To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2009.

Shape Discovery from Unlabeled Image Collections

Yong Jae Lee and Kristen Grauman
University of Texas at Austin
yjlee0222@mail.utexas.edu, grauman@cs.utexas.edu

Abstract

Can we discover common object shapes within unlabeled multi-category collections of images? While often a critical cue at the category-level, contour matches can be difficult to isolate reliably from edge clutter—even within labeled images from a known class, let alone unlabeled examples. We propose a shape discovery method in which local appearance (patch) matches serve to anchor the surrounding edge fragments, yielding a more reliable affinity function for images that accounts for both shape and appearance. Spectral clustering from the initial affinities provides candidate object clusters. Then, we compute the within-cluster match patterns to discern foreground edges from clutter, attributing higher weight to edges more likely to belong to a common object. In addition to discovering the object contours in each image, we show how to summarize what is found with prototypical shapes. Our results on benchmark datasets demonstrate the approach can successfully discover shapes from unlabeled images.

1. Introduction

Shape can be a powerful cue for object recognition, due to its invariance to lighting conditions and relative stability compared to intra-category appearance variations. At least for human perception, shape alone can often provide enough information for successful generic object categorization [3]—in fact, some classes are better defined by their shape than their appearance, *e.g.*, bottles, lamps, birds, etc. The success of recently developed shape matching algorithms and advances in shape descriptors [1, 2, 25, 10, 21, 7] are promising signs for using shape to recognize and detect objects. However, current algorithms rely on manually annotated training images to learn the target object shape to be detected in new images. Furthermore, many methods assume access to extracted silhouettes or contour point sets, which are notoriously difficult to pick out from a muddle of broken edge fragments, and are simply not available in unlabeled images of different categories.

In this work we consider the problem of discovering common object shapes within unlabeled, multi-category collections of images. An unsupervised method to discover shapes would be valuable to find interesting objects within unstructured image collections, and eventually to detect those objects in new images.

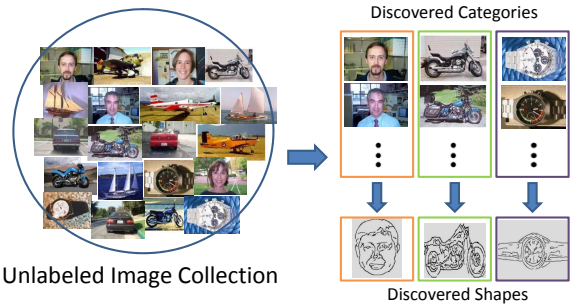


Figure 1. Goal: given unlabeled images, discover common shapes.

Unsupervised methods for object discovery have begun to be explored using distributions of local region features (i.e., bags-of-words or patches) [26, 24, 11, 17, 5, 15, 12]. Their key insight is that the frequently recurring appearance patterns in an image collection will correlate with objects of interest. Such representations are quite reliable for classes defined by repeated textures, but unfortunately by definition are insufficient to capture underlying shape or contours.[1]

What challenges are unique to shape discovery? Some of the most effective known descriptors based on histograms of oriented gradients (e.g. [18]) are purposefully insensitive to local changes. While this provides a (usually desirable) invariance to minor changes in the pixel-level data, the loss in the structure of the underlying gradients means it is generally too coarse to accurately describe contour-level detail.

Similarly, if a patch feature is extracted on an object boundary, the image portions on or off the foreground will contribute equally, which means that many matches will be missed on an object's shape-defining boundaries if it is surrounded by clutter. Interest point detectors can identify distinctive and repeatable regions, but textureless objects will largely lack patches on and/or within their boundaries. At

[1] Throughout, we use *shape* to mean an object's outer and internal contours; we use *appearance* to refer to texture and photometric properties, captured for example with local patch features like SIFT.

the same time, without good context or initialization, an average edge fragment is non-distinct and can match well with all sorts of structures within a cluttered image.

We introduce an algorithm that analyzes a collection of unlabeled images containing multiple categories of objects, and returns both a set of proposed prototypical shape models, as well as a list of edge fragments per input image weighted according to their confidence of belonging to the primary foreground class (see Fig. 1). The main idea is to use local features to anchor the edge fragments that surround them, and to learn which edges to emphasize as foreground based on their joint correspondences across image examples.

Our main contribution is a method to perform unsupervised shape discovery from unlabeled images—to our knowledge, the first approach proposed for this problem. Unlike existing unsupervised patch-based methods, shape discovery has the potential to mine for categories best defined by their overall shape; even for objects with partial textures in common, it stands to extract models that are more complete in their spatial extent. We demonstrate our approach using benchmark datasets and show that linking shape to sparse appearance agreement leads to better unsupervised discovery than when either cue is used alone.

2. Related Work

In this section we briefly review relevant work in unsupervised category learning, foreground segmentation from labeled images, and object detection using edge fragments.

