

# Image Representation by Active Curves

Wenze Hu    Ying Nian Wu    Song-Chun Zhu  
Department of Statistics, UCLA  
{wzhu, ywu, sczhu}@stat.ucla.edu

## Abstract

This paper proposes a sparse image representation using deformable templates of simple geometric structures that are commonly observed in images of natural scenes. These deformable templates include active curve templates and active corner templates. An active curve template is a composition of Gabor wavelet elements placed with equal spacing on a straight line segment or a circular arc segment of constant curvature, where each Gabor wavelet element is allowed to locally shift its location and orientation, so that the original line and arc segment of the active curve template can be deformed to fit the observed image. An active corner or angle template is a composition of two active curve templates that share a common end point, and the active curve templates are allowed to vary their overall lengths and curvatures, so that the original corner template can deform to match the observed image. This paper then proposes a hierarchical computational architecture of sum-max maps that pursues a sparse representation of an image by selecting a small number of active curve and corner templates from a dictionary of all such templates. Experiments show that the proposed method is capable of finding sparse representations of natural images. It is also shown that object templates can be learned by selecting and composing active curve and corner templates.

## 1. Introduction

Finding sparse representations of natural images is one of the most fundamental problems in both computer and biological vision. A sparse representation gives a meaningful interpretation of the image. By varying the attributes of the sparse coding elements, we can generalize from one image to another, or construct and learn statistical models that capture the regularities and variabilities of images from various object or scene categories.

This paper proposes a sparse representation of images using active curve templates and active corner templates. In an active curve template, the prototype template is a composition or grouping of a small number of Gabor wavelet el-

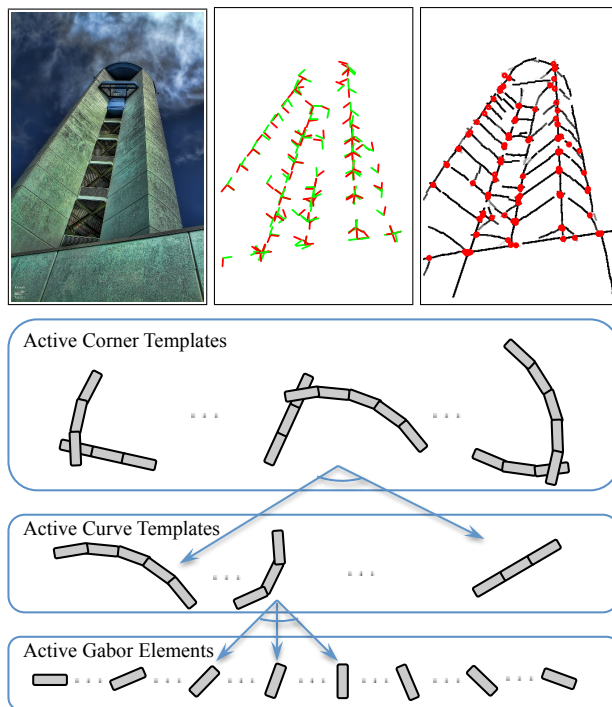


Figure 1. (Better viewed in color) An image is represented by a small number of active corner templates and active curve templates. **Top Row, Left:** Original image. **Middle:** Selected active corner templates, where a corner is illustrated by a red arm and a green arm. For clarity of illustration, the red and green arms do not cover the whole extents of the two curves of a corner template. **Right:** Sketch the image by deforming the active curve templates. **Bottom Three Rows:** Compositional relation between active curve templates and active corner templates.

ements placed with equal spacing on a straight line segment or a circular arc segment of constant curvature. So the prototype template is parametrized and very simple. However, these wavelet elements are allowed to locally shift their locations and orientations so that the prototype curve template can be deformed to different instances in order to fit observed images. Thus an active curve template is also flexible enough to account for deviations from the simple prototype.

Two active curve templates that share a common endpoint can be further composed into an active corner or angle template, where the two constituent active curve templates are allowed to vary their overall geometric attributes such as lengths and curvatures, so that a prototype corner template corresponds to many different variants. See Fig.1 for an illustration.

This paper then proposes a layered computational architecture of sum-max maps that pursues a sparse representation of an image by selecting a small number of active corner templates and active curve templates from a dictionary of all such templates. The computation has a bottom-up pass and a top-down pass. The bottom-up pass tests all the possible templates exhaustively but efficiently by parallel recursive computations that alternate between local sum pooling and local max pooling. The sum pooling scores the active curve and corner templates by combining the scores of their constituent components, while the max pooling optimizes the variations of the constituent components to infer deformations. After the bottom-up pass, the top-down pass selects the active corner templates and active curve templates, and deforms them into their optimal variants to represent the observed image, by retrieving the argmax elements in the local max pooling.

## 2. Literature Review

The proposed representation can be considered a concrete implementation of the idea of primal sketch [12, 7]. It is inspired by two theories on the primary visual cortex or V1. One is the sparse coding theory of Olshausen and Field [13] for the simple cells in V1. The other is the local max pooling proposed by Riesenhuber and Poggio [14] as a function of the complex cells in V1.

