Int J Comput Vis
DOI 10.1007/s11263-009-0228-y

Shape-Based Object Localization for Descriptive Classification

Geremy Heitz · Gal Elidan · Benjamin Packer · Daphne Koller

Received: 12 October 2007 / Accepted: 2 March 2009

Abstract Discriminative tasks, including object categorization and detection, are central components of high-level computer vision. However, sometimes we are interested in a finer-grained characterization of the object's properties, such as its pose or articulation. In this paper we develop a probabilistic method (LOOPS) that can learn a shape and appearance model for a particular object class, and be used to consistently localize constituent elements (landmarks) of the object's outline in test images. This localization effectively projects the test image into an alternative representational space that makes it particularly easy to perform various descriptive tasks. We apply our method to a range of object classes in cluttered images and demonstrate its effectiveness in localizing objects and performing descriptive classification, descriptive ranking, and descriptive clustering.

Keywords Probabilistic graphical models · Deformable shape models · Object recognition · Markov random fields

Authors G. H., G. E. and B. P. contributed equally to this manuscript.

G. Heitz (✉) · B. Packer · D. Koller
Stanford University, Stanford, CA 94305, USA
e-mail: gaheitz@cs.stanford.edu

B. Packer
e-mail: bpacker@cs.stanford.edu

D. Koller
e-mail: koller@cs.stanford.edu

G. Elidan
Department of Statistics, Hebrew University of Jerusalem, Jerusalem, 91905, Israel
e-mail: galel@huji.ac.il

1 Introduction

Discriminative questions such as "What is it?" (categorization) and "Where is it?" (detection) are central to machine vision and have received much attention in recent years. In many cases, however, we are also interested in more refined descriptive questions. We define a descriptive query to be one that considers characteristics of an object that vary between instances within the object category. Examples include questions such as "Is the giraffe standing upright or bending over to drink?", "Is the cheetah running?", or "Find me all lamps that have a beige, rectangular lampshade". These questions relate to the object's pose, articulation, or localized appearance.

In principle, it is possible to convert such descriptive queries into discriminative classification tasks given an appropriately labeled training set. But to take such an approach, we would need to construct a specialized training set (often a large one) for each descriptive question we want to answer, at significant human effort. Intuitively, it seems that we should be able to avoid this requirement. After all, to a person, we can convey the difference between a standing and bending giraffe using one or two training instances of each. The key, of course, is that the person has a good representational model of what giraffes look like, one in which the salient differences between the different descriptive labels are easily captured and learned.

In this paper, we focus on the aspect of object shape, an important characteristic of many objects that can be used as the basis for many descriptive distinctions (Felzenszwalb and Huttenlocher 2000), including the examples described above. We introduce a method that automatically learns a probabilistic characterization of an object class shape and appearance, allowing instances in that class to be encoded in this new representational space. We provide an algorithm

Published online: 24 March 2009

for converting new test images containing the object into this representation, and show that this ability allows us to answer difficult descriptive questions, such as the ones above, using a very small number of training instances. Importantly, the specific descriptive questions are not known at the time the probabilistic model is learned, and the same model can be used for answering multiple questions.

Concretely, we propose an approach called LOOPS: Localizing Object Outlines using Probabilistic Shape. Our LOOPS model combines two components: a model of the object's deformable shape, encoded as a joint probability distribution over the positions of a set of automatically selected *landmark* points on the outline of the object; and a set of appearance-based boosted detectors that are aimed at individually localizing each of these landmarks. Both components are automatically learned from a set of labeled training contours, of the type obtained from the LabelMe dataset (http://labelme.csail.mit.edu). The location of these landmarks in an image, plus the landmark-localized appearance, provide an alternative representation from which many descriptive tasks can be easily answered. A key technical challenge, which we address in this paper, is to correspond this model to a novel image that often contains a large degree of clutter and object deformation or articulation.

Contour-based methods such as active shape/appearance models (AAMs) (Cootes et al. 1998; Sethian 1998) were also developed with the goal of localizing an object shape model to an image. However, these methods typically require good initial guesses and are applied to images with significantly less clutter than real-life photographs. As a result, AAMs have not been successfully used for class-level object recognition/analysis. Some works use some form of geometry as a means toward an end goal of object classification or detection (Fergus et al. 2003; Hillel et al. 2005; Opelt et al. 2006b; Shotton et al. 2005). Since, for example, a misplaced leg has a negligible effect on classification, these works neither attempt to optimize localization nor evaluate their ability to do so. Other works do attempt to accurately localize objects in cluttered photographs but only allow for relatively rigid configurations (e.g., Berg et al. 2005; Ferrari et al. 2008), and cannot capture large deformations such as the articulation of the giraffe's neck (see Fig. 1).