Unsupervised category learning methods can largely be divided into two groups. The first group considers ways to discover latent visual topics using models developed for text, such as pLSA and LDA [23, 6, 24, 17]. The second group of methods treats the task as a hard-assignment clustering problem; graph-based algorithms using spectral clustering [11, 12, 15] and message-passing [5] have shown good results. However, all previous unsupervised approaches work solely with appearance (patch) features, and cannot capture shape. While the authors of [24, 29] first decompose the input images into segments or random partitions, the intent is to increase the specificity of the models learned; neither learns shape or matches examples according to contours.

Weakly-supervised methods can segment out a training image's foreground region in cluttered images, with the assumption that each image has the same single prominent object [28, 13, 27]. Implicitly, this is a form of shape recovery, in that ideally the outer boundary of the object forms the segment. Recent work shows how to learn explicit contour-based models from labeled training images cropped with a bounding box [9, 25]. Our method shares the goal of extracting object-level regions, although we seek shape-defining contours rather than figure segmentation. Unlike any of the above methods, our method is fully unsupervised and does not use labeled exemplars.

A number of methods consider how to simultaneously classify and localize objects. Methods using Hough-style voting with patches [16] or discriminative edge fragments [21] can backproject segmentation boundaries learned from labeled training examples to predict new objects' outlines. The authors of [8] extend the constellation model to include curve parts as well as patches. Our discovered models can be used for localization, but again our framework differs significantly since it forgoes annotated examples.

The proposed approach is the first to address unsupervised shape discovery. While some steps of this task have challenges in common with the methods above, the matching and grouping issues demand new strategies once we have jumbles of edge fragments and no prior knowledge about which images ought to have some corresponding features.

3. Approach

The goal is to identify which foreground contours in each image can form high quality clusters, and use any intra-cluster agreement to discover the underlying prototypical shapes. We expect the discovered shapes to often be representative of object categories. Since edge features often lack distinctiveness, we use patch matches to initialize regions for shape matching. The intuition is that if two local features are a good match in terms of appearance and describe the same object part, their surrounding regions may have similar contours (with some local shifts and deformations). We define an affinity function to cluster images based on these matches, and then infer a weight per edge fragment based on how consistently it matches other intra-cluster images. Finally, a voting-based step computes prototype summaries of the discovered shapes.

The upshot of our combined feature matching is twofold: first, we are able to eliminate many spurious matches that would occur if either feature were to be used independently, and second, we expand the coverage of object-to-object matches past their sparse repeated textures to include their neighborhood contours (see Fig. 2). In the following, we describe the details of our representation, how to distinguish foreground edges from clutter, and how to build a prototype shape from the estimated foreground contours.

3.1. Anchoring Edge Fragments to Local Patches

We represent an unlabeled image as a set of semi-local region features, $X = \{f_1, \ldots, f_{|X|}\}$, where each f_i consists of a local appearance descriptor and all the surrounding edge fragments and their weights. Specifically, $f_i = \{p_i, \langle e_{i,1}, w_{i,1}\rangle, \ldots, \langle e_{i,l}, w_{i,l}\rangle\}$, where p_i denotes a patch

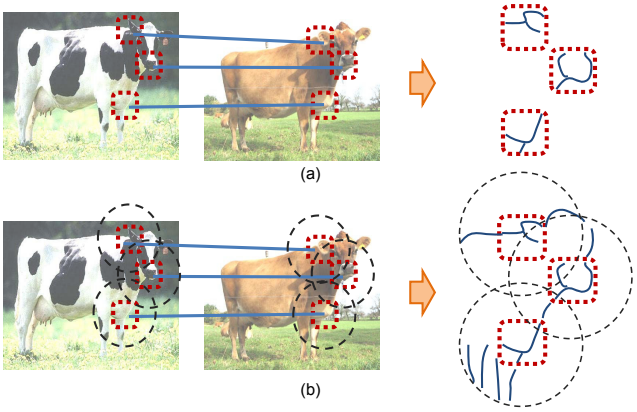


Figure 2. Two images, each with three detected patch matches. **(a)** There is a limit to how much shape information can be captured even with accurate patch matches, yet edge fragments can often be ambiguous to match in cluttered images. **(b)** By anchoring edge fragments to patch features, we can select the fragments that agree and describe the object's shape.

descriptor, each $e_{i,m}$ is an associated edge fragment, and l denotes the total number of edge fragments in the image. Each edge weight $w_{i,m} \geq 0$ reflects the emphasis given to that fragment when computing shape matches with the combined representation (the details of which will be defined below). Note that each patch maintains a full set of weights on all l of the image's edge fragments. Thus each fragment has a weight from the "point of view" of a given local appearance region, and is part of the combined feature representation exactly $|X|$ times. We extract edge fragments (smooth segments of chained edgels) using [10], and use the SIFT descriptor [18] to represent patches.

The motivation for this integrated representation is as follows. When comparing two images, we would like to use matched edges to determine whether they share a shape, and thus should be clustered together. However, many edge fragments are very generic and can produce spurious matches, which in turn result in unreliable similarity scores. (For example, an edge fragment extracted from the roof of a car could match well to the top of a monitor.) While this ambiguity is also an issue for weakly supervised algorithms, it is amplified when we lack image labels: for any two images, there is no guarantee whether some of their contours should agree or not.