In the sparse coding theory of Olshausen and Field, an image is represented by a linear superposition of a small number of wavelet elements selected from a large dictionary. The active curve templates and corner templates in our work are based on the representation of Olshausen and Field, where each template is a composition or grouping of a small number of Gabor wavelet elements. Because of such grouping, the representation based on such templates can be considered a further sparse coding of the selected Gabor wavelets in the Olshausen-Field representation.

The proposed architecture of sum-max maps is a variation of the cortex-like structure of Riesenhuber and Poggio [14]. The difference between the proposed sum-max architecture and the structure of Riesenhuber and Poggio is that we test and select explicit geometric templates of curves and corners, and there is a top-down pass that deforms the templates by retrieving the argmax elements in the local max pooling, in order to represent and sketch the observed image. This procedure is also similar to the inference algorithm in hierarchical object models such as [4].

The proposed representation follows the general principle of hierarchical compositionality and re-usable parts [1]. In terms of detecting large structures by grouping smaller ones, the proposed sum-max architecture tests all possible templates exhaustively instead of resorting to greedy schemes that make early decisions which may not be correct. Specifically, no edge detection is performed before the active curve and corner templates are detected. Decisions on edges are made after the templates are detected and deformed to match the observed data.

The proposed representation is different from feature detectors such as edge detectors [2, 16], corner detectors [8, 15], or invariant features such as SIFT [10]. The active curve and corner templates used in our representation are sparser and usually cover larger ranges than feature detectors.

The method of meaningful alignment [3] also seeks to find geometric structures in images. The line segments tested by meaningful alignment are not deformable, and the computation there does not employ recursive schemes such as those in sum-max maps.

## 3. Active Curve and Corner Templates

Following the active basis model [18], an image  $\mathbf{I}$  is represented by

$$\mathbf{I} = \sum_{i=1}^N c_i B_{\mathbf{x}_i + \Delta \mathbf{x}_i, \theta_i + \Delta \theta_i} + \mathbf{U}, \quad (1)$$

where each  $B$  is a Gabor wavelet element with position  $\mathbf{x}$  and orientation  $\theta$ . Each Gabor wavelet element has a pair of sine and cosine components, so the coefficient  $c_i$  also has a pair of components correspondingly.  $\mathbf{U}$  is the unexplained residual image. In this paper, we fix the scale of Gabor elements, so this attribute is not shown in the subscript. We allow each Gabor wavelet element  $i$  to shift from its nominal position  $\mathbf{x}_i$  and orientation  $\theta_i$  by  $\Delta \mathbf{x}_i$  and  $\Delta \theta_i$  respectively. The nominal positions and orientations are either designed as in this paper or learned as in the active basis model, and the perturbations  $\Delta \mathbf{x}_i$  and  $\Delta \theta_i$  deform the template to fit the image  $\mathbf{I}$ . These perturbations can be inferred by the local max pooling of the Gabor filter responses.

As is shown in Fig.2, an active curve template  $A$  is a composition or grouping of Gabor elements placed with equal spacing on a straight line segment or a circular arc segment of a constant curvature. By design, the Gabor elements are placed so that the distance between two consecutive elements is 0.9 times their lengths, so these elements are slightly overlapping. As an approximation, we still treat them as being orthogonal to each other, so that the coefficients  $c_i$  are simply the projections of the image  $\mathbf{I}$  on the Gabor wavelet elements. As Gabor wavelets resemble the

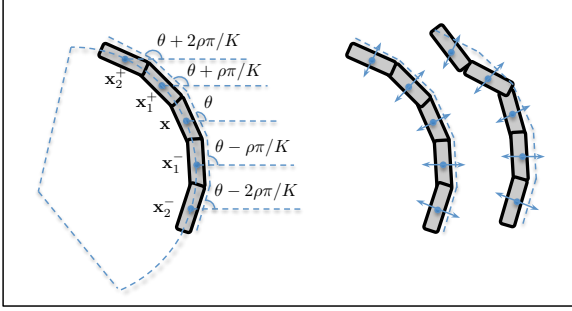


Figure 2. Illustration of active curve templates. **Left:** an example of active curve template  $A_{\mathbf{x}, \theta, \rho, 2}$ . **Right:** the active curve template and one of its possible variant. Blue arrows indicate the location deformation range of the Gabor elements.

edge and ridge patches, they are symbolically illustrated by bars.

We use the center position  $\mathbf{x}$ , orientation  $\theta$ , curvature  $\rho$  and length  $l$  to index each active curve template, which is a composition of Gabor wavelet elements:

$$A_{\mathbf{x}, \theta, \rho, l} = (B_{\mathbf{x}_0, \theta_0}, B_{\mathbf{x}_i^+, \theta_i^+}, B_{\mathbf{x}_i^-, \theta_i^-}, i = 1, \dots, l-1), \quad (2)$$

where  $\mathbf{x}_0 = \mathbf{x}$ ,  $\theta_0 = \theta$ ,  $\mathbf{x}_i^\pm$  refer to the position of the  $i$ -th Gabor element, and  $\pm$  denotes the two sides relative to the arc center.