To allow robust localization of landmarks in cluttered images with significant deformation, we propose a hybrid method, which combines both discrete global and continuous local search steps. In the discrete phase, we use a discrete space defined by a limited set of candidate assignments for each landmark; we use a Markov random field (MRF) to define an energy function over this set of assignments, and effectively search this multi-modal, combinatorial space by applying state-of-the-art probabilistic inference methods over this MRF. This global search step allows us to find a rough but close-to-optimal solution, despite the many local optima resulting from the clutter and the large deformations allowed by our model. This localization can then be refined using a continuous hill-climbing approach, allowing a very good solution to be found without requiring a good initialization or multiple random restarts. Preliminary investiga-

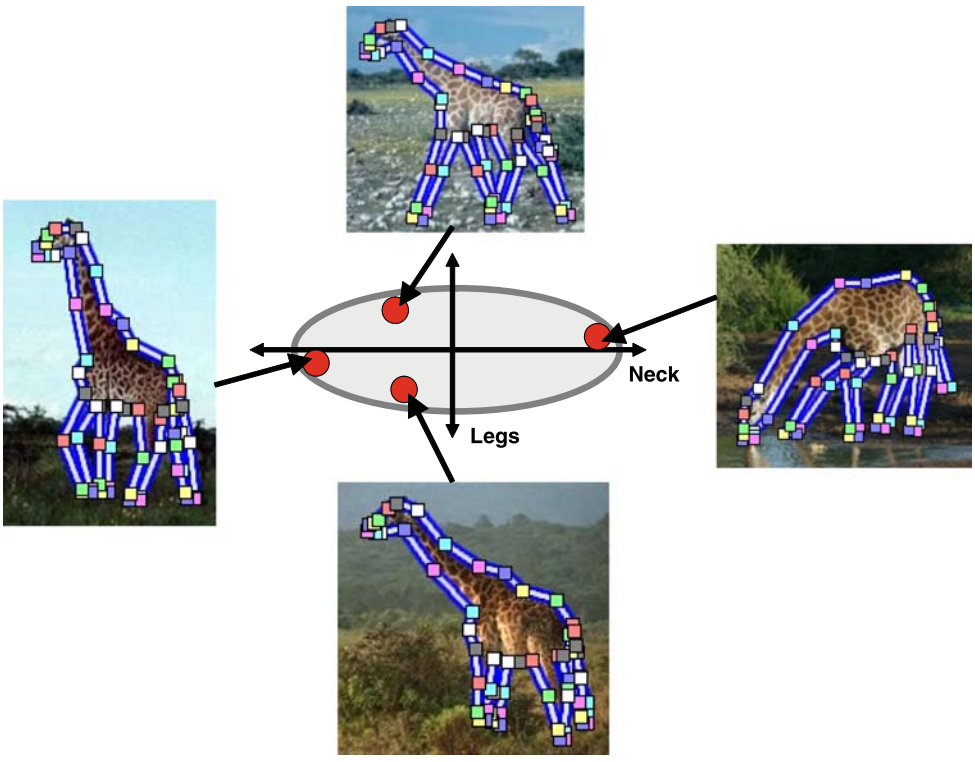


Fig. 1 Example deformations of the giraffe object class. The *axis* shows the location of instances along the two principal components in our dataset. The *horizontal axis* corresponds to articulation of the neck, while the *vertical axis* corresponds to articulations of the legs. The *ellipse* indicates the set of instances within one standard deviation of the mean. We show four typical examples of giraffes that illustrate the outer extremes along the first two modes of variation

tions showed that a simpler approach that does a purely local search, similar to the active appearance models of Cootes et al. (1998), was unable to deal with the challenges of our data.

To evaluate the performance of our method, we consider a variety of object classes in cluttered images and explore the space of applications facilitated by the LOOPS model in two orthogonal directions. The first concerns the machine learning task: we present results for classification, search (ranking), and clustering. The second direction varies the components that are extracted from the LOOPS outlines for these tasks: we show examples that use the entire object shape, a subcomponent of the object shape, and the appearance of a specific part of the object. Given enough task-specific data, each of these applications might be accomplished by other methods. However, we show that the LOOPS framework allows us to approach these tasks with a single class-based model, without the need for task-specific training.