By anchoring the edge fragments to patch descriptors, we can produce more reliable matches. A detected patch match serves to initialize the spatial placement of the surrounding edges from one image to the next. If the patch descriptors have good matches *and* describe the same object part, then some subset of their nearest associated edge fragments should also match well (see Fig. 2).

Which edge fragments should a given patch anchor most strongly? When working with unlabeled images, we do not know the spatial extent of the foreground region. Between this and the unknown clutter, we cannot immediately determine which edge fragments surrounding a given patch would produce meaningful (foreground-related) matches. Initially, we account for this uncertainty by imposing a Gaussian weighting a priori for all $w_{i,m}$ based on the spatial proximity of fragment $e_{i,m}$ to the patch center p_i. The width σ_i of each 2d Gaussian is set relative to the patch's scale—specifically, as three times its semi-major axis. Thus, closer edges are weighted higher, from the point of view of that particular patch. This reflects that at first we do not know which fragments are relevant versus clutter, but expect better shape agreement (if any) to be found near places where we find good appearance agreement. The edge weights are later updated based on cumulative matching results (see Sec. 3.3).

3.2. Grouping Cluttered Images with Similar Shapes

In order to discover common shapes, we first need to form fairly homogeneous groups from the image collection such that each group contains a number of images with similar foregrounds. To do this, we use spectral clustering with an affinity function that reflects the strongest shape and appearance correspondences found within two images. Assuming that frequently recurring objects have some repeated visual content, this stage will tend to group images containing the same category. Note that since each image is assigned to one cluster, our method discovers objects from one primary category of interest per image.

The cluster quality will depend heavily on the way affinities are measured. We design a new similarity function between feature sets X and Y that uses a two-step procedure to target possible agreement between contours amidst clutter. We first compute region-based edge matches; the local layout of fragments is more distinctive than are individual fragments, and can produce a more reliable but coarse assignment. Given a matching region, we then compare its individual fragments, refining the match to discern foreground edges from background edges (see Fig. 3).

In the **first step**, we find corresponding regions: for a given feature in X, we find the best matching feature in Y according to both an appearance-based distance and a coarse-shape distance (see Fig. 3(a)). Specifically, for each feature pair $f_{i,X}$, $f_{j,Y}$ we compute:[2]

- the *patch distance*, which is the L_2 distance between the descriptors: $d_{patch}(f_{i,X}, f_{j,Y}) = ||p_{i,X}, p_{j,Y}||_2$.
- the *coarse shape distance*, as measured by the symmetric chamfer distance, denoted d_{scd}. It is coarse in that we initially perform no shifting or local search,

[2]Here $f_{i,X}$ denotes the i-th feature within set X.

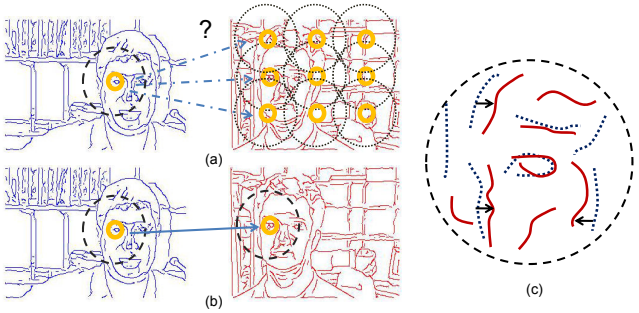


Figure 3. **(a)** A feature from image X and all features from image Y. **(b)** The best matching feature in Y is chosen, based on local appearance and coarse surrounding shape. **(c)** X's edgemap is aligned with Y's edgemap at the match point, and each fragment in X is fitted to the nearest best matching fragment in Y. Dotted circles represent the initial Gaussian weighting on the fragments.

and consider only inter-edgel distances, not orientations. The distance term for each edgel in the fragment is weighted (for now) by its Gaussian weighted distance from its anchor patch.

For each feature $f_{i,X}$ in X, we then choose the best matching feature $f_{j(i)^*,Y}$ in Y, where:

$$j(i)^* = \underset{1\le j\le |Y|}{\operatorname{argmin}} \Big(d_{patch}(f_{i,X}, f_{j,Y}) + d_{scd}(f_{i,X}, f_{j,Y}) \Big).$$

This matching can be many-to-one if multiple features in X have a good match with the same feature in Y.

In the **second step**, we place the edge image for X onto the edge image for Y according to the position and scale of a matched patch, for each match in turn.[3] Essentially, the distances from the first step determine a candidate rough alignment for each $f_{i,X}$ within Y (see Fig. 3(b)). Then, given the aligned images (centered at the positions of $f_{i,X}$ and $f_{j(i)^*,Y}$, respectively), we can more precisely evaluate the agreement of each edge fragment in X to some edge in Y.