We quantize the orientations of Gabor wavelets into  $K$  values equally spaced by  $\alpha = \pi/K$ . The curvature is then quantized such that the orientation difference between neighboring elements is a multiple of  $\alpha$ . We use this integer to index the curvature, thus an active curve template of curvature  $\rho$  satisfies:  $\theta_i^+ = \theta_{i-1}^+ + \rho \cdot \alpha$ , and  $\theta_i^- = \theta_{i-1}^- - \rho \cdot \alpha$ . The length is quantized by the length of Gabor wavelets. A curve of length  $l$  has  $2l + 1$  Gabor wavelet elements.

The active curve template reduces to a line segment if  $\rho = 0$ , and it further reduces to a single Gabor element if  $l = 0$ . We use arcs as prototype shape because they are more general than straight line segments, yet simpler to parametrize than more general curves such as splines.

An active corner or angle template is a composition of two active curve templates. As is shown in Fig.3, an active corner template  $C$  is defined in terms of two active curve templates that share a common end point  $\mathbf{x}'$ :

$$C = \{A_{\mathbf{x}', \theta'_1, \rho_1, l_1}, A_{\mathbf{x}', \theta'_2, \rho_2, l_2}\}, \quad (3)$$

where we use superscript  $'$  to indicate that the curve is indexed by its end point instead of its center. We require that the curvature  $\rho$  should be below a threshold  $b_1$ , the length  $l_1$  and  $l_2$  should be larger than a threshold  $b_2$ , and the angle  $\delta = \theta'_2 - \theta'_1$  should be confined to a proper range, *e.g.*  $\delta \in [\pi/2 - b_3, \pi/2 + b_3]$ , where  $b_1$ ,  $b_2$  and  $b_3$  are parameters. Through these constraints, only long curves with low curvatures can be composed into corners.

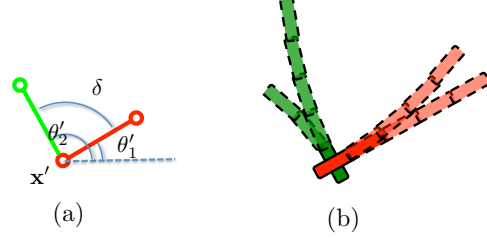


Figure 3. (a) A schematic plot of an active corner at position  $\mathbf{x}'$  and two arm orientations  $\theta'_1, \theta'_2$ . (b) By definition, an active corner corresponds to many possible instances of corners. For example, we can choose any pair of red and green arms in (b) to compose them into an instance of the active corner in (a).

#### 4. Image Model and Template Matching Score

Following the active basis model, we assume that the probability distribution of the image,  $p(\mathbf{I})$ , is specified by tilting a background distribution  $q(\mathbf{I})$ , which may be considered as the distribution of natural images. Theoretical underpinnings of this model can be found in [18].

Given the  $N$  Gabor elements of a template, the log likelihood ratio of foreground image versus the background image is:

$$\log \frac{p(\mathbf{I})}{q(\mathbf{I})} = \sum_{i=1}^N [\lambda h(r_i) - \log Z(\lambda)], \quad (4)$$

where  $h(r_i)$  is a monotone increasing transformation of Gabor element response  $r_i = |c_i|^2$ , and  $h(r_i)$  saturates for large  $r_i$ .  $\lambda$  is a parameter, and  $Z(\lambda)$  is the normalizing constant for the corresponding exponential distribution. This is essentially a linear scoring scheme. If  $r_i$  is small, it will lead to a negative score. We want to detect those curve templates whose constituent elements contribute large positive scores. Different values of  $\lambda$  correspond to different expectations  $E[h(r_i)]$  on foreground, and  $E[h(r_i)]$  is a monotone increasing function of  $\lambda$ . We treat all the selected Gabor elements with equal importance, so we use the same  $\lambda$  for all  $r_i$ . This  $\lambda$  is kept as a parameter in our model, and the corresponding  $Z(\lambda)$  is computed by pooling Gabor responses from natural images.

With such a model, we can simply define the template matching score of an active curve template as the log likelihood ratio score:

$$\lambda \left\{ h(r_0) + \sum_{i=1}^{l-1} [h(r_i^+) + h(r_i^-)] \right\} - (2l + 1) \log Z. \quad (5)$$

The template matching score of a corner template is defined by summing the scores of the two constituent curve templates.

## 5. Scoring All Templates by Sum-Max Maps

According to the definition of the active curve templates in Eqn.(2), an active curve template of length  $l$  can be composed by an active curve of length  $l - 1$ , plus two Gabor elements (i.e., active curves of length 0) at two ends:  $A_{\mathbf{x},\theta,\rho,l} = \{A_{\mathbf{x},\theta,\rho,l-1}, A_{\mathbf{x}_i^+, \theta+l\cdot\alpha,\rho,0}, A_{\mathbf{x}_i^-, \theta-l\cdot\alpha,\rho,0}\}$ .

