

Describe the major findings resulting from these activities

During the past year, we have been focusing on further developing the active basis model for visual patterns. We shall briefly review the model, and then we shall report our major findings.

1 The active basis model

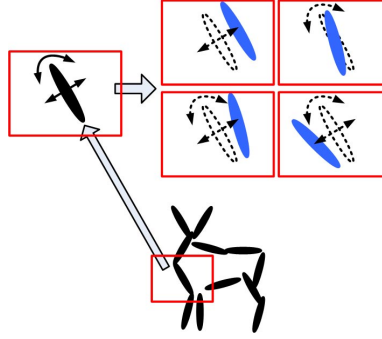


Figure 1: The active basis model. Each basis element is illustrated by a thin ellipsoid. Each element of the template is allowed to perturb its location and orientation to sketch the image.

The active basis model can be written in the following compact form,

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

where $\{\mathbf{I}_m, m = 1, \dots, M\}$ is a set of training images. The objects in these images are assumed to come from the same category and in the same pose, and they appear at the same location, scale and orientation in the training images. Such images are said to be aligned, and the learning in this setting is called supervised learning. Each \mathbf{I}_m is represented by a linear superposition of a set of basis elements that are perturbed versions of $(B_{x_i, s, \alpha_i}, i = 1, \dots, n)$, where each B_{x_i, s, α_i} is a localized, elongated and oriented Gabor wavelet element localized at location x_i , scale s and orientation α_i . For each image \mathbf{I}_m , B_{x_i, s, α_i} is perturbed to $B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}}$ to account for shape deformation, where $\Delta x_{m,i}$ and $\Delta \alpha_{m,i}$ are perturbations or activities in location and orientation respectively. $c_{m,i}$ is the coefficient of the perturbed basis element, and U_m is the unexplained residual image. The set of basis elements $\mathbf{B} = (B_{x_i, s, \alpha_i}, i = 1, \dots, n)$ are selected from a dictionary of Gabor wavelets. \mathbf{B} can be considered a deformable template, and $\mathbf{B}_m = (B_{x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}}, i = 1, \dots, n)$ is the deformed template for representing \mathbf{I}_m . We assume that the scale s is fixed and known.

Figure (1) illustrates the model, where each Gabor wavelet element B_{x_i, s, α_i} is illustrated by a thin ellipsoid, which is allowed to perturb its location and orientation. For statistical modeling, we assume

$(c_{m,i}, i = 1, \dots, n)$ and U_m follow certain non-Gaussian distributions. The distribution of \mathbf{I}_m given \mathbf{B}_m is in the form of an exponential family model.

The learning involves selecting $(B_{x_i, s, \alpha_i}, i = 1, \dots, n)$, or more specifically $((x_i, \alpha_i), i = 1, \dots, n)$, to form the template \mathbf{B} . The learning also involves estimating the parameters in the exponential family model. The learning is based on maximum likelihood. After learning the template, we can use the learned model to detect objects in testing images by likelihood-based template matching. This is often called inference in machine learning literature.

In both learning and inference, we need to estimate the perturbations or activities in location and orientation $(\Delta x_{m,i}, \Delta \alpha_{m,i})$, which are treated as hidden variables. Given the template \mathbf{B} , they can be inferred by a local maximization operation.

2 Learning with unknown locations, scales, orientations and left-right flips

We have generalized the learning algorithm for the situation where the objects may appear at different locations, scales, orientations and left-right flips in the training images. Such images are often called non-aligned. We can extend the basic model (1) to model non-aligned images. For simplicity, suppose only the location of the object is unknown, then the model is of the following form:

$$\mathbf{I}_m = \sum_{i=1}^n c_{m,i} B_{x^{(m)} + x_i + \Delta x_{m,i}, s, \alpha_i + \Delta \alpha_{m,i}} + U_m, \quad m = 1, \dots, M, \quad (2)$$

where $x^{(m)}$ is the unknown location of the object in image \mathbf{I}_m . Model (2) can be further extended to incorporate other unknown variables.

The learning can be accomplished by a EM-like algorithm, which iterates the following two steps: (1) Given the current template, infer the unknown location, scale, orientation and left-right flip of the object in each training image. This enables us to align the images. (2) Given the inferred locations, scales, etc., re-learn the model using the basic learning algorithm from the aligned images.