Even if the two matching regions contain the same object parts, in general, we can expect there to be some differences in shape. Thus, for each edge fragment $e_{i,m}$ in the shifted version of X, we independently find its best-matching nearby edge fragment $e_{j(i)^*,n}$ in Y with the Oriented Chamfer Distance (OCD) [25], which is sensitive both to nearness in space as well as the gradient orientation (see Fig. 3(c)). Candidate fragments from Y are those within a local window relative to $e_{i,m}$'s initial placement in the matching region. The total shape distance from feature $f_{i,X}$ to feature $f_{j(i)^*,Y}$ is the weighted average of all the best edge fragment distances:

$$d_{shape}(f_{i,X}, f_{j(i)^*,Y}) = \frac{1}{l} \sum_{m=1}^{l} w_{i,m}\, d_{OCD}(e_{i,m}, e_{j(i)^*,n(m)^*}),$$

where l is the number of edges in X's image, and subscript $n(m)^*$ denotes the index of the best match for fragment m. We normalize d_{patch} and d_{shape} to be in $[0, 1]$.

This gives us the feature-to-feature cost. The overall directed patch and shape distance from image X to image Y is the average over the component feature distances between each $f_{i,X}$ in X and its best matching feature $f_{j(i)^*,Y}$ in Y:

$$D_{patch}(X,Y) = \frac{1}{|X|} \sum_{i=1}^{|X|} d_{patch}(f_{i,X}, f_{j(i)^*,Y}), \text{ and}$$

$$D_{shape}(X,Y) = \frac{1}{|X|} \sum_{i=1}^{|X|} d_{shape}(f_{i,X}, f_{j(i)^*,Y}).$$

Since the matching is many-to-one, the cost of matching X to Y is not necessarily equivalent to the cost of matching Y to X. We obtain a symmetric cost via the sum: $D'(X,Y) = D(X,Y) + D(Y,X) = D'(Y,X)$.

Given the distances between all pairs of the N unlabeled images, we form an N x N affinity matrix A, where

$$A_{r,s} = \exp\left(-\frac{1}{\sigma^2} D'_{patch}(X_r, X_s) * D'_{shape}(X_r, X_s) \right),$$

for all $r, s = 1, \ldots, N$. We take the product of the costs to reward most those images that have high matching scores in terms of both cues. For each node in A, we retain the top $10 \log(N)$ largest values (as in [12]) in order to form a sparser affinity matrix. This affinity matrix is the input to spectral clustering, which groups the images; we use the method of [20].

3.3. Inferring Foreground Contours

Next we analyze the pattern of the intra-cluster edge matches. Even within the best image-to-image matches, some fragments are actually irrelevant to the common object. For example, two images containing cows may happen to have similar spots on their backs, while others have none. Part of the shape discovery phase must be to emphasize those contours that repeatedly match the same things in all intra-cluster images. To do this, we identify fragments with the most consistent correspondences, and increase their weights.

Specifically, to update edge fragment weight $w_{i,m}$ within feature f_i of some image, we compute the median of its best match distances across all other images within the cluster:

$$w_{i,m} = \exp(-\frac{1}{\sigma_w^2} Z_{i,m}),$$

where $Z_{i,m} = \text{median}_k(d_{patch}(f_i, f_{j(i)^*,Y_k}) + d_{shape}(f_i, f_{j(i)^*,Y_k}) + d_{OCD}(e_{i,m}, e_{j(i)^*,n(m)^*}))$, Y_k is the k-th image within the cluster, and as before $j(i)^*$ indexes the best region match, and $n(m)^*$ indexes the best fragment match when aligned according to that region. Thus we weight the contribution of an edge fragment by the

[3] Our current implementation aligns the regions for scale and position; one could additionally add rotation invariance by rotating the edgemaps according to the patch's dominant gradient direction.

combined matching score of its individual match as well as its region match. The purpose of the median is to ensure that high weight goes only to those edge fragments that produce low matching costs against most cluster members (versus a very low cost against a few). We compute a single weight for each fragment by averaging the fragment's weights across all the features that contain it. At this point we have gone from the input set of unlabeled images, to an output estimating each contour's strength within each image, based on the common shapes that have been discovered (see Fig. 6 (a) and Fig. 7 (a,d) for examples).

3.4. Prototypical Shape Formation

Now that we have found the common foreground contours for each image, we can generalize these shapes to produce a prototype summarizing each cluster. There are two important considerations: not all images in a cluster will necessarily contain an object of the same category, and not all objects of the same category agree in terms of shape anyhow. We handle these issues by creating a simple vote space based on the discovered edge weights, such that the common shape of the object can be reinforced in the output, while parts that agree less can be discarded.

Using each cluster's center image as a target, we match all other within-cluster images to it using a modified chamfer distance, where each edgel's matching cost is penalized according to its weight. This way, higher weighted (most confident) fragments have more influence in the match. Once aligned, we accumulate the weighted fragments as votes, for all images in the group (see Fig. 6 (b) and Fig. 7 (b,e) for examples). The chamfer distance gives us a straightforward way to coordinate the foreground contours; more elaborate shape matching algorithms (e.g., allowing deformations) could also be used in this step and may make the alignment even more robust.

4. Experiments

We present results to analyze our method's unsupervised category and shape discovery. We work with images from the Caltech-101 [4], ETHZ shape [10], and LabelMe [14] datasets. The only supervised information is the number of categories.

