Learning a Dictionary of Shape Epitomes with Applications to Image Labeling: Supplementary Material

Liang-Chieh Chen[1], George Papandreou[2], and Alan L. Yuille[1,2]
Departments of Computer Science[1] and Statistics[2], UCLA
lcchen@cs.ucla.edu, {gpapan, yuille}@stat.ucla.edu

1. Introduction

In the main paper, we have proposed a new representation for image shape: a dictionary of shape epitomes, which explicitly include hidden variables to encode shift and rotation. We have also applied this representation for semantic image labeling. In this supplementary material, we further show the versatility of shape epitomes by employing them for edge detection and local appearance modeling.

2. Edge detection

In our learned dictionary of shape epitomes, we find that the first five shape epitomes can encode the ground truth of standard image labeling datasets (MSRC-21, and Stanford Background datasets) with high accuracy. It is natural to ask if we can apply this epitomic shape dictionary to edge detection, since the dictionary provides mid-level edge orientations.

Object shapes in images can be encoded either in a region-centric or its dual edge-centric fashion, which are respectively more suitable for the tasks of image labeling and edge detection. Shape epitomes can naturally be employed in both of these shape representations. In the context of edge detection, we encode every image patch by one of the candidate segmentation templates. We use the region boundaries within the selected segmentation template as the detected edges reported by our method.

Similarly to our image labeling experiments reported in the main paper, we quantize shapes into the same five 25×25 shape epitomes. We extract 17×17 segmentation templates from the shape epitomes at 9 possible postitions (using a stride of 4 pixels) and 4 rotations, resulting in a total of 181 segmentation templates, including the flat template ($9\times5\times4+1$).

2.1. Model

Most existing edge detection systems such as [3, 4, 1] employ a pixel-based representation of image edges. In this context, one needs to assign to each pixel in the image an on/off edge label, also possibly accompanied by the orientation of the candidate boundary. Extra post-processing steps such as non-maxima suppression are needed to ensure consistency among the pixel-level decisions. On the contrary, our epitomic shape representation reasons about the existence or lack of edges at the patch level. The learned dictionary of shape epitomes effectively regularizes the decision process by only allowing edges of plausible shape to be detected.

Suppose T_j is the segmentation template type for patch P_j, where T_j can take values from $\{0, 1, ..., 180\}$. The flat template is indexed by 0. We follow the same pipeline and code from [1]. Within each region determined by a segmentation template, we extract the histograms of CIE Lab colors and textons, and compute the chi-square distances between the two histograms as features to train a multinomial logistic regression model. Therefore, given the type of segmentation template $T_j = i$, we extract the features $\boldsymbol{f}_i(j) = (1, G_{iT}(j), G_{iL}(j), G_{iA}(j), G_{iB}(j))$, where $G_{iT}(j)$ is the chi-square distance between the two histograms of textons, and similarly $G_{iL}(j)$, $G_{iA}(j)$, and $G_{iB}(j)$ for CIE Lab colors. Note the feature depends on both P_j and segmentation template type indexed by i. The model parameters $\boldsymbol{\alpha} = (b, \alpha_T, \alpha_L, \alpha_A, \alpha_B)$ are the weights for each feature (b is the bias term). We tie the model parameters for all the types of segmentation templates, i.e., $\boldsymbol{\alpha}$ is the same for $i \in \{0, ..., 180\}$. The multinomial logistic regression model is:

$$Pr(T_j = i|P_j) = \frac{1}{Z}\exp\{-\boldsymbol{\alpha}^T \boldsymbol{f}_i(j)\} \quad (1)$$

for $i \in \{1, ..., 180\}$ and

$$Pr(T_j = 0|P_j) = \frac{1}{Z}$$

for the flat template $i = 0$, where

$$Z = 1 + \sum_{1}^{180} \exp\{-\boldsymbol{\alpha}^T \boldsymbol{f}_i(j)\}$$

2.2. Learning model parameters

We find the 5 parameters $\boldsymbol{\alpha}$ in our model by maximum likelihood estimation (MLE). We extract N training patches