This immediately implies a recursive algorithm to compute the template matching score Eqn.(5) of long active curve templates by first computing the scores of shorter ones. Scores of length zero active curves are obtained by local maximum pooling of the Gabor filter responses, because the Gabor elements are allowed to shift their locations and orientations. The computation can be highly parallel, since scores of active curves at different positions, orientations and curvatures can be computed independently. Based on these two features, the following sum-max maps are proposed to compute scores of all active curve and corner templates.

**1. S1 maps.** S1 maps store the responses of Gabor filters at all positions and orientations. It is computed by convolving the Gabor filter with the image, followed by local normalization and saturation transformation on the responses. The local normalization step divides the response of a Gabor wavelet element by the local average of responses of all the Gabor elements within a window centered at this element.

**2. M1 maps.** To account for the deformations of active curve templates, each element on M1 maps is computed as the maximum of S1 map values within its local deformation range:  $\mathbf{M1}(\mathbf{x}, \theta) = \max_{\Delta\mathbf{x}, \Delta\theta} \mathbf{S1}(\mathbf{x} + \Delta\mathbf{x}, \theta + \Delta\theta)$ , where  $\Delta\mathbf{x} = d \cdot (\cos \theta, \sin \theta)$ ,  $d \in [-b_4, b_4]$  and  $\Delta\theta \in [-b_5, b_5]$ .

**3. S2 maps.** The S2 maps store the scores of all the active curve templates. According to curve length, S2 maps can be further divided into sub-layers. Scores at different sub-layers are computed as follows:

$$\mathbf{S2}(\mathbf{x}, \theta, \rho, 0) = \lambda \mathbf{M1}(\mathbf{x}, \theta) - \log Z, \quad (6)$$

$$\begin{aligned} \mathbf{S2}(\mathbf{x}, \theta, \rho, l) &= \mathbf{S2}(\mathbf{x}, \theta, \rho, l-1) \\ &+ \mathbf{S2}(\mathbf{x}_{i-1}^+, \theta + l \cdot \alpha, \rho, 0) \\ &+ \mathbf{S2}(\mathbf{x}_{i-1}^-, \theta - l \cdot \alpha, \rho, 0), \quad (7) \end{aligned}$$

where in Eqn. (7), we assume  $l > 0$ .

The template matching score of active corner templates can be expressed as  $\mathbf{S2}(\mathbf{x}', \theta'_1, \rho_1, l_1) + \mathbf{S2}(\mathbf{x}', \theta'_2, \rho_2, l_2)$ .

According to Eqn.(3), for a given point  $\mathbf{x}'$  and a given supporting curve orientation  $\theta'_1$ , there are  $2b_1 \cdot (|l| - b_2) \cdot |\delta|$  different corners. If we assume that there can be at most one corner instance among these, we shall choose it by solving the following problem:

$$\max : \mathbf{S2}(\mathbf{x}', \theta'_1, \rho_1, l_1) + \mathbf{S2}(\mathbf{x}', \theta'_2, \rho_2, l_2),$$

subject to the constraints on the ranges of  $\rho$  and  $l$  for each curve, and  $\delta$  spanned by the two curves.

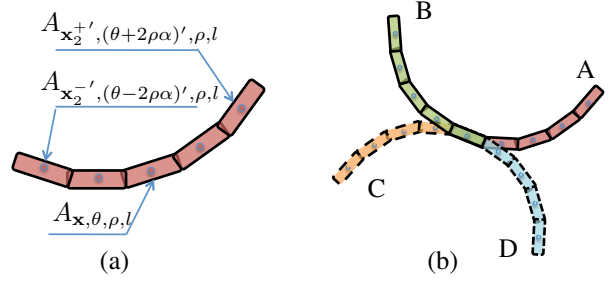


Figure 4. (a) Three indices of an active corner, by center point index, end point index, and two ends. (b) When changing center point index to end point index, one index will correspond to two curves (such as A,B or C,D). We solve this overriding issue by extending the orientation index from  $[0, K - 1]$  to  $[0, 2K - 1]$ , and setting the orientation of one curve to  $\theta + K$ .

Since there is no constraint on the relation between  $(\rho_1, l_1)$  and  $(\rho_2, l_2)$ , this problem can be solved by first solving smaller problems. The corresponding computation steps can be summarized as sequentially computing the following maps:

**4. M2 maps.**  $\mathbf{M2}(\mathbf{x}', \theta') = \max_{\rho, l} \mathbf{S2}(\mathbf{x}', \theta', \rho, l)$ , where  $-b_1 < \rho < b_1$  and  $l > b_2$ .

**5. S3 maps.**  $\mathbf{S3}(\mathbf{x}', \theta'_1, \theta'_2) = \mathbf{M2}(\mathbf{x}', \theta'_1) + \mathbf{M2}(\mathbf{x}', \theta'_2)$ , where S3 maps are scores for all the possible active corner templates. In computing S3 maps, we should only compute over  $(\theta'_1, \theta'_2)$  pairs so that the spanned angle  $\delta$  is within the defined range.