Implementation Details: We use the Berkeley edge detector [19], from which we extract fragments using [10]. To reduce the number of chamfer comparisons when matching regions, we only compute d_{scd} for regions we already know have good patch matches (in practice, the top 5%). To extract patch features, we densely sample SIFT descriptors at every 10 pixels in the image, using small patches with a radius of 8 pixels. We set $\sigma = \sigma_w = 0.15$.

Datasets: We first test with the Caltech dataset since all previous unsupervised methods have chosen to test with it. We use the same categories as [12]: *Faces, Airplane, Motorbikes, Cars Rear, Watches, Ketches*. We compare against the state-of-the-art methods of [12, 11, 15] because they share our goal of discovering categories and selecting foreground features based on commonly reoccurring features.

Since not all the Caltech categories have characteristic shape, we also experiment with the ETHZ shape dataset, which consists only of objects well-defined by their shape. The categories are: *Applelogos, Bottles, Giraffes, Mugs, Swans*. This dataset was used in [9] to learn a shape model for each category using the labeled ground-truth bounding box regions. We experiment with both (1) those same bounding box regions and (2) expanded regions that enclose the bounding box (at four times the initial bounding box area) to learn our models. Following [9], we normalize to maintain the average aspect ratio over all category instances. Unlike in [9], our algorithm learns five shape models at once over the entire dataset without knowing the class labels of the images.

Evaluation Metrics: We use *purity* to evaluate our method's object category discovery. Purity measures the extent to which a cluster contains images of a single dominant class. Since the datasets have ground truth class labels, this allows us to quantify the quality of the groups we learn.

Since our method discovers the outer *and* internal object contours, we quantify the extent to which the shapes we identify per image agree with the true foreground region using the *Bounding Box Hit Rate* (BBHR) [22]. The BBHR measures the percentage of images in the dataset that have at least h foreground features selected, as a function of the selection threshold applied to the feature weights. It is recorded with respect to the False Positive Rate (FPR), which counts the average number of selected features falling outside of the bounding box. If our shape discovery performs well, we expect more coverage of the object from the discovered features than with patch matches alone, since the agreement between patches will generally be sparser than with our anchored edge fragments (even though patches densely cover the image).

4.1. Unsupervised Category Discovery

To measure category discovery on the Caltech categories, we follow the same experimental setup proposed in [11]. In Table 1 (top), we compare the mean purity obtained by our method to [12, 11, 15]. The results show that our method is comparable or better than related methods. Upon inspection, we found that most of the misclassified examples are images that do not have many edges detected on the foreground (due to shadows or bright illumination), or objects that do not follow the general shape of the other objects in its category.

In Table 1 (bottom), we show our method's mean purity on the ETHZ dataset. The first row shows results ob-

CT-Categories	Our Method	Patch-only	[12]	[11]	[15]
A,C,F,M	98.03 ± 0.66	87.37	98.55	86.00	88.82
A,C,F,M,W	96.92 ± 0.63	83.78	97.30	N/A	N/A
A,C,F,M,W,K	96.15 ± 0.52	83.53	95.42	N/A	N/A

ETHZ-Categories	Our Method	Patch-only
A,B,G,M,S (bbox)	95.85	78.89
A,B,G,M,S (expanded)	76.47	61.25

Table 1. Category discovery accuracies measured by mean purity for the categories of the Caltech [A: Airplanes, C: Cars, M: Motorbikes, W: Watches, K: Ketches] (top) and ETHZ [A: Applelogos, B: Bottles, G: Giraffes, M: Mugs, S: Swans] (bottom) datasets.

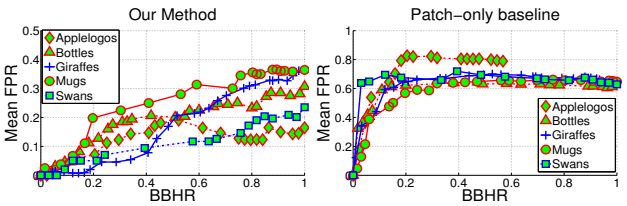


Figure 4. Bounding box hit rates (BBHR) vs. mean false positive rates (FPR) for the expanded ETHZ regions. Lower curves are better. We compare results using our patch-anchored shape matching (left) with a baseline using patches only (right).

tained using only bounding box regions, and the second row shows the expanded region results. The decrease in accuracy on the expanded region images is mainly due to the large amount of clutter that is included in those regions. Still, overall the purity rates are high, such that accurate contours can be learned per group.

We also compare against a patch-only baseline, in which we use the same steps as our method, but use only patch features (without shape information). Our method significantly outperforms this baseline on both datasets.

4.2. Foreground Shape Discovery

Foreground Localization: We next evaluate our method's foreground localization. We compute a single weight for each feature by averaging its edge fragment weights and consider a "hit" if the selected feature's center is within the object's bounding box. We use $h = 5$ and take the top 20% of the highest weighted features in each image, following [12]. To evaluate how much our patch-anchored edge fragments contribute to foreground discovery, we again test against the patch-only baseline.

