Automatic Learning and Extraction of Multi-Local Features

Oscar Danielsson, Stefan Carlsson and Josephine Sullivan
School of Computer Science and Communications
Royal Inst. of Technology, Stockholm, Sweden
{osda02,stefanc,sullivan}@csc.kth.se

Abstract

In this paper we introduce a new kind of feature - the multi-local feature*, so named as each one is a collection of local features, such as oriented edgels, in a very specific spatial arrangement. A* multi-local feature *has the ability to capture underlying constant shape properties of exemplars from an object class. Thus it is particularly suited to representing and detecting visual classes that lack distinctive local structures and are mainly defined by their global shape. We present algorithms to automatically learn an ensemble of these features to represent an object class from weakly labelled training images of that class, as well as procedures to detect these features efficiently in novel images. The power of* multi-local features *is demonstrated by using the ensemble in a simple voting scheme to perform object category detection on a standard database. Despite its simplicity, this scheme yields detection rates matching state-of-the-art object detection systems.*

1. Introduction

Features for representation and detection of visual classes should ideally have several properties that, in general, are mutually exclusive. These properties include ease of computation, robustness against occlusions and clutter, generality over the visual class considered and discriminability against other classes. Ease of computation, robustness and generality are achieved by simple local features, but local features that are shared among exemplars in a class tend to be too simplistic to have any discriminative power. Discriminability requires features of higher complexity. For most visual classes, automatic extraction of complex local features from exemplars is a difficult problem due to large intra-class variation. In this work we describe a method for extracting and using constellations of simple local features that retain the advantages of simple local features while having more discriminative power. We call these constellations *multi-local features*.

Consider the exemplars of chairs in figure 1. The intra-class variation of the circled corresponding local features is too large to allow a compact representation. The same applies to the global shape of the chairs which would prevent any simple generic shape template to be computed from these exemplars. However, there is obviously some structure that is shared among these exemplars that represents the shape constancy of the class of chairs. This is exemplified by the sets of oriented edges that have been marked in each exemplar. Each oriented edge by itself has very small discriminative power but occurs frequently among the examples in a class. As we add more locations of simple oriented edges, the discriminative power will increase. The challenge is then to add locations such that the generality of the multi-local feature is retained while making it discriminative against other visual classes. This provides us with a simple mechanism to exchange generality for discriminability while still keeping the robustness properties of local features. In the limit as we add more locations to the multi-local feature we will end up with a global shape template. This represents the opposite end of the spectrum where each template will be very discriminative at the price of being very specific for a particular example in the class. This situation is schematically illustrated in figure 2.

Multi-local features have similar properties to spatial constellation models based on local features or patches [8, 19, 5]. The information represented by a constellation model is divided between the local features and the spatial constellation structure. Typically, constellation models are used to represent classes that have discriminative local features that are shared across the exemplars of the class. Thus, most of the information is contained in the local features. The information in the multi-local features is dominated by the spatial constellation of the local features since the local features by themselves are not very discriminative. This makes them ideal for recognition of categories that lack distinctive local features but are instead characterized by their global shape properties. However, multi-local features could potentially be designed based on more informative local features as in standard constellation models or pictorial structures [4].

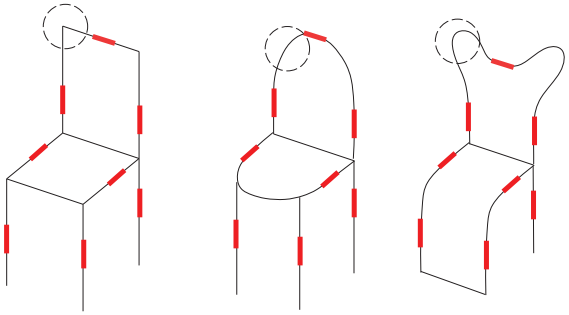


Figure 1. *Local features (circled) can display very large variation within a class, while multi local features (sets of oriented edgels) can capture the shape constancy properties of the class.*