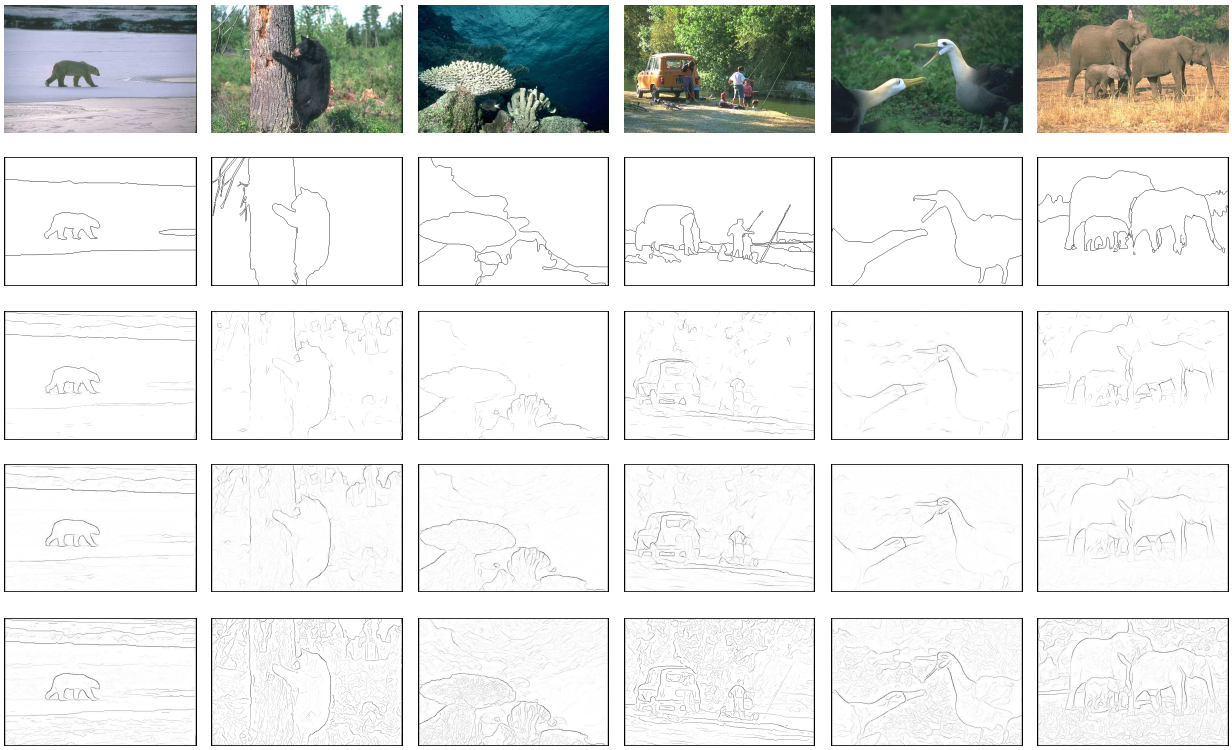


Figure 1. The test images are shown in the first row, and ground truths (randomly selected from one of the annotators) are shown in the second row. Our method (third row) is compared with Pb (fourth row) and mPb (fifth row).

$\{(T_j, P_j\}_{j=1}^N$ from the BSDS500 dataset. Since a training patch may not be well-encoded by only one of the segmentation templates, we softly encode it. That is, we associate each training patch with a probability q_j, where $q_j = (q_{j,0}, q_{j,1}, ..., q_{j,180})$ and $q_{j,i}$ is the probability of the training patch P_j being encoded by i-th segmentation template. We compute the *covering* [1] of the training patch by the i-th segmentation template, and the probability $q_{j,i}$ is proportional to the measure. The likelihood function becomes

$$L(\boldsymbol{\alpha}) = \prod_{j=1}^{N}[\prod_{i=0}^{180} Pr(T_j = i|P_j)^{q_{j,i}}]$$

and the log likelihood function is

$$l(\boldsymbol{\alpha}) = \sum_{j=1}^{N}[\sum_{i=1}^{180} q_{j,i}(-\boldsymbol{\alpha}^T \boldsymbol{f}_i(j) - \log Z) - q_{j,0} \log Z]$$

Taking the derivative of $l(\boldsymbol{\alpha})$ with respect to α_p, the p-th component of $\boldsymbol{\alpha}$, we have

$$\frac{\partial l}{\partial \alpha_p} = \sum_{j=1}^{N}\{\sum_{i=1}^{180} Pr(T_j = i|P_j) f_{i,p}(j) - \sum_{i=1}^{180} q_{j,i} f_{i,p}(j)\}$$

where $f_{i,p}$ is the p-th component of $\boldsymbol{f}_i$.