The BBHR-FPR curves are shown in Fig. 4 (ETHZ) and Fig. 5 (Caltech). Our patch-anchored shape matching significantly outperforms the baseline using only patch features (note the FPR axes range difference). The reason is twofold: (1) we can form purer clusters than patches alone; incorrectly clustered examples will often have the highest weighted features on the background, and (2) shape information leads to more accurate matching, especially for objects that have less local appearance agreement as in the ETHZ images.

We also achieve better localization on the Caltech dataset than the unsupervised baseline [12], an appearance-based approach that also uses spectral clustering. The comparable levels of purity by our approach and [12] (see Table 1 (top)) suggest that background contextual features may have contributed to its accuracy. By considering shape information, our method focuses on the object such that more foreground features are given highest weight. In Fig. 6 (a) and Fig. 7 (a,d), we show the highest weighted edge fragments for example images of each shape discovered by our method.

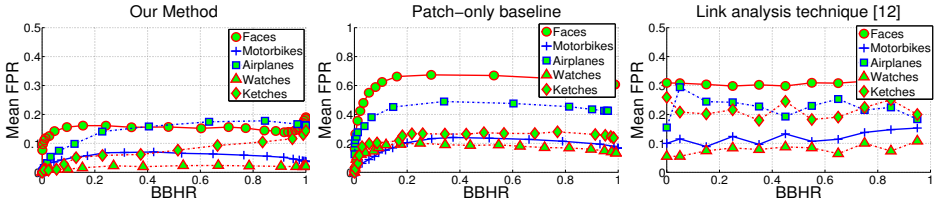
Prototypical Shape: We generate prototypical shapes (as explained in Sec. 3.4) for each of the shapes found by our method on the Caltech and ETHZ datasets. Fig. 6 (b) and Fig. 7 (b,e) show the results.

We compare our shape discovery to two baselines: first, a shape-only baseline in which all edges are weighted equally when computing image similarities. We cluster the images with an affinity matrix computed from the symmetric chamfer distance between their edgemaps. Once the clusters are formed, the prototypical shape is computed in the same manner as our method. This shape-only baseline is intended to give a sense of the degree of ambiguity when matching cluttered edge images. The second baseline is a sanity check to assure the difficulty of the task: 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 [19]. This baseline will indicate the contribution made by our fragment weighting and prototype formation (see Supplementary Material for this result).

Figure 6 (c) shows the prototype shapes found for the Caltech dataset by the shape-only baseline. It discovers two Motorbike shapes, one Watch shape, and three that do not clearly belong to any category. This is due to inaccurate matches that lead to heterogeneous clusters: the mean purity is only 55.67%. Among the clusters that do have relative homogeneity are two comprised mainly of Motorbikes, and one comprised mainly of Watches. This is reasonable, since most of the Motorbike and Watch images have little background clutter and similar shapes throughout.

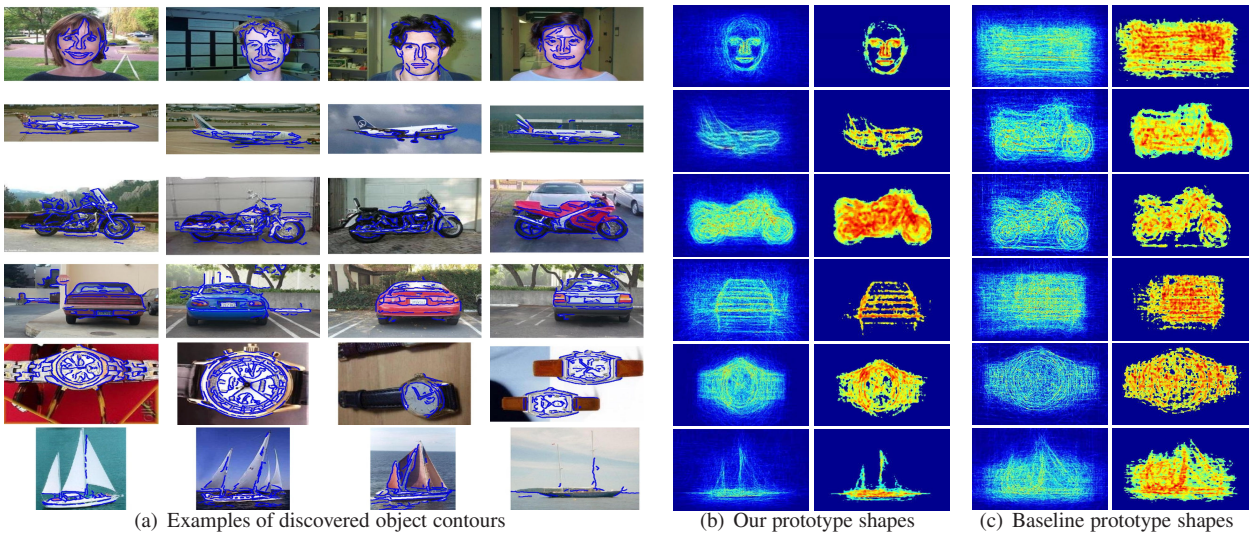
The prototypical shapes found by our method fairly accurately describe the shapes of the dominant objects. Most background clutter fragments have been removed. We not only discover the boundary contours, but also find some inner contours that are unique to each object (e.g., eyes, nose, and mouth for Faces). We also inevitably discover repeated curves that do not actually belong to the object (e.g., the pavement line for Cars Rear, and the horizon for Ketches), which makes sense, since they too are reoccurring.