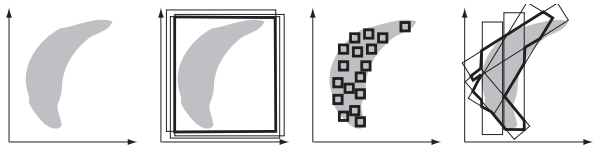


Figure 2. *Modeling a visual category. a) The target category is a set (gray) in some space. b) Using simple features: each feature occurs on many exemplars within and outside of the target category. c) Using complex features: each feature is essentially a template and represents a single exemplar [10]. d) Using intermediate features: each feature is discriminative for the category but not specific for a particular exemplar. This is the case for multi-local features that will be shared among exemplars in the class and each exemplar is represented by a set of multi-local features.*

The use of multi-local features for object representation and detection is analogous to the use of discriminative local features where they are applicable. Therefore we give a brief review of part-based object detection. To detect constellation models one normally begins by detecting parts and then looks for plausible configurations of these parts. The computational complexity of this search for plausible configurations grows quickly with the number of parts in the model and the number of detections of each part in the test image [19, 5]. But recently many researchers have experimented with various voting schemes, where the location of the object is stored relative to each part and each part detection then votes for one or many object occurrences [12, 11, 18, 15]. A common choice is to vote for the centroid of the object (sometimes referred to as centroid boosting). Parts that vote for the same (or similar) centroid agree on the location of the object and are thus likely to form a plausible configuration. Essentially, the explicit search for plausible configurations has thus been bypassed and replaced by a simple voting/clustering procedure. Each part detector acts as a weak detector of one or many object categories, generating hypotheses of object occurrences. The subsequent clustering step determines which hypotheses have sufficient support. Various types of parts have been used, ranging from image patches [12] to edge segments [18, 15].

This bottom-up approach to object detection is very appealing. We would like to leverage the power of part-based models and voting schemes for description and detection using multi-local features. We present a learning algorithm that, given example images of the target category annotated with bounding box information, learns such features and we wrap this learner in a boosting scheme to produce feature ensembles. We also describe an algorithm that efficiently detects these features in test images. We present a performance evaluation on the task of object detection, showing results that match comparable methods.

In summary, this paper presents:

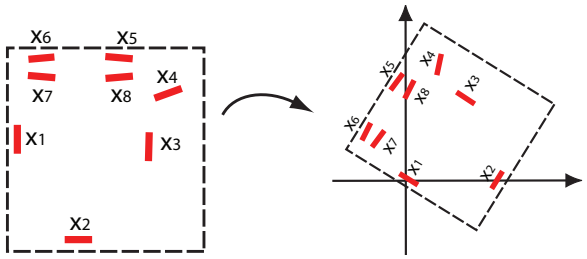
1. A definition of multi-local features.
2. A procedure for automatically learning multi-local features from a set of image exemplars of a class and a computationally efficient way of detecting multi-local features in test images and integrating information from collections of multi-local features for object detection.
3. A demonstration of the usefulness of these features when applied to the problem of object detection.

This paper is structured as follows. In the next subsection, we review some related work. In section 2, we describe the multi-local feature. In sections 3 and 4 we present the algorithms to learn and detect these features. We present an experimental evaluation in sections 5 and 6. Finally, we discuss some properties of multi-local features and suggest some topics for future study in section 7.

1.1. Related Work

Voting schemes are inspired by the Hough transform [1]. Leibe and Schiele applied it to object detection [12], using local appearance patches to vote for the centroid of the target object. Local shape features and boundary fragments have been used for object categories lacking discriminative local appearance [11, 18, 15].

Multi-local features capture global shape properties of the object. There are several methods that describe the global shape of the object using one or a few templates (prototypes) [7, 6, 17, 16, 21]. Typically, an explicit deformation model is used to account for intra-class variation [7, 6, 16]. Detection may then be performed by iteratively hypothesizing point-to-point correspondences and estimating deformation parameters [6] (in [6], a voting scheme using local shape features is used to initialize the shape matcher). Alternatively, detection can be based on finding (grouping) edge segments belonging to the object [16, 21, 7]. Leordeanu et. al. represent an object as a sparse set of directional features, connected by pairwise relationships in a graph structure [13]. Wu et. al. also present an



(a) The multi-local feature is a constellation of oriented edge elements (edgels).

(b) The feature is represented in Bookstein coordinates, using the first two edgels for alignment.

