Supplementary Material for "Shape Discovery from Unlabeled Image Collections"

Yong Jae Lee and Kristen Grauman

1 Sanity Check Baseline Results: Averaging Images per Category

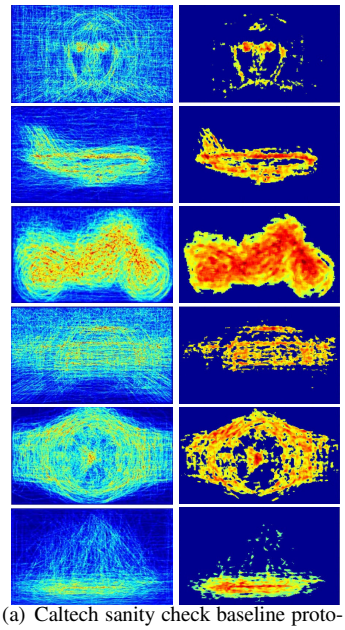
As described in the text of the main paper, this baseline is a sanity check to assure the difficulty of generating prototypical shapes. We manually partition the images into the "ideal" clusters, so that each cluster has 100% purity, and then simply average the aligned edge images, using the confidence weights given by the Pb detector [1]. The sanity check baseline helps to indicate the contribution made by our fragment weighting and prototype formation.

Fig. 1 (a-c) show the prototypical shapes formed by the baseline on the Caltech images, ETHZ bounding box regions, and ETHZ expanded regions, respectively. For the Caltech images, the baseline clearly cannot discover the shape agreement, even though the input clusters were perfect. The baseline does pretty well to discover shape on the ETHZ bounding box regions, which is expected, since those regions are scale-normalized and aligned. The baseline performs worse on the ETHZ expanded regions due to clutter in the images; it discovers shapes with some accuracy for only a couple of categories (Applelogos and Bottles). (To compare against our method's prototypical shapes, see Fig. 6 (b) (Caltech) and Fig. 7 (b,e) (ETHZ bounding box and expanded, respectively) in the main paper.)

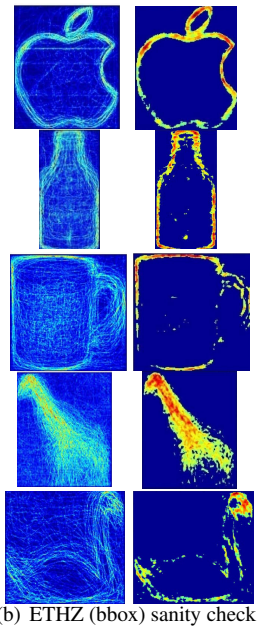
These results confirm that even with perfect clusters, simply stacking the edgemaps will not produce accurate shape models. Our method clearly outperforms this baseline for most of the generated shapes (a few are comparable), even without the advantage of starting with perfect clusters.

2 Description of LabelMe Test Set

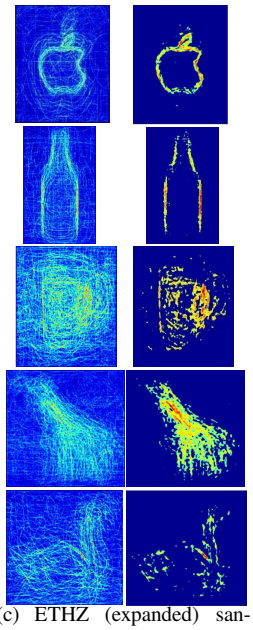
To form the LabelMe test set used in Section 4.3 of the main paper, we downloaded images for each of the Caltech categories with the requirements that (1) there be at least one instance of an annotated object, and (2) there be at least 10K pixels on the object of



(a) Caltech sanity check baseline prototype shapes



(b) ETHZ (bbox) sanity check baseline prototype shapes



(c) ETHZ (expanded) sanity check baseline prototype shapes

Figure 1: Prototypical shapes found by the sanity check baseline on the (a) Caltech images, (b) ETHZ bounding box regions, and (c) ETHZ expanded regions. The right images are thresholded images of the left. (**Best viewed in color.**)

interest. The second requirement is to ensure that edges will be detected on the object of interest. To this end, we created a testset for the Faces (F), Airplanes (A), Cars Rear (C), and Motorbikes (M) categories, each having 15 images. (LabelMe's images for the Watches and Ketches categories did not meet the requirements.)

We perform object detection by matching our prototypical shapes to the test images using a simple modified chamfer distance: we give a penalty to each edgel in the shape mask that is proportional to their weights (vote counts). For this detection task, we are given an image containing an instance of the object, and we have to determine its precise location. To account for scale differences between the models and objects in test images, we search at multiple scales, scaling each test image from half to twice its original size, in increments of 0.1.