2 Overview of the LOOPS Method

As discussed, having a representation of the constituent elements of an object should aid in answering descriptive questions. For example, to decide whether a giraffe is standing upright or bending down to drink, we might look at the characteristics of the model elements that correspond to the head, neck, or legs. In this section we briefly describe our representation over these automatically learned elements (landmarks). We also provide an overview of the LOOPS method for learning an object class model over these landmarks and using it to localize corresponded outlines in test images. A flowchart of our method is depicted in Fig. 2.

We start by representing the shape of an object class via an ordered set of N landmark points that together constitute a piecewise linear contour (see Fig. 1). It is important that these landmarks be *corresponded* across instances—landmark '17' in one instance should correspond to the same meaningful point (for example, the nose) as landmark '17' in another instance. Obtaining training instances labeled with corresponded landmarks, however, requires painstaking supervision. Instead, we would like to use simple outlines such as those in the LabelMe dataset (http://labelme.csail.mit.edu), which requires much less supervision and for which many examples are available from a variety of classes. We must therefore *automatically* augment the simple training outlines with a corresponded labeling. That is, we want to specify a relatively small number of landmarks that still represent the shape faithfully, and consistently position them on all training outlines. This stage transforms arbitrary outlines into useful training data as depicted in Fig. 2(a). We describe our method for performing this automatic correspondence in Sect. 4. Once we have our set of training outlines, each with N corresponded landmarks, we can construct a distribution of the geometry of the objects' outline as depicted in Fig. 2 (middle) and augment this with appearance based features to form a LOOPS model, as described in Sect. 5. With a model for an object class in hand, we now face the real computational challenge that our goal poses: how to localize the landmarks of the model in test images in the face of clutter and large deformations and articulations (fourth box in Fig. 2). Our solution involves a two-stage scheme: we first use state-of-the-art inference from the

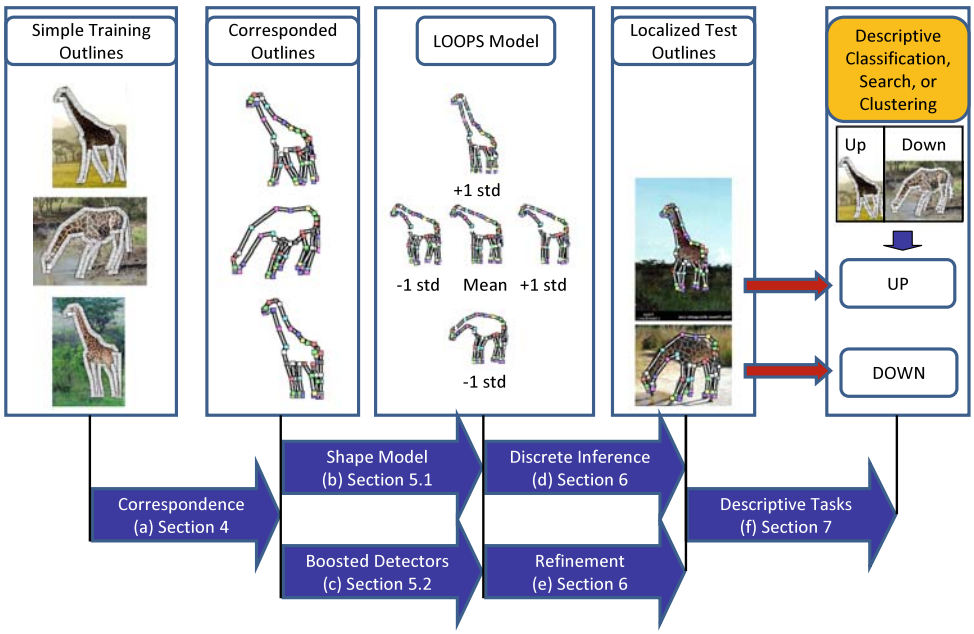


Fig. 2 A flowchart depicting the stages of our LOOPS method

field of graphical models to achieve a coarse solution in a discretized search space, and then refine this solution using greedy optimization. The localization of outlines in test images is described in detail in Sect. 6. Finally, once the localization is complete, we can readily perform a range of descriptive tasks (classification, ranking, clustering), based on the predicted location of landmarks in test images as well as appearance characteristics in the vicinity of those landmarks. We demonstrate how this is carried out for several descriptive tasks in Sect. 8.