Figure 3. *Multi-local feature description.*

object model consisting of a sparse set of directional features (Gabor filters), but locations and orientations of the features are given relative to an object centered reference frame [20].

We have not seen any approach that specifically models the object as a *set* of geometrically consistent, discriminative, *global* shape features.

2. Multi-Local Features

In this section we describe the multi-local features. The multi-local features are specific spatial constellations of oriented edge elements (edgels). A typical example used for the detection of mugs is shown in figure 3. Two of the edgels are used for alignment. The locations and orientations of the remaining edgels are described relative to the feature coordinate system defined by letting the first edgel in the aligning pair be the origin and the second edgel be located at unit distance from the origin on the x-axis (Bookstein coordinates). We can also describe the bounding box of the target object in this coordinate system. The bounding box has five parameters; the centroid (x_c, y_c), the width w, the height h and the orientation α. A multi-local feature represents the sparse edgel constellation and the object bounding box in the feature coordinate system. A multi-local feature can thus be described by a parameter vector, $\Theta^{(k_e)}$, containing the orientations of the aligning edgel pair, the locations and orientations of the rest of the edgels and the parameters of the bounding box, as shown in equation 1 (where k_e is the number of edgels in the feature). Omitting the orientation of the edgels would make the feature significantly less discriminative.

$$\Theta^{(k_e)}=(\varphi_1,\varphi_2,x_3,y_3,\varphi_3,\ldots,x_{k_e},y_{k_e},\varphi_{k_e},x_c,y_c,w,h,\alpha) \quad (1)$$

Multi-local features will typically be applied to object detection. In this case, a multi-local feature should be shared across several exemplars of the target category while being discriminative against other visual inputs. Using too few edgels would yield a feature with no discriminative power. Increasing the number of edgels, k_e, makes the feature more discriminative and in the limit we get a shape template, representing a particular exemplar. For object detection, the most efficient representation of a given target category should be achieved using an ensemble of intermediate features that are shared by different sets of exemplars. The features can then be combined in different ways to generate a large number of exemplars. We shall now describe how to automatically learn an ensemble of multi-local features for object category detection.

(a) (b) (c) (d)

Figure 4. *a)-c) Edge maps of typical example training images. d) Learnt multi-local feature (aligning edgels marked in red).*

3. Learning

The extraction of multi-local feature ensembles should be adapted to the task at hand. For object detection, we want to learn a minimal ensemble that yield a sufficiently specific representation of all exemplars of the target category. We employ a simple boosting type algorithm and maintain a distribution over the examples reflecting the relative importance of the examples. We iteratively call a base learning algorithm to learn one multi-local feature using the current distribution. The distribution is then updated before the next call to the base learner. We first describe the base learning algorithm and then the boosting scheme to learn the ensemble.

3.1. Learning a single multi-local feature

The input to the learning algorithm is a set of example images, $\{I_i | i = 1, \ldots, n_t\}$, containing the target object category (no negative training examples are used) and a distribution, $\mathbf{d} = \{d_1, \ldots, d_{n_t}\}, \sum_{i=1}^{n_t} d_i = 1$, over these examples. Each training image is annotated with the bounding box(es) of the object(s) in the image.

In order to find a representative shape feature, $\Theta^{(k_e)}$, we want to maximize $P_{i\sim\mathbf{d}}\left(\Theta^{(k_e)} \in I_i\right)$ under the condition that $\Theta^{(k_e)}$ is sufficiently discriminative. In order to enforce discriminability, we require that $k_e \geq K$ (typically $K = 12$) and that the edgels of the feature are spread out, i.e. $\| (x_i - x_j, y_i - y_j) \| \geq \delta$ (typically $\delta = 0.1$). Figure 4 shows the edge maps of three example training images (4(a) - 4(c)) and a multi-local feature learnt by searching for the most common edgel constellation (4(d)).