In the local maximization steps 2 and 4, the optimal deformations that achieve the maxima can be obtained by arg max operation.

---

### Algorithm 1: Matching Pursuit of Active Curve Templates

---

**Input:** S1 maps after inhibition, threshold  $T$ .

**Output:** Selected active curve templates

$$\mathbf{A} = \{A_1, \dots, A_N\}$$

```

1 repeat
2   Compute M1 and S2 maps.;
3    $s^* \leftarrow \max \mathbf{S2}(\mathbf{x}, \theta, \rho, l)$ ;
4    $A^* \leftarrow \arg \max \mathbf{S2}(\mathbf{x}, \theta, \rho, l)$ ;
5   Decompose  $A^*$  into a set of Gabor elements  $\mathbf{B}$ ;
6   Backtrack the deformed  $\mathbf{B}^*$  of  $\mathbf{B}$ ;
7   foreach  $B \in \mathbf{B}^*$  do
8      $\lfloor$  Set correlated entries on S1 to 0;
9      $\mathbf{A} \leftarrow \mathbf{A} \cup \{A^*\}$ ;
10 until  $s^* < T$ ;

```

---

The sum-max maps score all the active curve and corner templates without early decisions such as edge detection. The max operations make the scores invariant to shape deformations.

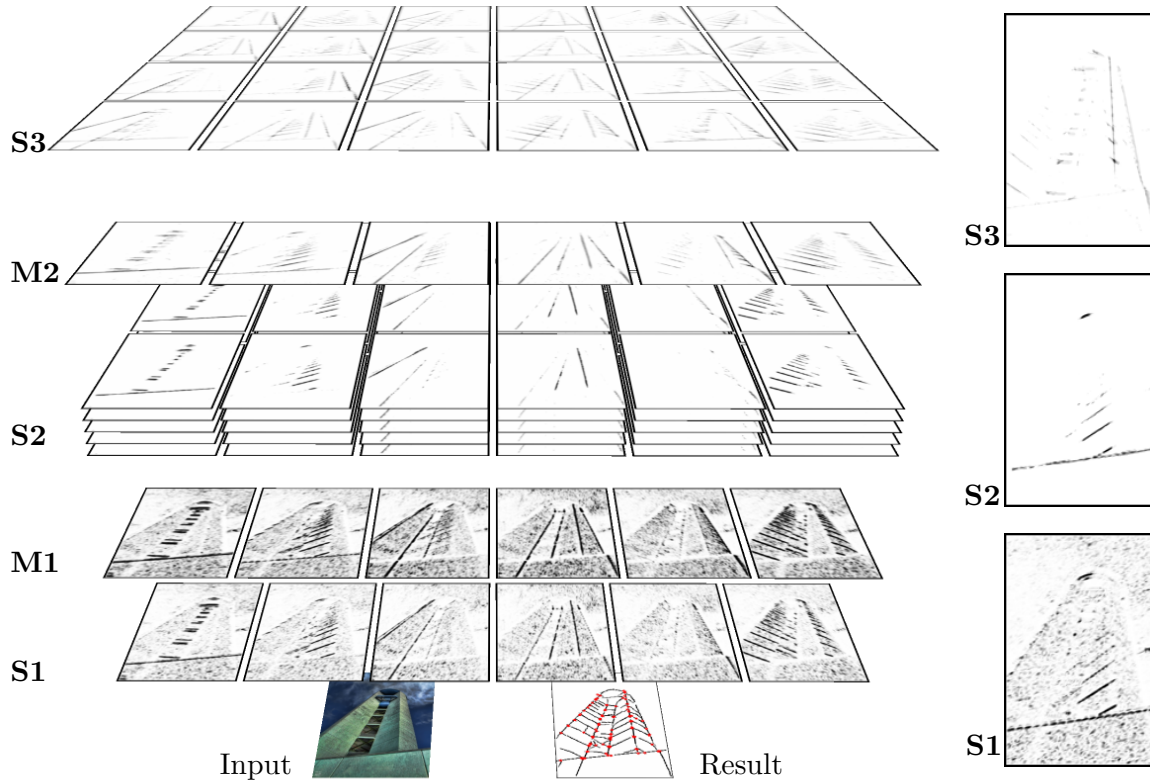


Figure 5. The sum-max data structure, using the image in Fig.1 as an example. Maps for different orientations are laid out horizontally, and maps for different layers are laid out vertically. Sub-layers in the **S2** maps correspond to active curves of different lengths, and depth direction in the **S2** maps corresponds to curves of different curvatures. Depth direction in the **S3** maps corresponds to orientation of the second arm. Maps on the right are typical maps from each sum layer.

## 6. Selection of Corner and Curve Templates

As can be seen from Fig.5, most responses on **S3** maps are very low, because corners are rare events and highly location sensitive. So, after the **M3** maps are computed, we can simply apply a threshold and use non-maximum suppression to select active corner templates for image representation. Specific curves and their Gabor elements that belong to these selected corners can then be obtained by a top-down process that retrieves the optimal deformations from the  $\arg \max$  maps of the local max operations.