The prototype shapes found for the ETHZ data by our method and the shape-only baseline are shown in Fig. 7 (b,e) and (c,f), respectively. Again these results show that our method does well to discover shapes illustrating the



	Our Method	Patch-only	[12]
F	0.15	0.67	0.30
M	0.07	0.24	0.11
A	0.17	0.48	0.21
W	0.03	0.19	0.08
K	0.08	0.27	0.19

Figure 5. Bounding box hit rates (BBHR) vs. mean false positive rates (FPR) for the Caltech images. Lower curves are better. We compare results using our patch-anchored shape matching (left), with a baseline using patches only (center), and with those obtained by [12] (right). The table summarizes the approximate FPR at BBHR=0.5 for the three methods.



(a) Examples of discovered object contours (b) Our prototype shapes (c) Baseline prototype shapes

Figure 6. (a) Examples of Caltech images, with our method's most confident discovered contours overlayed. (b) Prototypical shapes found by our method. (c) Prototypical shapes found by the shape-only baseline method (see text for details). Our method produces prototypical shapes that accurately illustrate the common objects. The baseline method only produces three such shapes (one of them twice) with a lot more noise. For (b,c), the right images are thresholded images of the left. (**Best viewed in color.**)

common objects. Most background clutter is removed and foreground fragments are emphasized. An exception is the Mug prototype for the expanded region clusters. This may be due to the low purity rate for that cluster: 63.34% compared to [A: 77.27%, B: 87.27%, G: 76.92%, S: 81.82%]. Many non-Mug edge fragments contributed to the prototype shape formation. The baseline shows much worse results, again due to inconsistent feature matches that result in heterogeneous clusters: mean purity is 63.32% and 52.94% for the bounding box and expanded regions, respectively.

For the bounding box regions, the baseline discovers three shapes that resemble Giraffes (along with an Applelogo and a Bottle shape). This is reasonable considering that 91 of 289 regions are Giraffes, which also have the most textured regions among the categories (leading to false chamfer matches). For the expanded regions, the shape-only baseline falls apart completely: only one of the discovered shapes resembles an object (a Giraffe).

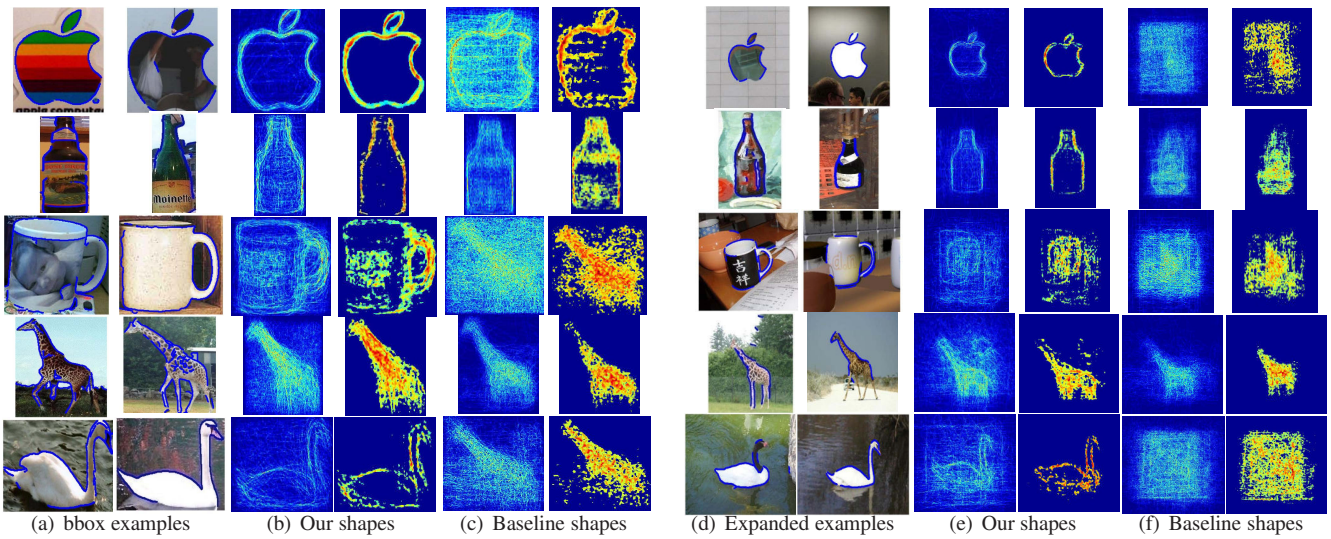
4.3. Generalization to Detection in Novel Images

Finally, we test the generality of our method's discovered shapes by using them to perform a detection task on images from the LabelMe dataset [14]. While all previous unsupervised category discovery methods have been evaluated only on partitions of the same prepared datasets from which they were learned, this seems like a good challenge to assure that what was discovered is not purely due to peculiarities of the dataset.