We search for a single instance of the object by choosing the region that corresponds to the lowest symmetric chamfer distance. Fig. 2 and Fig. 3 show correct and incorrect detection examples, respectively. (The main paper summarizes the quantitative results.)

Our discovered prototypical shapes lead to some accurate detections even in complex images taken from another dataset. The incorrect detections are mainly due to the limitations of the chamfer matching: it has trouble producing accurate matches when the image has large amounts of clutter, when the object of interest is rotated, and/or

when there are not enough edges detected on the object of interest (due to shadows or bright illumination).

3 Complexity Analysis

We analyze the computational complexity of our method's matching algorithm. Let P and E be the maximum number of patch descriptors and extracted edge fragments in each image, respectively. Let S be the spatial extent (height and width) of the local region in which each fragment in one image can shift to find its best corresponding edge fragment in another.

Computing $D_{patch}(X, Y)$ between feature sets X and Y requires $O(P^2)$ L_2 distance computations. To efficiently compute the shape distances, we pre-compute the distance transform and argument distance transform (both of which can be computed in linear time) on the edgemaps of each image. Computing the coarse shape similarity, $D_{scd}(X, Y)$, requires $O(P^2)$ dot-products between matching features and the shifted distance-transformed image; when the regions overlap, we can memoize distances, which for extremely dense patches reduces the cost to $O(P)$. Once the best-matching pair of regions in X and Y are found, computing the fine shape similarity, $D_{shape}(X, Y)$, requires $O(PES^2)$ dot-products between each edge fragment and the distance-transformed image.

References

[1] D. Martin, C. Fowlkes, and J. Malik. Learning to Detect natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *TPAMI*, 26(5):530–549, May 2004.

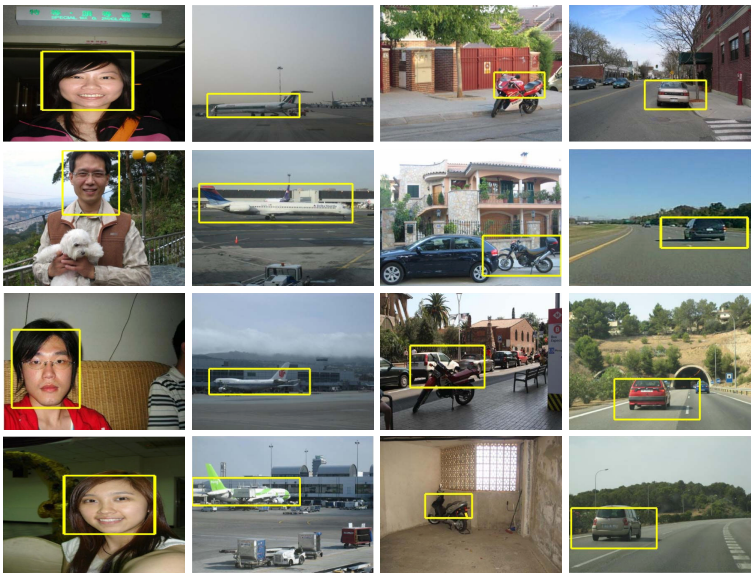


Figure 2: Examples of correct detections. These LabelMe images are scanned with the shape models discovered by our method from the Caltech images; the position and scale yielding the minimal chamfer distance to our shapes is shown with a bounding box.

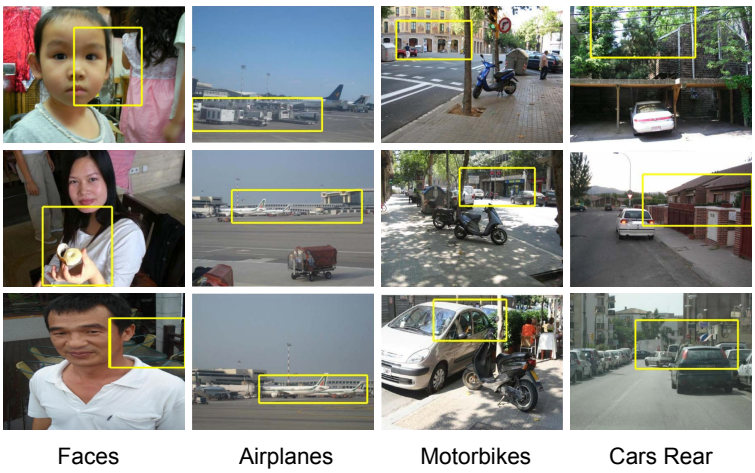


Figure 3: Examples of incorrect detections. These images have large amounts of clutter, object rotation, and/or not enough edges detected on the object that lead to inaccurate chamfer matching.