We use a greedy algorithm to incrementally construct a multi-local feature. The first step is to find an edgel pair that occurs in as many training examples as possible, i.e. maximize $P_{i\sim\mathbf{d}}\left(\Theta^{(2)} \in I_i\right)$. This implies searching over $\Theta^{(2)} = (\varphi_1, \varphi_2, x_c, y_c, w, h, \alpha)$ for the feature that is found in the largest number of training examples (weighted by the distribution $\mathbf{d}$). In practice we sample $P_{i\sim\mathbf{d}}\left(\Theta^{(2)} \in I_i\right)$ on a regular lattice. For each sample point, we count the number of training examples that generate at least one sufficiently similar feature. We take the sample with the highest (weighted) count as the sought maximum. Recall that all parameters are expressed in the coordinate system defined by the first two edgels (the aligning pair).

After finding the most common pair, we incrementally add one edgel at a time to the feature until we have reached the required number of edgels, k_e. At each step in this process we select the edgel that maximizes the probability of finding the extended feature in an image drawn from the distribution $\mathbf{d}$, under the constraint that the new edgel is sufficiently far from all previously chosen edgels. Specifically, to find the kth edgel we want to maximize $P_{i\sim\mathbf{d}}\left(x_k, y_k, \varphi_k \in I_i | \Theta^{(k-1)} \in I_i\right)$. This requires searching over (x_k, y_k, φ_k), the coordinates of the kth edgel in the coordinate system defined by the aligning pair. In practice, we sample $P_{i\sim\mathbf{d}}\left(x_k, y_k, \varphi_k \in I_i | \Theta^{(k-1)} \in I_i\right)$ densely. Typically, about 2000 samples are used. If the base learner is called several times with similar distributions, $\mathbf{d}$, we would like the learner to return complementary features rather than several copies of the same feature. Therefore we add an element of chance by generating sample points randomly through uniform sampling in the aligned bounding box. For each sample point, we count the number of training examples containing $\Theta^{(k-1)}$ that also has an edgel sufficiently close to (x_k, y_k, φ_k). The count is weighted by the distribution $\mathbf{d}$. We then set x_k, y_k, φ_k equal to the sample point with the highest count, but constrain the selection so that $\| (x_k - x_i, y_k - y_i) \| \geq \delta, \forall i \leq k-1$. The selection process is terminated when the feature contains k_e edgels. See figure 6 for some sample features learnt this way.

This step of the learning process is somewhat similar in spirit to the method for learning active basis models in [20].

3.2. Learning an ensemble of multi-local features

Now that we have an algorithm for learning a single global shape feature, $\Theta^{(k_e)}$, given a set of examples and a distribution giving the relative importance of these examples, we can use a simple boosting algorithm to learn an ensemble of global shape features, $\left\{\Theta_1^{(k_e)}, \ldots, \Theta_{k_f}^{(k_e)}\right\}$. This algorithm should ensure that each training example is detected by a sufficient number of shape features. For this purpose we maintain weights representing the importance of each training example. The weights are initialized uniformly and are then updated after each call to the base learner described above. The weights are updated according to 2, where $\rho < 1$. The weights of examples that are detected by the learnt shape feature are thus decreased, forcing the base learner to focus on the other examples in future rounds. The process is repeated until k_f shape features are learnt (typically, $k_f = 40$).

$$d_i^{k+1} = \begin{cases} \rho \cdot d_i^k, & \text{if } \Theta_k^{(k_e)} \in I_i \\ d_i^k, & \text{if } \Theta_k^{(k_e)} \notin I_i \end{cases} \tag{2}$$

The output of the learning algorithm is thus a set of k_f learnt shape features described as in equation 1.

4. Detection

We now assume that we have learnt an ensemble of multi-local features such that each exemplar of the target category contains a number of features and that the different features are somewhat independent. We then want to aggregate information from different features. The detection process can be divided into two steps; (1) detecting individual multi-local features and (2) combining these detections into object detections (voting).

4.1. Detecting a single multi-local feature

See figure 5 for an illustration of how to detect individual multi-local features. We consider all (ordered) pairs of edgels in the image. For each such pair, we investigate if it might correspond to the pair of aligning edgels of a learnt multi-local feature. If there is any multi-local feature with sufficiently similar orientations of the aligning edgels, we hypothesize that the aligning edgels of that feature correspond to the pair of edgels in the image. The remaining edgels of that feature are then used to verify that hypothesis; the predicted image locations of these edgels are computed and we use the distance from these locations to the closest image edgel as given by the distance transform to determine if these verification edgels are present in the image. If the distance from each predicted location to the closest image edgel is sufficiently small, a feature was detected. Distances, coordinates and angles are always expressed relative to the coordinate system defined by the pair of aligning edgels, according to section 2.