This is a continuation of our work presented in our *Statistical Science* paper. In comparison to our previous work which allows unknown locations and scales of the objects, we now also allow the objects to appear at unknown orientations and left-right flips. We have done extensive experiments with the current learning algorithm. As a matter of fact, we have built a small library of more than 130 templates learned from non-aligned training images.

Figures (2), (3) and (4) show some of the examples of learning man-made objects, animals and birds, and leaves and flowers. Our experiments show that the model is capable of representing many natural or man-made objects.

The following webpage contains the code and data for this project:

<http://www.stat.ucla.edu/~ywu/AB/templates.html>



Figure 2: Learning from non-aligned images. In each row, the first plot shows the learned template, where each basis element is illustrated by a bar. The remaining plots show a few of the training images accompanied by the deformed templates that are matched to them.

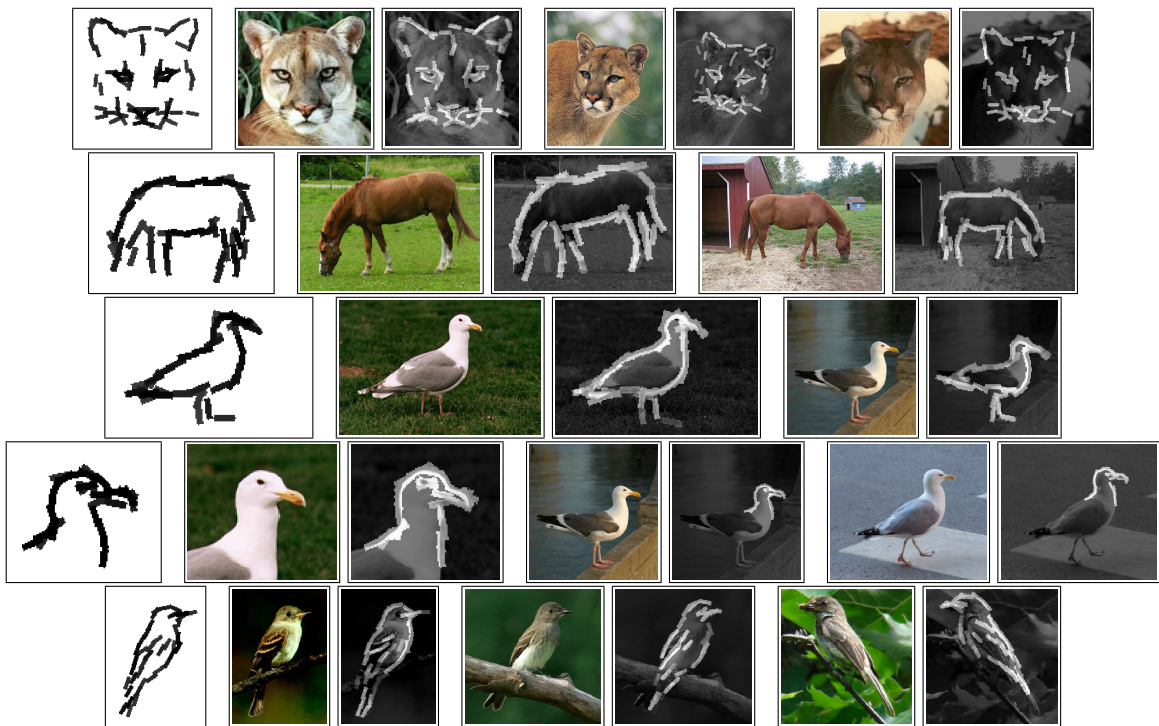


Figure 3: Learning from non-aligned images. In each row, the first plot shows the learned template, where each basis element is illustrated by a bar. The remaining plots show a few of the training images accompanied by the deformed templates that are matched to them.



Figure 4: Learning from non-aligned images. In each row, the first plot shows the learned template, where each basis element is illustrated by a bar. The remaining plots show a few of the training images accompanied by the deformed templates that are matched to them.

3 Discriminative adjustment by regularized logistic regression

The active basis model is a generative model, where the selected and perturbed basis elements seek to explain the training images. One can also select the basis elements by a discriminative approach such as adaboost, by bringing in negative examples. We have compared the behaviors of the generative and discriminative learning.



Figure 5: Discriminative versus generative learning. For each of the two experiments on bear and cat, the first plot shows the template learned by the discriminative approach. The second by the generative approach. The third by the generative approach that allows the local shift of the template in location, scale and orientation. The discriminative approach requires thousands of negative examples. The generative approach does not require negative examples.