To select active curve templates that are not overlapping with the selected active corner templates, we need to inhibit the scores of those curves that overlap with the selected corners. To do this, we trace back the active curve templates that belong to the selected active corner templates, and trace back the Gabor elements of these curves. For each Gabor element of the selected and deformed corner, we let it inhibit its nearby Gabor elements whose correlations with this Gabor element exceeds a certain threshold, by setting the filter responses of those Gabor elements to zero. In practice, the correlations can be computed before the algorithm starts, so that inhibition can be done very efficiently.

After inhibition, we can select salient active curve templates by sequentially selecting the current best curve with the highest score, and let the selected curve inhibit the overlapping ones. This is essentially a matching pursuit [11] process, where in each step, we select a group of Gabor elements packed into an active curve template. The pursuit algorithm stops once the selected score is lower than a threshold  $T$ . The process is summarized in Algorithm 1.

## 7. Experiments

### 7.1. Implementation and Parameters

The recursive and parallel algorithm is mainly composed of sum and max operations, which can be computed very efficiently. However, if we keep all the maps and  $\arg \max$  maps, the memory consumption can be excessive. In practice, since only a small number of Gabor elements will be backtracked in top-down process, we choose not to save the  $\arg \max$  maps and instead re-do the local maximum computing for selected corners or curves to retrieve their  $\arg \max$  deformations. For an image of size  $341 \times 512$  and parameters in Table.1, the algorithm would consume about



Parameter	Notation	Value
Gabor Filter Size	$s$	$15 \times 15$ pixel
Number of Orientations	$K$	25
Position Change	$b_4$	$\pm 1$ pixel
Orientation Change	$b_5$	$\pm 1$ orientation
Curve Curvature Range	$\rho$	$\rho \in [-4, 4]$
Curve Length Range	$l$	$l \in [0, 3]$
Corner Arm curvature Threshold	$b_1$	2
Corner Arm Length Threshold	$b_2$	0
Corner Angle Range	$b_3$	$\pi/6, \delta \in [\pi/3, 2\pi/3]$
Weight	$\lambda$	2.5
Arc Score Threshold	$T$	$3\{\lambda \cdot E[h(r); \lambda] - \log Z\}$
Corner Score Threshold	$b_6$	$2.3T$

Table 1. Parameters for all the experiments on single images

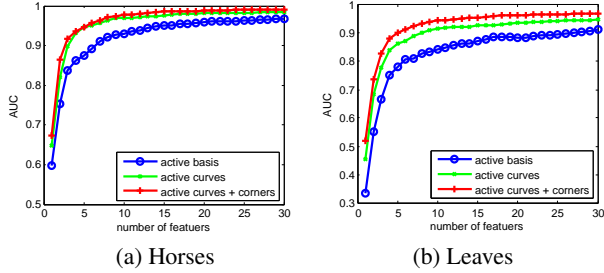


Figure 6. AUC curves of boosted classifiers on horse images and leaves images, using active Gabor features, active curve templates, and both active corner and curve templates.

1.2 GB of memory. Although we listed 12 parameters in the table, most of them are not essential, and only for properly discretizing the space of possible curves and corners. Only  $\lambda$ ,  $T$ , and  $b_6$  are of importance. We calibrate the last three parameters according to the image in Fig.1, and fix these parameters for all the experiments on single images below and those in the project page.

## 7.2. Sketch Single Images

Fig.5 shows the intermediate results for the example image in Fig.1, where each row shows maps from different layers, and each column corresponds to curves or corners in different orientations. As the size of active curve and corner templates become larger, the high scores on corresponding maps become sparser and more clustered. Maps at  $S3$  level become very sparse, and high scores only appear around the true corner points of the image. As evidences are accumulated over larger areas, it becomes safer to make decisions at higher levels. Since the selected templates are much larger than a single Gabor wavelet, the resulting representation is much sparser than wavelet representation.

More results are shown in Fig.9 and in the project page. For each image, we also show the result from edge link [9] as a reference. We would like to stress that our representation is much sparser than the representation based on edge points.

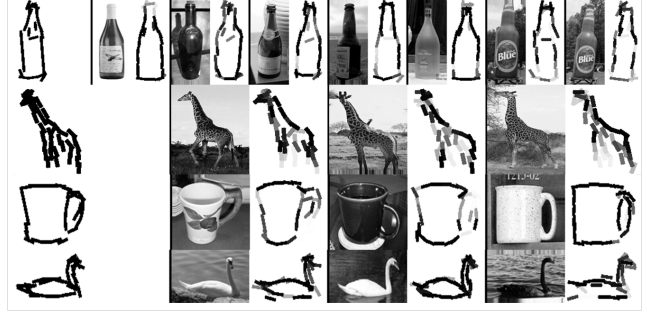


Figure 7. Learned object templates and their deformations on training images. The learned template for each category is shown in the first column. The rest of the templates are deformed versions for images on their left. Grayscale of the strokes in the deformed templates indicates the strengths of Gabor responses, where darker means stronger.