We created a testset for the Faces (F), Airplane (A), Cars Rear (C), and Motorbike (M) categories, each having 15 images (see Supplementary Material for details).

We perform object detection by matching our prototypical shapes to the test images. We measure detection accuracy by the area overlap over the combined area of the ground-truth bounding box and the detector's output bounding box: $a_0 = (BB_{gt} \cap BB_d)/(BB_{gt} \cup BB_d)$. The average a_o for each category is: [F: 0.47, A: 0.43, M: 0.38, C: 0.31]. Chance detection would be: [F: 0.03, A: 0.02, M: 0.03, C: 0.02]. Even with a weak chamfer matching detector, our discovered prototypical shapes serve as good templates to detect objects in novel images.

Conclusions: We have developed an algorithm to discover common object shapes in unlabeled images. We have shown the strength of our patch-anchored shape matching



(a) bbox examples (b) Our shapes (c) Baseline shapes (d) Expanded examples (e) Our shapes (f) Baseline shapes

Figure 7. Results on the bounding box regions (a-c) and expanded regions (d-f) of the ETHZ dataset. (a,d): Example images with our method's most confident discovered contours overlayed. (b,e): Prototypical shapes found by our method. (c,f): Prototypical shapes found by the shape-only baseline. For (b,c,e,f), the right images are thresholded images of the left. (**Best viewed in color.**)

by comparing against baseline methods that use each feature in isolation, as well as against previous unsupervised learners.

Acknowledgements: We would like to thank Gunhee Kim for sharing experimental results. This research was supported in part by NSF CAREER 0747356, Microsoft Research, Texas Higher Education Coordinating Board award 003658-01-40-2007, the DARPA VIRAT program, NSF EIA-0303609, and the Henry Luce Foundation.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape Context: A New Descriptor for Shape Matching and Object Recognition. In *NIPS*, 2000.

[2] A. Berg, T. Berg, and J. Malik. Shape Matching and Object Recognition Low Distortion Correspondences. In *CVPR*, June 2005.

[3] I. Biederman and G. Ju. Surface vs. Edge-Based Determinants of Visual Recognition. *Cognitive Psychology*, 20:38–64, 1988.

[4] Caltech 101 Image Database, L. Fei-Fei, R. Fergus, and P. Perona.

[5] D. Dueck and B. Frey. Non-metric Affinity Propagation for Unsupervised Image Categorization. In *ICCV*, 2007.

[6] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *CVPR*, 2005.

[7] P. Felzenszwalb and J. Schwartz. Hierarchical Matching of Deformable Shapes. In *CVPR*, 2007.

[8] R. Fergus, P. Perona, and A. Zisserman. A Visual Category Filter for Google Images. In *ECCV*, 2004.

[9] V. Ferrari, F. Jurie, and C. Schmid. Accurate Object Detection with Deformable Shape Models Learnt from Images. In *CVPR*, 2007.

[10] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object Detection by Contour Segment Networks. In *ECCV*, 2006.

[11] K. Grauman and T. Darrell. Unsupervised Learning of Categories from Sets of Partially Matching Image Features. In *CVPR*, 2006.

[12] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised Modeling of Object Categories Using Link Analysis Techniques. In *CVPR*, 2008.

[13] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[14] Labelme: the open annotation tool. *http://labelme.csail.mit.edu/*.

[15] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning From Partially Matching Images. In *BMVC*, 2008.

[16] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *Wkshp on Statistical Learning in Computer Vision*, 2004.

[17] D. Liu and T. Chen. Unsupervised Image Categorization and Object Localization using Topic Models and Correspondences between Images. In *ICCV*, 2007.

[18] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004.

[19] 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.

[20] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *NIPS*, 2001.

[21] A. Opelt, A. Pinz, and A. Zisserman. A Boundary-Fragment-model for Object Detection. In *ECCV*, 2006.

[22] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient Mining of Frequent and Distinctive Feature Configurations. In *ICCV*, 2007.

[23] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. V. Gool. Modeling Scenes with Local Descriptors and Latent Aspects. In *ICCV*, 2005.

[24] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006.

[25] J. Shotton, A. Blake, and R. Cipolla. Multi-Scale Categorical Object Recognition Using Contour Fragments. *TPAMI*, 30(7), 2008.

[26] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Object Categories in Image Collections. In *ICCV*, 2005.

[27] S. Todorovic and N. Ahuja. Extracting Subimages of an Unknown Category from a Set of Images. In *CVPR*, 2006.

[28] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 2005.

[29] J. Yuan and Y. Wu. Spatial Random Partition for Common Visual Pattern Discovery. In *ICCV*, 2007.

See Supplementary Material `www.ece.utexas.edu/~ylee/shapediscovery/supp.pdf` for complexity analysis, prototypical shape second baseline results, and details on the LabelMe results.