We then use stochastic gradient ascent to find the maximum likelihood value of the model parameters, which is guaranteed to converge to the global optimum as the log-likelihood function $l(\boldsymbol{\alpha})$ is concave.

Method	ODS	OIS
Pb	0.66	0.68
ours	0.66	0.68
mPb	0.68	0.70

Table 1. Contour detection (F-measure) on BSDS500 for methods based on single-scale cues (Pb [4] and our proposed epitomic shape representation) and multi-scale cues (mPb [1]). ODS: fixed threshold for all images in the data set. OIS: optimal oracle threshold on a per-image basis.

2.3. Experimental results

The work of Arbelaez et al. [1] explores a hierarchy of edge detectors employing increasingly complex features. Their baseline Pb system from [4] uses only single-scale features. Their mPb detector improves upon Pb by fusing features at multiple scales. Their most advanced gPb detector attains state-of-the-art boundary detection performance by also incorporating global spectral clustering information. The current version of our system is more closely comparable to Pb, as it uses the same features and also only employs single-scale cues.

We apply our epitomic edge detector separately on each

17×17 image patch. We retain the top 10 most probable shape templates from the pool of 181 candidates, along with their soft assignment probabilities $Pr(T_j = i|P_j)$ from Eq. (1). Each of the overlapping image patches contributes a term $\sum_i Pr(T_j = i|P_j)[T_j = i]$ in forming our global image-level edge map, where the template's edge indicator $[T_j = i]$ is 1 at the edges of the i-th segmentation template and 0 otherwise. The overlapping templates do not necessarily agree on their decisions, resulting in dispersed boundaries. We employ an extra grayscale morphological thinning step to obtain our final image-level soft edge maps. By thresholding these at different levels, we obtain precision-recall curves for edge detection.

Some qualitative and quantitative results are shown in Fig. 1, and Table. 1, respectively. As shown in the figure, our proposed method is less sensitive to weak edges (e.g., textures on the tree in the second image). The overall performance in terms of F-measure is the same as Pb. Our performance is behind mPb; we anticipate that using cues at multiple scales similarly to mPb can improve the performance of our epitomic shape representation. We leave this for future work.

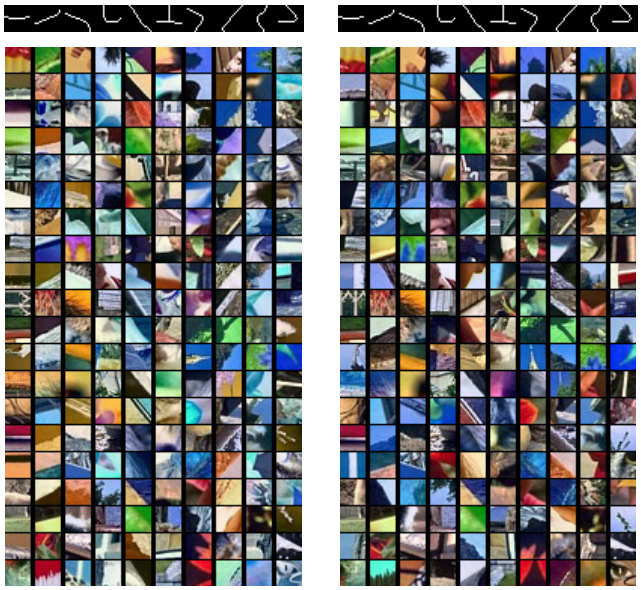
3. Local appearance modeling

In this section we explore how shape epitomes can be employed for the task of image appearance modeling. Our strategy is to model the raw appearance of image patches after first aligning them using shape epitomes. Many authors have shown that separating the effect of shape deformations and texture variability can greatly facilitate appearance modeling. For example, Active Appearance Models [2] compactly represent the appearance of faces by separately modeling shape and texture with low-dimensional PCA models.