As is shown in Figure (5), the templates learned by the generative approach tend to be cleaner than those obtained by the discriminative approach, especially if we allow the local shift of the template in generative learning using the method in section 2. The generative approach does not require negative examples, and is therefore much faster than the discriminative approach. For this reason, the local shift of the template is difficult to implement in adaboost, which is much more time-consuming than learning the active basis model.

Given the deformed template \mathbf{B}_m , the distribution of \mathbf{I}_m in the active basis model is in the form of an exponential family model, relative to a reference distribution which is assumed to be the distribution of natural images. If we treat the reference distribution as the distribution of negative examples, then the exponential family model leads to a logistic regression model for predicting class label. Therefore, given \mathbf{B}_m , we can re-estimate the parameters of the exponential family model by fitting a logistic regression, and this amounts to maximizing the partial likelihood instead of the full likelihood. The partial likelihood is less efficient than the full likelihood, but it is more immune to model misspecification. Such a discriminative adjustment leads to better classification performance on testing data.

In order to avoid overfitting, we introduce a ℓ_2 penalty term to regularize the logistic regression. We can also use SVM for discriminative adjustment.

Figure (6) illustrates the performance of discriminative adjustment on a head-shoulder data set. The ℓ_2 regularized logistic regression improves the classification performance of the original active basis model. It performs better than SVM and adaboost in terms of discriminative adjustment.

The following webpage contains the code and data for this project:

<http://www.stat.ucla.edu/~ywu/AB/ABEXP12.html>

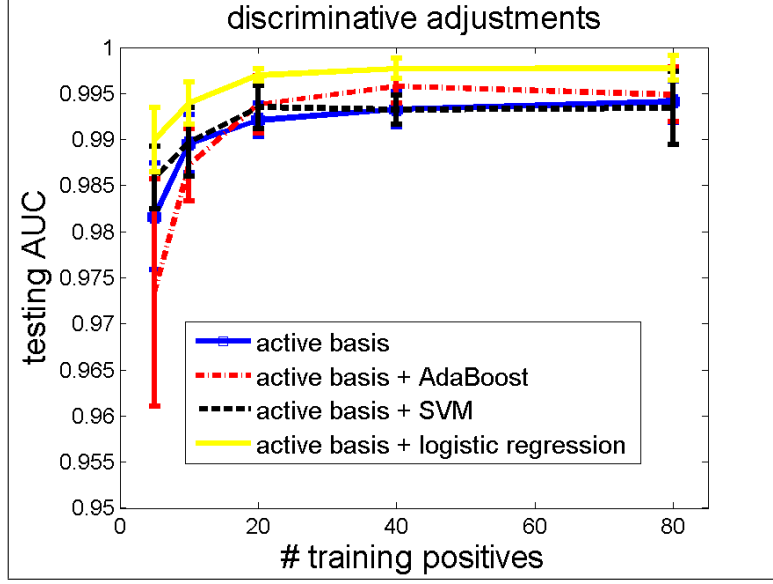


Figure 6: Adjusting the active basis model by discriminative methods on a head-shoulder data set. The curves are the testing AUC (area under curve) scores of the ROC curves.

4 Hierarchical active basis

Recall that an active basis model can be written in the following form

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

where \mathbf{B}_m is a deformed template that is composed of Gabor wavelet elements. We can further generalize the model to

$$\mathbf{I}_m = \sum_{k=1}^K C_{m,k} \mathbf{B}_{X_k + \Delta X_{m,k}, S_k + \Delta S_{m,k}, A_k + \Delta A_{m,k}} + U_m, \quad (4)$$

which is a composition of K deformable templates, each of which can be considered a part-template that can shift from its nominal overall location X_k , scale S_k and orientation A_k . Model (4) has a similar form to the basic model (3). Model (4) is a hierarchical model with two layers of movements: the movements of parts and the movements of the Gabor wavelet elements of the parts. Such a hierarchical active basis model is more flexible for modeling large deformations or articulations.

Figure (7) illustrates the basic idea. We can learn the part-templates such as ears, eyes, etc., and allow these part-templates to move relative to each other. Such extra flexibility allows the hierarchical active basis model to account for large deformations, as shown in the six examples in Figure (7). The learning of the part-templates is based on the method in section 2.

Figures (8) and (9) show two experiments where the learned hierarchical active basis models are matched to the testing images. The squared blocks are the part-templates, which are allowed to move relative to each other.

