hat{x}$, we can then retrieve the activities of the elements of the active basis template centered at $\hat{x}$.

Algorithm 4: Object detection by template matching

1 For every x, compute

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

2 Select $\hat{x} = \arg\max_{x,y} l(x)$. For $i = 1, ..., n$, 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 deformed template $(B_{\hat{x}+x_i+\Delta x_i, s, \alpha_i+\Delta\alpha_i}, i = 1, ..., n)$.

Step 1 of the above algorithm can be realized by a computational architecture called sum-max maps.

Algorithm 4.1: sum-max maps

1 For all (x, α), compute $\text{SUM1}(x, \alpha) = h(|\langle \mathbf{I}, B_{x,s,\alpha_i} \rangle|^2)$.

2 For all (x, α), compute $\text{MAX1}(x, \alpha) = \max_{(\Delta x, \Delta\alpha) \in A(\alpha)} \text{SUM1}(x, \alpha)$.

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

$\text{SUM2}(x)$ is $l(x)$ in Algorithm 4.

The local maximization operation in Step 2 of Algorithm 4.1 has been hypothesized as the function of the complex cells of the primary visual cortex [12]. In the context of the active basis model, this operation can be justified as the maximum likelihood estimation of the activities. The shared sketch learning algorithm can also be written in terms of sum-max maps.

Algorithm 4 with the Step 1 implemented by algorithm 4.1 can be considered a simplified special case of dynamic programming [2]. It is also related to belief propagation for fitting deformable templates [5], although it is much simpler than the latter, thanks to the simple independent structure of the model.

The activities $(\Delta x_{m,i}, \Delta \alpha_{m,i}, i = 1, ..., n)$ should be treated as latent variables in the active basis model. However, in both learning and inference algorithms, we treat them as unknown parameters, and we maximize over them instead of integrating them out. According to Little and Rubin [9], maximizing the complete-data likelihood over the latent variables may not lead to valid inference in general. However, in natural images, there is little noise, and the uncertainty in the activities is often very small. So maximizing over the latent variables can be considered a good approximation to integrating out the latent variables.

References

[1] A. Bell, and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, 37, 3327-3338, 1997.

[2] R. Bellman, "Dynamic programming," *Science*, 1966.

[3] E. J. Candes and D. L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," *Curves and Surfaces*, L. L. Schumakeretal. (eds), 1999.

[4] S. Chen, D. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, 20(1), 33-61, 1999.

[5] J. M. Coughlan and S. J. Ferreira, "Finding deformable shapes using loopy belief propagation," *Lecture Notes in Computer Science*, Springer, 2002.

[6] 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.

[7] D.L. Donoho, M. Vetterli, R.A. DeVore, and I. Daubechie, "Data compression and harmonic analysis", *IEEE Transactions on Information Theory*, 6, 2435-2476, 1998.

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

[9] R. J. A. Little and D. B. Rubin, "On jointly estimating parameters and missing data by maximizing the complete data likelihood," *The American Statistician*, 37, 218-220, 1983.

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

[11] 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.

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

[13] E. P. Simoncelli, W. T. Freeman, E. H.Adelson, D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Infomation Theory*, 38, 587-607, 1992.

[14] Tibshirani, R. "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, B*, 58, 267-288, 1996.

[15] Y. N. Wu, Z. Si, C. Fleming, and S. C. Zhu, "Deformable template as active basis," *Proceedings of International Conference of Computer Vision*, 2007.

[16] Z. J. Xu, H. Chen, S. C. Zhu, and J. Luo, "A Composite Template for Human Face Modeling and Sketch," *IEEE Transations on Pattern Analysis and Machine Intelligence*, 30, 955-969, 2008.