4.2. Object detection using multi-local features

Different multi-local features are integrated by letting each detected feature vote for a specific bounding box in the image, described by parameters $\left(x_c^{(im)}, y_c^{(im)}, w^{(im)}, h^{(im)}, \alpha^{(im)}\right)$. The votes are combined in a Hough transform-like way. However, since we want to be able to extend this algorithm to high dimensional parameter spaces, we do not partition the parameter space

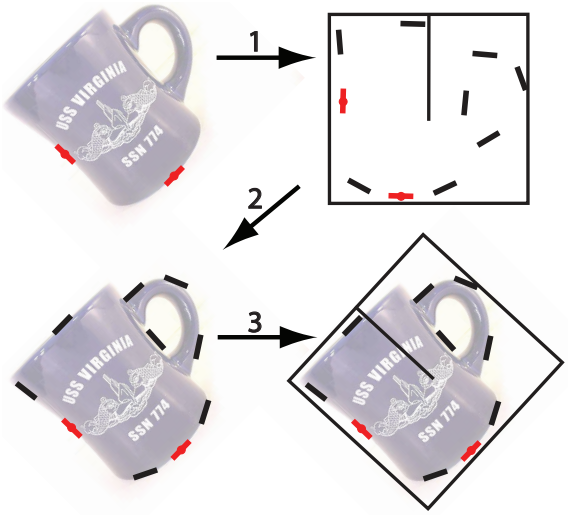


Figure 5. *Detection of individual shape features. 1) A pair of edgels in the image has orientations that match the aligning pair (red) of a stored multi-local feature. 2) The aligning pair of the stored multi-local feature is aligned with the pair of image edgels and we check if the rest of the edgels in the multi-local feature are present sufficiently close to their predicted locations in the image. 3) If all the remaining edgels are present, the detected feature casts a vote for the predicted bounding box of the object.*

into a regular lattice. Instead we employ a simple greedy clustering algorithm. We first compute the overlap between each pair of detected bounding boxes. The overlap is defined as the area of the intersection divided by the area of the union of the bounding boxes. If two bounding boxes have sufficient overlap and similar orientation (as given by the α parameter), they are said to vote for each other. For each feature detection, we can now count the number of other feature detections voting for it. We iteratively select the detection with most votes and add it to a list of candidate object detections. After each iteration, the selected detection and all other detections voting for it are removed from further voting. We finally threshold the number of votes of each candidate object detection to produce the output object detections.

5. Experiments and Dataset

We have claimed that an ensemble of multi-local features can be used to efficiently represent shape-based object categories and that such an ensemble can be learned automatically from training images. We now demonstrate this by evaluating the object detection performance when using multi-local features in a simple voting scheme on the ETHZ Shape Classes dataset [7]. This dataset is challenging due to large intra-class variation, clutter and varying scales. This dataset has been used by several other authors; in [16, 21, 7, 17] hand drawn models are evaluated and in [6, 9] models learnt from images are evaluated.

Experiments were performed on all classes in the dataset and results are produced as in [6], using 5-fold cross-validation. We build 5 different detectors for each class by randomly sampling 5 subsets of half the images of that class. All other images in the dataset are used for testing. The dataset contains a total of 255 images and the number of class images varies from 32 to 87. Thus, the number of training images will vary from 16 to 43 and the test set will consist of about 200 background images and 16 to 44 images containing occurrences of the target object.

Image Representation and Preprocessing All images (both for training and test) are preprocessed by extracting a set of edgels, $E = \{(x_i, y_i, \theta_i) \mid i = 1 \dots n\}$ (where n is the number of edgels), and by computing the distance transform [2] (which contains the distance from any pixel to the closest edgel). In fact, we divide the set of edgels into several subsets by quantizing the edgel orientation into several overlapping intervals and we compute one distance transform table for each subset. When doing a lookup in this table, we use the orientation, θ_q, of the query edgel, (x_q, y_q, θ_q), to determine which distance transform table to use. The returned value is thus the distance from the query edgel to the closest image edgel with similar orientation.