## 7.3. Learn from Multiple Images

We can also select and compose curve templates into an object template by learning from multiple images, either generatively or discriminatively. To accommodate geometric variations of active curve templates across multiple object images, we extend the  $S2$  and the  $S3$  maps as below, and use them as features to model object categories:  $M2^+(x, \theta, \rho, l) = \max_{\Delta x, \Delta \theta, \Delta \rho} S2(x + \Delta x, \theta + \Delta \theta, \rho + \Delta \rho, l)$ , and  $M3^+(x', \theta'_1, \theta'_2) = \max_{\Delta x} S3(x' + \Delta x, \theta'_1, \theta'_2)$ .

To illustrate the use of the above features for object modeling, we collect a dataset and compare the testing errors of adaBoost classifiers [6, 17] based on different sets of features. We collected 280 horse images, 186 leaf images as positive examples, and 559 images as negative examples. The leaf images are from Vision lab of Caltech. Using 20% of those images as training examples, we use adaBoost to build three discriminative models, by replacing the Haar features by the feature pools of  $M1$  maps,  $M2^+$  maps, and both  $M2^+$  and  $M3^+$  maps. We compute the AUC scores of the classification results on the rest 80% of the images. For each category, we repeat the above procedure five times by randomly splitting the data at each time, and show the averaged AUC in Fig.6. From the results, we can see that to achieve the same classification performance (AUC), the number of active corners or curves used is always less than that of active Gabor wavelets.

To show the potential of learning deformable object templates, we use the ETHZ object dataset [5], and run the shared matching pursuit algorithm in [18] over the  $M2^+$  maps to select curve templates which are composed into a deformable object template. Templates learned using the first half of images in each category are shown in Fig.7.

We then perform the detection task on 5 random splits of the dataset as in [5]. For each split, we learn the tem-

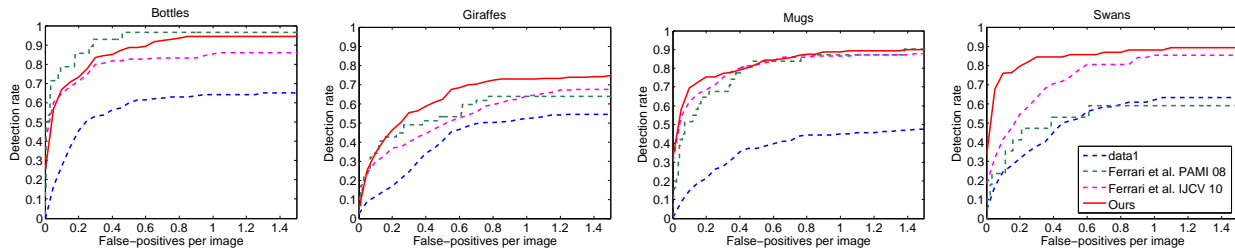


Figure 8. Detection results on the ETHZ dataset. We use the same evaluation procedure and criterion as [5]. Blue, green and purple curves are from [5]. Note that only one repetition is used in producing the green curve.

plates using training data, and use  $\ell_2$  regularized logistic regression to adjust the weight of each Gabor element in the learned template. Negative examples are randomly cropped from backgrounds of training images. We then match our template by sliding windows over multiple scales of testing images, and report detection windows by non-maximum suppression.

We follow the evaluation criterion in [5] and show the detection performance in Fig.8. From the results, we can see that models using our representation can achieve comparable performance as [5].

## 8. Discussion

We propose a sparse image representation based on deformable templates of simple geometric structures such as curves and corners. We also propose a computational architecture that exhaustively tests all the possible deformable templates in the bottom-up pass and then retrieves the deformed templates in the top-down pass. Such templates can be pursued from single images. They can also be composed into object templates by learning from multiple images. The template matching scores of curve and corner templates can also be used as features for discriminative learning.

Besides corners and angles, it is possible to compose the curve templates into templates of more complex geometric shapes such as triangles, rectangles and ellipsoids. But these are less frequent than corners or angles, and it takes more computations to test such templates.

In our current scheme of template matching, we only include the responses of Gabor wavelet elements on the line or arc segments. We should also pool the responses of those elements that are away from line or arc segments, in the form of local averages or local max, which are weighted negatively in order to model the flatness of the regions around the line or arc segments.

## Reproducibility

All the results reported in this paper can be reproduced by code and data included in the project page: <http://www.stat.ucla.edu/~wzhu/IRAC>

## Acknowledgement

This project is supported by NSF DMS 1007889 and NSF IIS 1018751.