3 Related Work

Our method relies on a joint probabilistic model over landmark locations that unifies boosted detectors and global shape toward corresponded object localization. Our boosted detectors are constructed, for the most part, based on the state-of-the-art object recognition methods of Opelt et al. (2006b) and Torralba et al. (2005) (see Sect. 5.2 for details). In this section, we concentrate on the most relevant works that employ some notion of geometry or "shape" for various image classification tasks.

Shape Based Classification Several recent works study the ability to classify the shape of an outline directly (Basri et al. 1998; Grauman and Darrell 2005; Belongie et al. 2000; Sebastian et al. 2004; Felzenszwalb and Schwartz 2007). Such techniques often take on the form of a matching problem, in which the cost between two instances is the cost of "deforming" one shape to match the other. Most of these methods assume that the object of interest is either represented as a contour or has already been segmented out from the image. While many of these approaches could aid classification of objects in cluttered scenes, few of them address the more serious challenge of extracting the object's outline from such an image. One exception is the work of Thayananthan et al. (2003), which uses shape-contexts for object localization in cluttered scenes. Their method demonstrates matching of rigid templates and a manually-constructed model of the hand, but does not attempt to learn models from images.

Contour Propagation Methods Some of the best-studied methods in machine vision for object localization and boundary detection rely on shape to guide the search for a known object. In medical imaging, active contour techniques such as snakes (Cremers et al. 2002; Caselles et al. 1995), active shape models (Cootes et al. 1995), or more recently level set and fast marching methods (Sethian 1998) combine simple image features with relatively sophisticated shape models. In medical imaging applications, where the shape variation is relatively low, and a good starting point can easily be found, these methods have achieved impressive results.

Attempting to apply these methods to recognizing object classes in cluttered scenes, however, faces two problems. The first is that we generally cannot find a good starting point, and even if we do, localization is complicated by the many local maxima present in real images. Secondly, edge cues alone are typically insufficient to identify the object in question. In an attempt to overcome this problem, active appearance models have been introduced by Cootes et al. (1998). While this method achieved success in the context of medical imaging or face outlining in images with little additional clutter, they have not (to the best of our knowledge) been successfully applied to the kind of cluttered images that we consider in our experimental evaluation. Indeed, our own early experiments confirmed that a contour front propagation approach with several starting points is generally unable to accurately locate outlines in complex images, and simple methods for finding a good starting point—such as first detecting a bounding box—are insufficient.

Contour Outlining in Real Images Ferrari et al. match test images to a contour-based shape model constructed from a single hand-drawn outline (Ferrari et al. 2006) and learn a similar model automatically from a set of training images with supervised bounding boxes (Ferrari et al. 2007, 2008). Their algorithm achieves impressive results at multiple scales and in a fair amount of clutter using only detected edges in the image. However, they do not, in these papers, attempt to use the discovered shapes for any descriptive classification task. In experiments below, we compare to this method, and show that our method produces more useful outlines for the descriptive classification tasks that we target. Furthermore, our method is more appropriate for classes in which shape is more deformable or cluttered images where features other than edge fragments might be useful.

"Parts"-Based Methods Following the generative constellation model of Fergus et al. (2003), several works make use of explicit shape by relying on a model of parts (patches) and their geometrical arrangement (e.g., Hillel et al. 2005; Fergus et al. 2005). Most of these models are trained for detection or classification accuracy, rather than accurate localization of the constituent "parts", and do not provide evidence that the discovered parts have any consistent meaning or localization. In fact, even with a "correct" set of parts, localization need not be accurate to aid detection/classification. To see this, consider an example where the giraffe abdomen and legs are localized correctly but the head and neck are incorrectly pointing upwards. In this case, classification and bounding box detection are still probably correct but the localization cannot be used to determine the pose of the giraffe. Indeed, instead of trying to localize

parts, the constellation approach *averages* over localization. While this is the correct strategy when the goal is detection/classification, it fails to provide a single most-likely assignment of parts that could be used for the further processing that we consider.

The work of Crandall et al. (2005) uses a parts-based model with manually annotated parts. They evaluate the accuracy of localizing these parts in test images. Their follow up work (Crandall and Huttenlocher 2006) automatically learns the parts, but similar to the constellation method, uses these only as cues for object detection. Our work learns object landmarks automatically and explicitly uses them not only for detection, but for further descriptive querying.