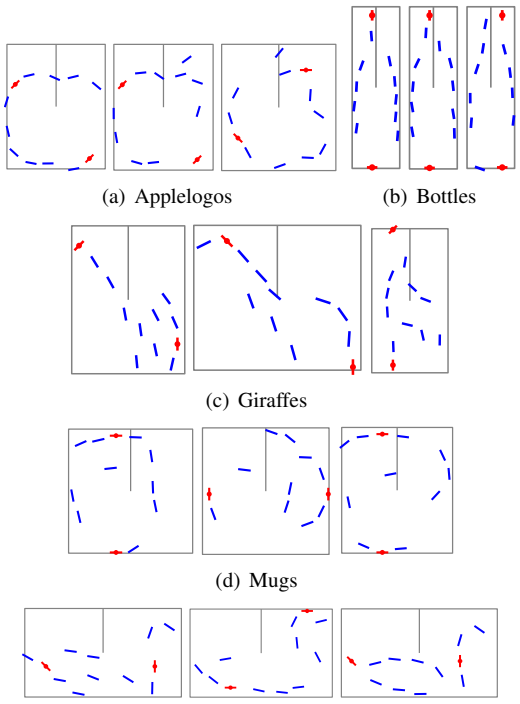
Edgels are extracted by sampling the image gradient at locations returned by an edge detector. In these experiments we used the edge maps provided with the ETHZ dataset (computed using the Berkley edge detector [14]).

6. Results

Figure 6 shows some sample shape features learnt by the training algorithm; the aligning edgels are marked in red. We notice that the aligning edgels tend to be far apart on the object; this makes the bounding box prediction robust against slight changes in the position of the aligning edgels. The remaining edgels are spread out to cover the whole object. Different shape features in general represent different subsets of the training set (however, each training example should be represented by several shape features).

Figure 7 shows some sample detections. The detected bounding boxes are displayed with a vertical line indicating the orientation of the detection. We also highlight all image edgels that were found to be in correspondence with the edgels of the shape features voting for a given bounding box. We can see that these edgels tend to be on the contour of the object and together they actually delineate the object boundary quite well. This could potentially be used to estimate the object boundary (even though only bounding box information was used during training).

In figure 8 we show some false positives. False positives typically occur in images with many edgels, where several



(a) Applelogos (b) Bottles

(c) Giraffes

(d) Mugs

(e) Swans

Figure 6. *Some sample multi-local features output by the learning algorithm (aligning edgels highlighted in red).*

shape features are accidentally detected at the same location. Since edge continuity is never enforced by our method, these features need not combine to form a connected structure. Appending a post-verification stage that checks for connectedness would undoubtedly reduce the number of false positives. In the future we aim to integrate some sort of verification into the method.

Quantitative results are plotted in figure 9 as the detection rate (the number of true positives divided by the number of occurrences of the target object in the test set) versus the number of false positives per image (FPPI). We prefer using FPPI, instead of precision, as a measure of error rate, since it is not biased by the number of positive and negative examples in the test set. Precision would for example be unsuitable for comparison to methods evaluated using a different cross-validation scheme, since this might affect the number of positive test examples.

A detection is counted as correct if the detected bounding box overlaps more than 50 % with the ground truth bounding box. Bounding box overlap is again defined as the area of intersection divided by the area of union of the bounding boxes. As a reference, [6] used 20 % and the PASCAL Challenge [3] uses 50 % overlap to define a correct detection. Our algorithm detects objects at any orientation, but to allow a fair comparison to previous methods that have only detected vertically oriented objects, we filter out all non-vertical detections.