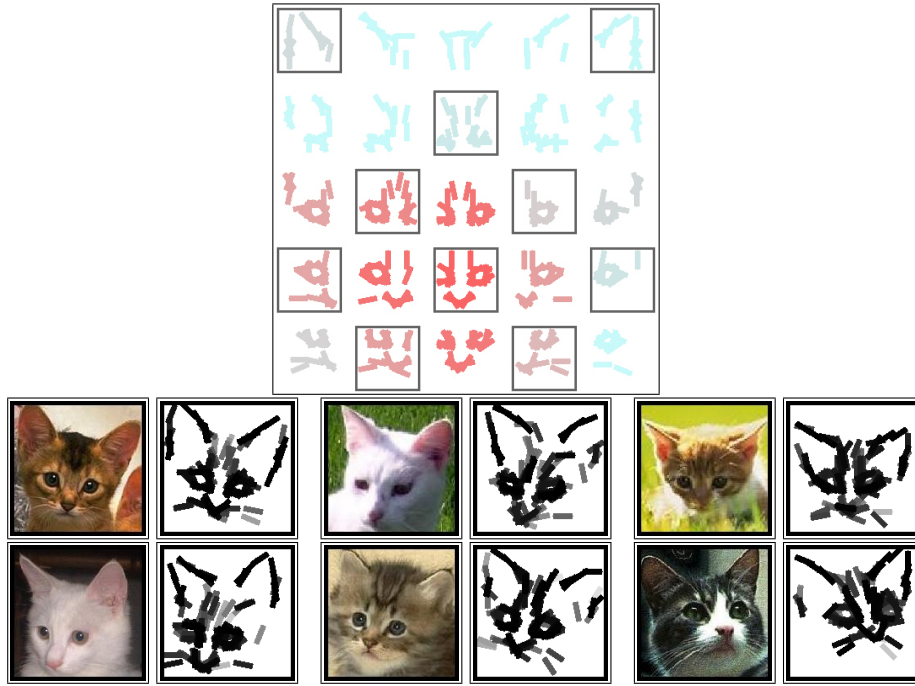


Figure 7: Learning part-templates from training images. The plot on top shows the learned part-templates. The color illustrates the log-likelihood scores (blue means low, and red means high). The remaining plots display some of the training images and the matched templates. By allowing the movements of part-templates, the hierarchical active basis model is more capable of capturing large deformations than the basic model.

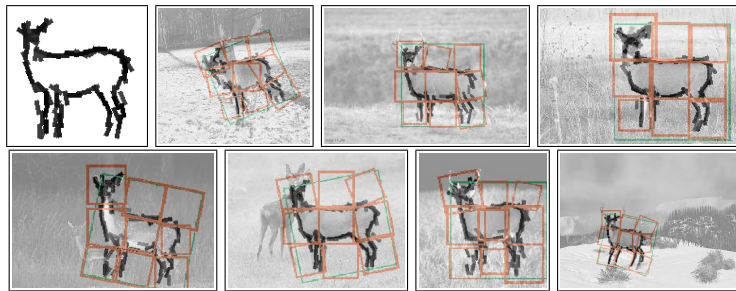


Figure 8: Matching the learned hierarchical active basis model to testing images. Each squared block is a part-template that is allowed to change its overall location, scale and orientation.

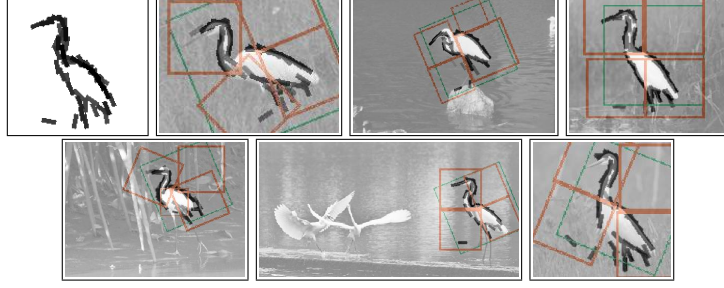


Figure 9: Matching the learned hierarchical active basis model to testing images. Each squared block is a part-template that is allowed to change its overall location, scale and orientation.

The following webpage contains the code and data for this project:

<http://www.stat.ucla.edu/~ywu/AB/ABEXP15.html>

5 Image representation by active primitives

In the hierarchical active basis model (4), the part-templates can also be designed to be simple geometric primitives such as corners, line segments and arc segments.