Additional Use of Shape for Classification, Detection and Segmentation Foreground/background segmentation is another task that is closely related to object outlining. The class-based segmentation works of Kumar et al. (2005) and Borenstein et al. (2004) rely on class-specific image features to segment objects of the relevant class. These approaches, however, aim to maximize pixel (or region) label prediction and do not provide a corresponded outline. Winn and Shotton (2006) attempt simultaneous detection and segmentation using a CRF over object parts. They evaluate their segmentation quality, but do not consider the accuracy of their part localization. In experiments below, we use existing code from Prasad and Fitzgibbon (2006), which is based on the work of Kumar et al. (2005), to produce uncorresponded segmentation outlines that are compared to de-corresponded outlines produced by our method.

Many recent state-of-the-art methods rely on implicit shape to aid in object recognition. For instance, Torralba et al. (2005) use image patches, and Opelt et al. (2006b) and Shotton et al. (2005) use edge fragments as weak detectors with offsets from the object centroid that are boosted to produce a strong detector. These works combine features in order to optimize the aggregate decision, and typically do not aim to segment/outline objects. Leordeanu et al. (2007) and Berg et al. (2005) similarly combine local image features with pairwise relationship features to solve model matching as a quadratic assignment problem. While such models give a correspondence between model components and the image, the training regime optimizes recognition rather than the accuracy of these correspondences. Despite the training regimes of the above models, the implicit shape used by these models can be used to produce both a detection and a localization through intelligent use of the "weak" detectors and their offsets. Indeed, both Leibe et al. (2004) and Opelt et al. (2006a) produce both a detection and soft segmentation. While these methods tend to show appealing anecdotal results, the merit of their localizations (both corresponded and uncorresponded) is hard to measure, as no quantitative evaluation of segmentation quality or "part" localization is provided.

4 Automatic Landmark Selection

In this section we describe our method for transforming simple outlines into corresponded outlines over a relatively small and consistent set of landmarks, as in Fig. 3. Many works have studied the problem of automatic landmark selection for point distribution models (PDMs) for use in ASMs or AAMs. In particular, Hill and Taylor (1996) developed a robust method based on optimization of a cost function over a sparse polygonal representation of the outlines. Any robust method for correspondence between training outlines and salient landmark selection should work for our purposes, and for completeness we present our method, which is simple and was robust and accurate for all classes that we considered. Our method builds on an intuition similar to that of Hill and Taylor (1996).

Intuitively, a good correspondence trades off between two objectives. The first is that corresponding points should

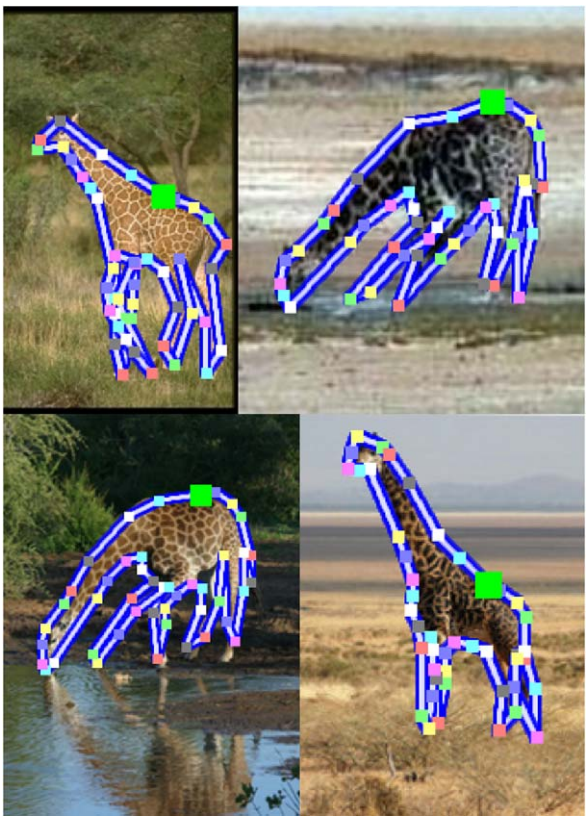


Fig. 3 Example training instances with an outline defined by a piecewise linear contour connecting 42 landmarks. Note that the landmarks are not labeled in the training data but automatically learned using the method described in Sect. 4. These instances demonstrate the effectiveness of the correspondence method (e.g., the larger green landmark in the middle of the back is landmark #7 in all instances) as well as imperfections (e.g., misalignments near the feet). The most systematic problems occur in articulated instances, such as these. While the three landmarks at the lower back of the giraffe are consistent across these instances, some of the landmarks along the head must "slide" in order to keep the overall alignment correct