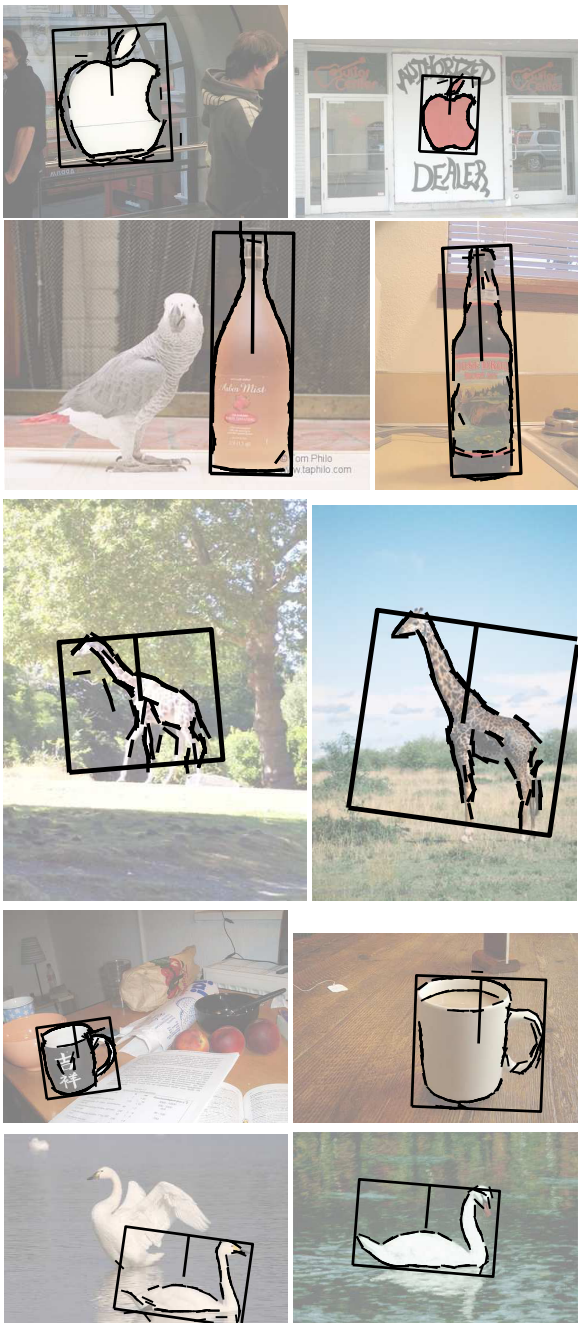


Figure 7. *Example detections. The image edgels that were active in producing a detection are highlighted in black.*

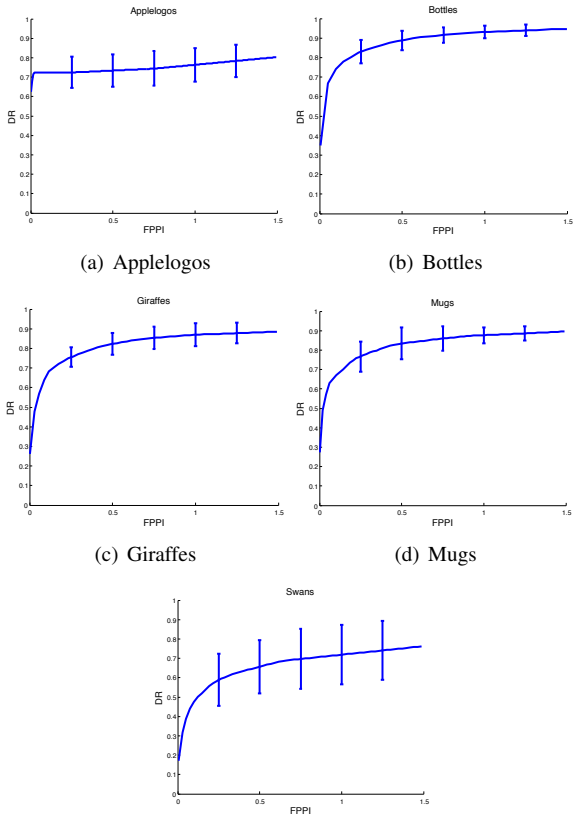
In table 1 we compare the detection performance of our system to the results presented in [6]. We choose to compare to [6] since that is the only model learnt from images that has been evaluated on this database (to the knowledge of the authors). We notice particularly that our performance on the Giraffe class is significantly better than [6]. One possible reason is that it is difficult to extract good edge segments from the giraffe images and since our method uses

Figure 8. *Example false positives of applelogo, bottle, giraffe, mug and swan ordered from top left to bottom right. The image edgels that were active in producing a detection are highlighted in black.*

edgels rather than edge segments, it has an advantage on this data. Our method performs worse on the applelogos and swans. In the case of applelogos, we believe these results to be partly due to the subsampling that we perform to produce the list of edgels, $E = \{(x_i, y_i, \theta_i) \mid i = 1 \dots n\}$, in the preprocessing stage. Some applelogos are quite small and are only represented by a small number (6-12) of edgels and are not detected at all. In the case of swans, individual shape features tended to generate bounding-box votes that were too different to end up in the same cluster in the voting/clustering step. Thus many swans were detected multiple times and each detection received only a few votes.