Figure (10) illustrates the basic idea of our recent work, which seeks to represent an image by simple geometric primitives that can deform. This leads to more compact representation than wavelets.

The paper is to appear in the proceedings of 2011 *International Conference on Computer Vision*.

The following webpage contains the code and data for this project:

<http://www.stat.ucla.edu/~wzhu/projects/AMSA/page11061601/index.html>

6 Model-based clustering

Recently, we have been focusing on model-based clustering and codebook learning. Finding clusters in the training data is an important problem in unsupervised learning. The active basis model, which is a generative model, can be naturally used for model-based clustering. We only need to assume the following model

$$\mathbf{I}_m = \sum_{i=1}^n c_{m,i} B_{x^{(m)}+x_i+\Delta x_{m,i}, s, \alpha_i+\Delta \alpha_{m,i}}^{(k)} + U_m, \quad m = 1, \dots, M, \quad (5)$$

where $\mathbf{B}^{(k)}$ is the template of category k . We can estimate the set of templates $\{\mathbf{B}^{(k)}, k = 1, \dots, K\}$ by fitting a mixture model using a EM-like algorithm.

Figure (11) shows two experiments. In the first row, our method is able to identify the six clusters in the training images which mix six categories of animal heads. In the second row, our method successfully identifies the heads of two types of birds. In model fitting, we allow the templates to shift locally, using the method described in section 2.

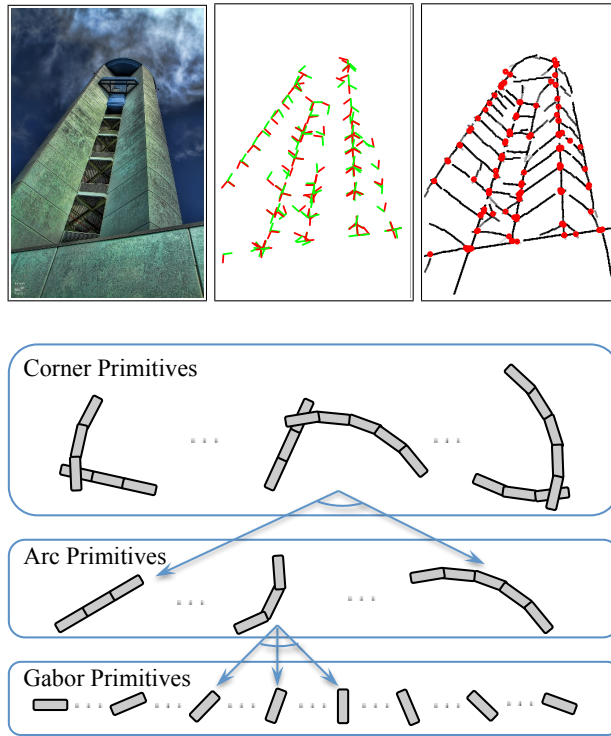


Figure 10: An image is represented by a small number of active corner and arc primitives. Left: Original Image. Middle: Selected corner primitives, where a corner is illustrated by the red arm and the green arm. For clarity of illustration, the red and green arms do not cover the whole extents of the two arc primitives of a corner. Right: Sketch the image by deforming the active primitives. Bottom: The primitives and their compositional relations.

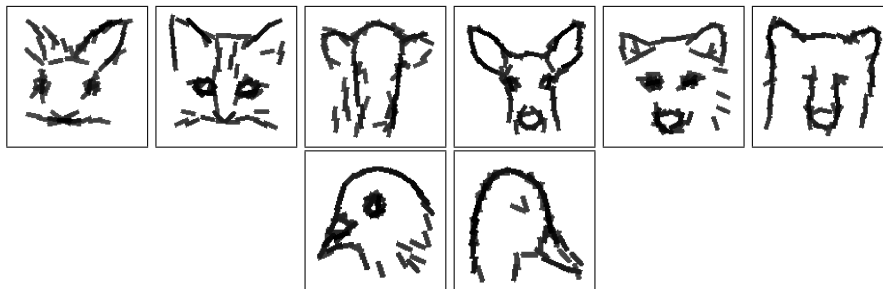


Figure 11: Finding clusters in images of animals and birds by fitting the mixture model using EM-like algorithm initialized by random clustering. First row: rabbit, cat, cow, deer, wolf, bear. Second row: pigeon, duck.