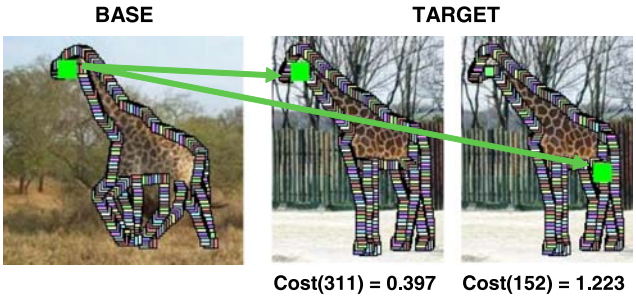


Fig. 4 Our arc-length correspondence procedure. (**a**) The first outline, $\mathbf{O}^{(1)}$, with landmark 0 marked. (**b**), (**c**) The second outline, $\mathbf{O}^{(2)}$, with two different choices for landmark 0, along with the correspondence cost for that choice

be "equally spaced" around the boundary of the object, and the second is that the nonrigid transformation that aligns the landmark points should be small, in some meaningful sense. In our case, we achieve the correspondence using a simple two-step process: find a correspondence between high-resolution outlines then prune down to a small number of "salient" landmarks.

Arc-Length Correspondence Our correspondence algorithm searches for a correspondence between pairs of instances that achieves the lowest cost. Suppose we have a candidate correspondence between two outlines, represented by the vectors of points $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$. We first analytically determine the transformation of $\mathbf{O}^{(2)}$ that minimizes its least-mean-square distance to $\mathbf{O}^{(1)}$. Applying this transformation to each landmark produces $\tilde{\mathbf{O}}^{(2)}$. Our score penalizes changes in the offset vectors between distant landmarks on our two outlines. In particular, we compute our cost as:

$$Cost(\mathbf{O}^{(1)}, \tilde{\mathbf{O}}^{(2)}) = \sum_{i,j} \|\delta_{ij}^{(1)} - \tilde{\delta}_{ij}^{(2)}\|^2,$$

where $\delta_{ij}^{(m)}$ is the vector offset between landmark i and landmark j on contour m. The sum over i, j includes all landmarks i together with the landmark j that has largest geodesic distance along the contour from landmark i. We note that our correspondence is scale-invariant since the target contour is affine transformed before the cost is computed.

In our search for the best correspondence according to this score, we leverage the fact that our outlines are ordered one-dimensionally. We begin by sampling a large number (500) of equally spaced landmarks along each of the training outlines. We fix a "base" contour (randomly selected), and a landmark numbering scheme for that base. We can now correspond the entire dataset by corresponding each outline to our base. Figure 4 shows the cost assigned by our algorithm for two sample correspondences from a target outline to the base outline. Furthermore, because of our ordering assumption and our fixed choice of landmarks (the 500 equally spaced points), there are only 500 possible correspondences. This can be seen because landmark 0 in the base outline only has 500 choices for a corresponding landmark in each target outline. We can thus simply compute the cost for all 500 choices, and select the best one. Once we have selected the correspondence for each target contour, we have a fully corresponded dataset.

The entire process is simple and takes only a few minutes to choose the best amongst 10 random base instances and correspond each to the 20 training outlines. While a more sophisticated approach might lead to strictly more accurate correspondences, this approach was more than sufficient for our purposes.

Landmark Pruning The next step is to reduce the number of landmarks used to represent each outline in a way that takes into account *all* training contours. Our goal is to remove points that contribute little to the outline of any instance. Toward this end, we greedily prune landmarks one at a time. At any given time in this process, each landmark can be scored by the error its removal would introduce into the dataset. Suppose that our candidate for removal is landmark i at location l_i. If i is removed from instance m, the outline in the vicinity of landmark i will be approximated by the line segment between landmarks $i-1$ and $i+1$ at locations $l_{i-1}^{(m)}$ and $l_{i+1}^{(m)}$. We define $dist_{\mathbf{O}^{(m)}}(l_{i-1}^{(m)}, l_{i+1}^{(m)})$ to be the mean segment-to-outline squared distance, and let the cost of removing landmark i be the average of these distances across all M instances in the entire dataset:

$$C_i = \frac{1}{M} \sum_m dist_{\mathbf{O}^{(m)}}(l_{i-1}^{(m)}, l_{i+1}^{(m)}).$$

At each step, we remove the landmark whose cost is lowest, and terminate the process when the cost of the next removal is above a fixed threshold (2 pixels^2).