7. Discussion and Future Work

We have demonstrated multi-local features for shape-based object description and detection.



(a) Applelogos (b) Bottles

(c) Giraffes (d) Mugs

(e) Swans

Figure 9. *Detection rate (DR) plotted versus false positives per image (FPPI).*

Table 1. Comparison of detection performance. We state the average detection rate at 0.4 FPPI and its standard deviation in parentheses. We compare to the full system of [6], using models learnt from images.

	A. logos	Bott.	Gir.	Mugs	Swans
[6]:	83.2 (1.7)	83.2 (7.5)	58.6 (14.6)	83.6 (8.6)	75.4 (13.4)
our syst:	73.0 (8.2)	86.9 (5.2)	80.3 (6.2)	81.6 (7.5)	63.5 (12.4)

By learning an ensemble of multi-local features we not only get an efficient representation of the intra-class variation but also redundancy, yielding a robustness against occlusion. Furthermore, it is not always the case that parts of the object are occluded and the rest is visible; it may also happen that the whole object is partially occluded; consider for example an animal seen through foliage. Ideally, multi-local features should be less sensitive to this type of situation, since it does not rely on extracting complete parts or edge segments.

By selecting edgel constellations that are common to

several training examples the learning algorithm is able to reject clutter in the training data. During detection, the algorithm only requires presence of edgels and not absence, so the addition of clutter does not cause false negatives. However, it can cause false positives.

There are a number possibilities for future research. Firstly, in order to build a complete object detection system, it will be necessary to add a verification step, for example append a shape matching stage as in [6]. As mentioned, this would reduce the number of false positives. Possible extensions include: 1) using multi-local features to describe parts of articulated objects (e.g. animals), 2) using multi-local features for 3D object detection, 3) combining multi-local features with discriminative local features and 4) building hierarchies of multi-local features.

8. Conclusion

We have motivated using multi-local features to efficiently model and detect object categories defined by their global shape properties. Each feature is represented and detected independently of other features. We thus leverage many of the advantages of part-based models for object categories with distinctive local features. Multi-local features may also be a good complement to local features for objects that have discriminative local features. We have shown that multi-local features, combined in a manner similar to the voting schemes used with traditional constellation models, yield a detection performance comparable to state of the art methods. We view this as a proof of concept and we suggest several possibilities for future work in hope to inspire attention to an area that we think has not been sufficiently explored.

Acknowledgement

This work was supported by the Swedish Foundation for Strategic Research (SSF) project VINST.

References

[1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time euclidean distance transform algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(5):529–533, 1995.

[3] M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2008 (voc2008) results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, pages 55–79, 2005.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2003.

[6] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2007.

[7] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. *Proc. of the European Conference on Computer Vision*, 2006.

[8] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):67–92, 1973.

[9] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2008.

[10] D. M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8), 2007.

[11] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2004.

[12] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. *Proc. of the British Machine Vision Conference*, 2003.

[13] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2007.

[14] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[15] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment model for object detection. *Proc. of the European Conference of Computer Vision*, 2006.

[16] S. Ravishankar, A. Jain, and A. Mittal. Multi-stage contour based detection of deformable objects. *Proc. of the European Conference on Computer Vision*, 2008.

[17] K. Schindler and D. Suter. Object detection by global contour shape. *Pattern Recognition*, 41(12):3736–3748, 2008.

[18] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. *Proc. of the International Conference of Computer Vision*, 2005.

[19] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for visual object class recognition. *Proc. of the European Conference on Computer Vision*, 2000.

[20] Y. N. Wu, Z. Si, C. Fleming, and S. C. Zhu. Deformable template as active basis. *Proc. of the International Conference on Computer Vision*, 2007.

[21] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. *Proc. of the European Conference on Computer Vision*, 2008.