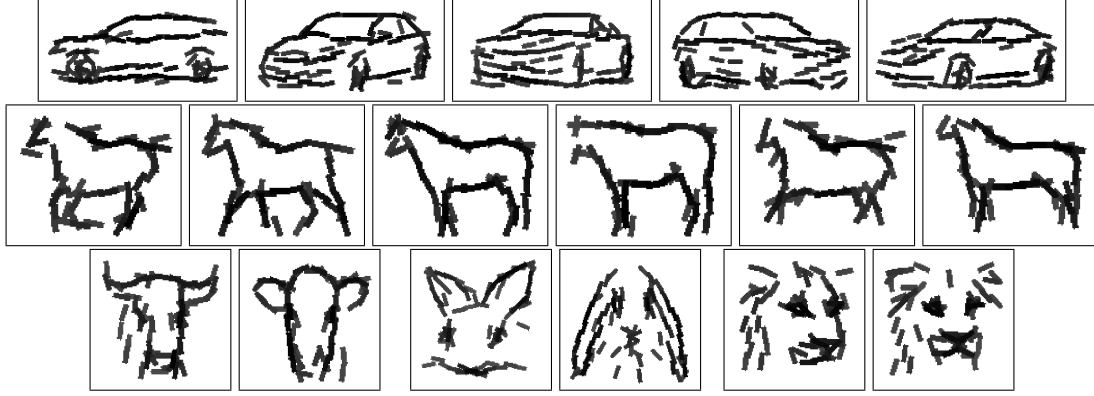


Figure 12: Finding clusters of different views, poses and subcategories within the same category by fitting the mixture model using EM-like algorithm initialized by random clustering: different views of cars, different poses of horses, different types of cows, different types of rabbits, different views of lions.

Our algorithm starts from random clustering. The convergence of the EM-like algorithm is very fast. In order to avoid being trapped by local modes, we re-start the algorithm multiple times, and choose the result that attains the highest likelihood.

It is also important to find clusters within the same category, because objects within the same category may appear at different poses and views, or there may be subcategories within the same category. Figure (12) illustrates some examples, where our method is capable of finding different views of the cars, different poses of the horses (here we show 6 out of 10 poses), different subcategories of cows and rabbits, and different views of lion heads.

The following webpage contains the code and data for this project:

<http://www.stat.ucla.edu/~ywu/AB/ABEXP17.html>

7 Unsupervised learning of codebooks of visual words

The ultimate goal of our research is the unsupervised learning of codebooks of part-templates, or what can be intuitively called “visual words.” This is perhaps the most fundamental issue in vision, and such unsupervised learning is what a generative model such as the active basis model is called for. In order to achieve this goal, we need to scale up the method in section 6. We have done some preliminary experiments, where we crop a large number of overlapping patches from training images and find clusters in these image patches using the method in section 6. Figure (13) displays the codebooks learned from some data sets.

We are now pursuing a more rigorous learning scheme that fits the following model

$$\mathbf{I} = \sum_{k=1}^K C_k \mathbf{B}_{X_k, S_k, A_k}^{(t_k)} + U, \quad (6)$$

where $\{\mathbf{B}^{(t)}, t = 1, \dots, T\}$ is a codebook of part-templates. We believe that this project will shed light on the neurons of the V2 area of the visual cortex. The learned codebooks can also be used for classification.



Figure 13: Learning codebooks of visual words from training images. The first row is the codebook learned from face images. The second row is the codebook learned from horse images. The third row is the codebook learned from a mixture of horse and face images.

The following webpage contains the code and data for this project:

<http://www.cs.ucla.edu/~yihong/ABM.html>

8 Incorporating texture statistics

We have extended the active basis model to incorporate texture statistics. We have also studied the modeling of motion sequences based on both sketches and textures.

The following webpage contains the code and data for this project:

http://www.stat.ucla.edu/~zzsi/mixed_template.html

Reproducibility

The code and data sets for reproducing most of the above findings can be downloaded at

<http://www.stat.ucla.edu/~ywu/AB/ActiveBasisMarkII.html>

http://www.stat.ucla.edu/~zzsi/hab/hab_changelog.html

<http://www.cs.ucla.edu/~yihong/ABM.html>

<http://www.stat.ucla.edu/~wzhu/projects/AMSA/page11061601/index.html>

http://www.stat.ucla.edu/~zzsi/mixed_template.html