## References

- [1] E. Bienenstock, S. Geman, and D. Potter. Compositionality, mdl priors, and object recognition. In *NIPS*, 1997.
- [2] J. Canny. A computational approach to edge detection. *PAMI*, 1986.
- [3] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *IJCV*, 2000.
- [4] B. Epshtein, I. Lifshitz, and S. Ullman. Image interpretation by a single bottom-up top-down cycle. *Proceedings of the National Academy of Sciences*, 2008.
- [5] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *IJCV*, 2010.
- [6] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *ICML*, 1996.
- [7] C. Guo, S.-C. Zhu, and Y. N. Wu. A mathematical theory of primal sketch and sketchability. In *ICCV*, 2003.
- [8] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference*, 1988.
- [9] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. <<http://www.csse.uwa.edu.au/~pk/research/matlabfns/>>.
- [10] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
- [11] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 1993.
- [12] D. Marr. *Vision*. W. H. Freeman, 1982.
- [13] A. B. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 1996.
- [14] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 1999.
- [15] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.
- [16] I. Sobel. An isotropic 3x3 image gradient operator. *Machine Vision for Three-Dimensional Scenes*, 1990.
- [17] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
- [18] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *IJCV*, 2010.

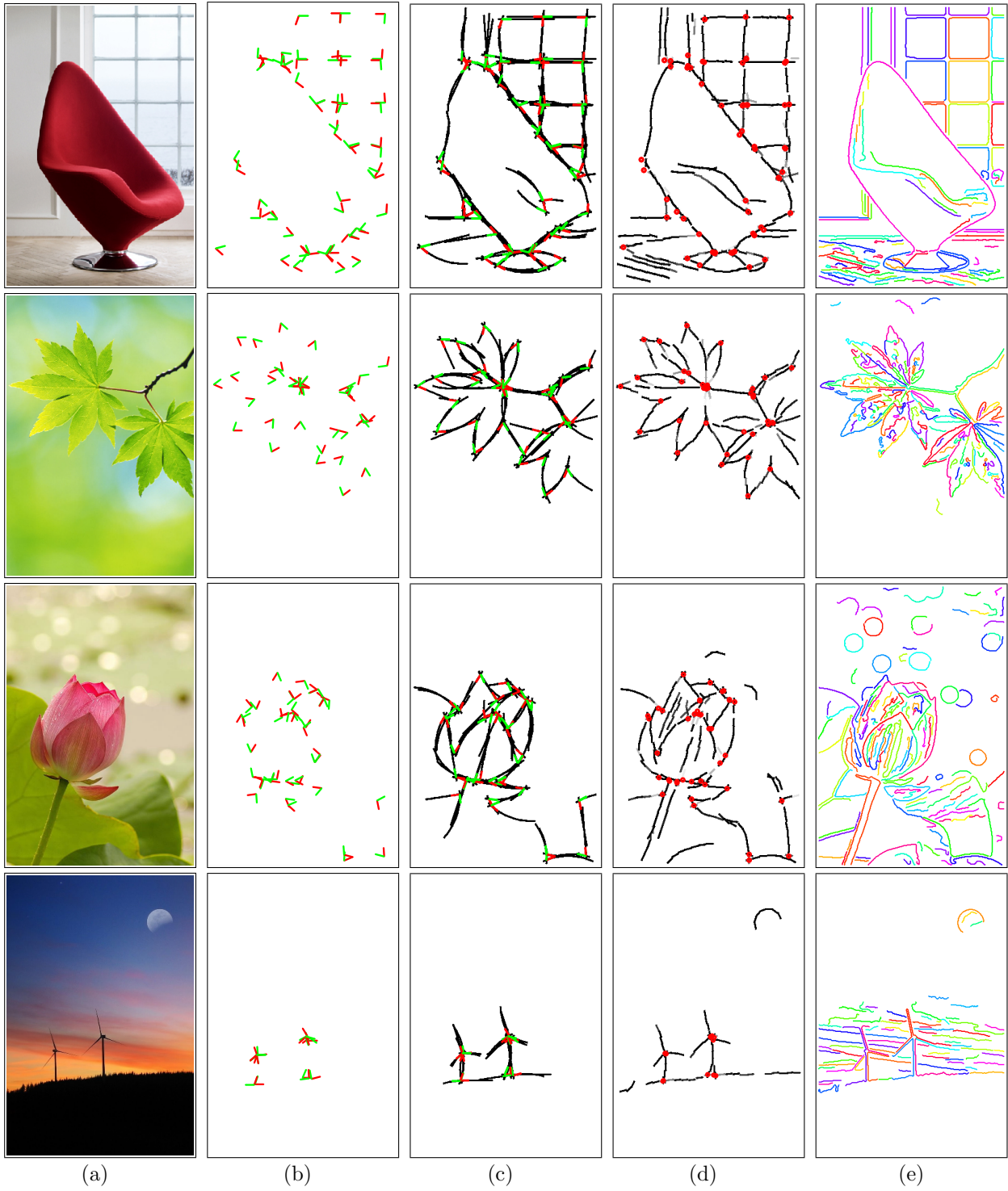


Figure 9. More results on single image representation. (a) Input image. (b) Selected corners. (c) Selected corners with supporting curves. (d) Deformed curves. (e) Output from edgeline code.