Herein we study a similar approach for modeling the appearance of small image patches extracted around edges in the BSDS 500 dataset. We assign each 25×25 patch P into one of our first ten shape epitomes based on their region overlap distance (see main paper), and we also compute the translation (t_x, t_y) and rotation that best aligns each patch to its corresponding shape epitome. This matching step employs the ground-truth shape annotations but does not take appearance information into account. We then extract the color values from the 17×17 part of the patch $P(t_x, t_y)$ aligned with the 17×17 central patch of the epitome. As an additional pre-processing step, we normalize the vector of color values of each patch to have zero mean and unit variance. A subset of the patches belonging to each of the ten epitomic clusters is shown in Fig. 2(a). We also show in Fig. 2(b) the result of the previous procedure when we retain the cluster assignments but skip aligning the patches to the epitomes, thus using $P(0, 0)$ instead of $P(t_x, t_y)$.

We then learn a separate PCA appearance model for the patches that belong to each shape epitome cluster. The underlying assumption is that patch appearance is conditionally Gaussian (and possibly low-dimensional), given the patch edge label. Equivalently, this can be seen as a Gaussian mixture model for the appearance of patches, with the edge label being the latent assignment variable. The leading PCA eigenvectors are shown in Fig. 3, both for the aligned and non-aligned patches. We see that including alignment makes the principal components coherent with the shape of the underlying edge structure.

Some analysis related to the dimensionality of PCA analysis for these data is shown in Fig. 4. We show the number of eigenvectors needed to achieve reconstruction error below some threshold (20% of the data variance). Assigning the appearance patches to different shape-based clusters allows us to approximate the vectorized patch color appearance (the vector length is $17 \times 17 \times 3 = 867$) in compact PCA subspaces of roughly 10-15 dimensions. Retaining the epitome assignment labels but skipping the alignment step requires using on average 15% more components to achieve the same reconstruction error.



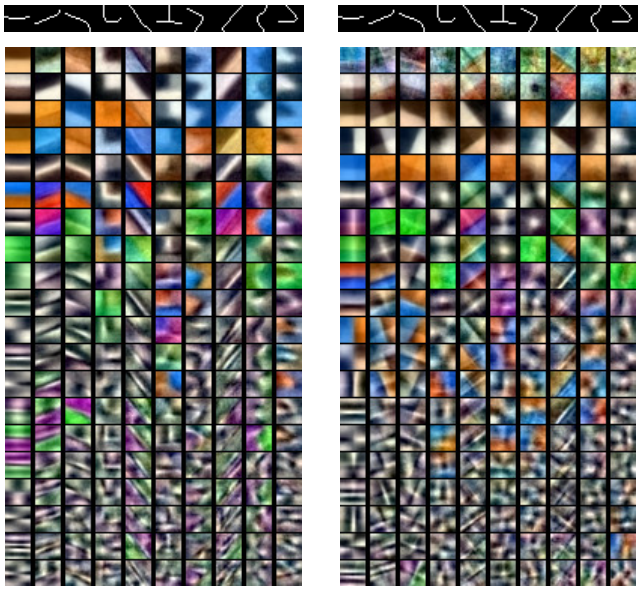
(a) (b)

Figure 2. Image patches corresponding to edge-based clusters (one cluster per column). (a) Aligned to the shape epitome, allowing shift and rotation. (b) Without alignment.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[2] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.



(a) (b)

Figure 3. The principal components of patch appearance variation. Each column corresponds to appearance patches assigned to the same shape epitome. The mean patch is shown at the top, followed by the leading eigenvectors in subsequent rows. (a) When patches are aligned to the epitome. (b) When the patches belong to the same epitome but are not fully aligned (no shift or rotation).

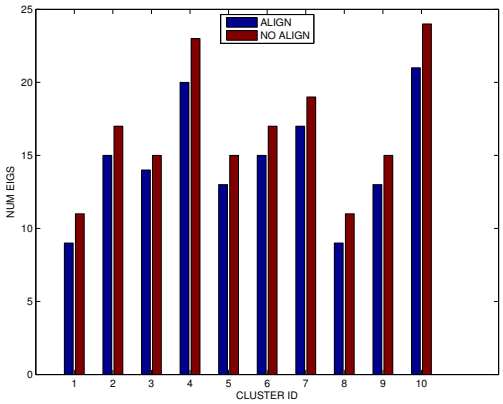


Figure 4. PCA modeling. Number of PCA components needed to explain 80% of the variance for each of the ten shape epitome clusters.

[3] S. Konishi, A. Yuille, J. Coughlan, and S.-C. Zhu. Statistical edge detection: Learning and evaluating edge cues. *PAMI*, 25(1):57–74, 2003.

[4] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.