Figure 3 shows an example of the correspondence of giraffe outlines. Note that the fully automatic choice of landmarks is both sparse and similar to what a human labeler might choose. In the next section we show how to use corresponded outlines to learn a shape and appearance model that will later be used (see Sect. 6) to detect objects and precisely localize these landmarks in cluttered images.

5 The LOOPS Model

Once we have corresponded the training outlines so that each is specified by N (corresponded) landmarks, we are ready to construct a distribution over possible assignments of these landmarks to pixels in an image. Toward this end, the LOOPS model combines two components: an explicit representation of the object's shape (2D silhouette), and a set of image-based features. We define the shape of a class of

objects via the locations of the N object landmarks, each of which is assigned to one of the pixels in the image. We represent such an assignment as a $2N$ vector of image coordinates which we denote by $\mathbf{L}$. Using the language of Markov random fields (Pearl 1988), the LOOPS model defines a conditional probability distribution over $\mathbf{L}$:

$$\begin{aligned} &P(\mathbf{L} \mid \mathcal{I}, \mathbf{w}, \mu, \mathbf{\Sigma}) \\ &\quad = \frac{1}{Z(\mathcal{I})} P_{\text{Shape}}(\mathbf{L}; \mu, \mathbf{\Sigma}) \prod_i \exp\left(w_i F_i^{\text{det}}(l_i; \mathcal{I})\right) \\ &\qquad \times \prod_{i,j} \exp\left(w_{ij} F_{ij}^{\text{grad}}(l_i, l_j; \mathcal{I})\right), \end{aligned} \tag{1}$$

where $\mu, \mathbf{\Sigma}, \mathbf{w}$ are model parameters, and i and j index landmarks of the model. P_{Shape} encodes the (unnormalized) distribution over the object shape (outline), $F^{\text{det}}(l_i)$ is a landmark specific detector, and $F_{ij}^{\text{grad}}(l_i, l_j; \mathcal{I})$ encodes a preference for aligning outline segments along image edges.

The shape model and the detector features are learned in parallel, as shown in Fig. 2(b). Below, we describe these features and how they are learned. The weights $\mathbf{w}$ are set as follows: $w_i = 5$, $w_{ij} = 1$, for all i, j. In principle, we could learn these weights from data, for example using a standard conjugate gradient approach. This process, however, requires an expensive inference step at each iteration. Our preliminary experiments indicated that our outlining results are relatively robust to a range of these weights and that learning the weights provides no clear benefit. We therefore avoid this computational burden and simply use a fixed setting of the weights.

We note that our MRF formulation is quite general, and allows for both flexible weighting of the features, and for the incorporation of additional features. For instance, we might want to capture the notion that internal line segments (lines entirely contained within the object) should have low color variability. This can naturally be posed as a pairwise feature over landmarks on opposite sides of the object.

5.1 Object Shape Model

We model the shape component of (1) as a multivariate Gaussian distribution over landmark locations with mean μ and covariance $\mathbf{\Sigma}$. The Gaussian parametric form has many attractive properties, and has been used successfully to model shape distributions in a variety of applications (e.g., Cootes et al. 1995; Anguelov et al. 2005). In our context, one particularly useful property is that the Gaussian distribution decomposes into a product of quadratic terms over pairs of variables

$$\begin{aligned} &P_{\text{Shape}}(\mathbf{L} \mid \mu, \mathbf{\Sigma}) \\ &\quad = \frac{1}{Z} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu)\right) \\ &\quad = \frac{1}{Z} \prod_{i,j} \exp\left(-\frac{1}{2}(x_i - \mu_i)\mathbf{\Sigma}_{ij}^{-1}(x_j - \mu_j)\right) \\ &\quad = \frac{1}{Z} \prod_{i,j} \phi_{i,j}(x_i, x_j; \mu, \mathbf{\Sigma}), \end{aligned} \tag{2}$$

where Z is the normalization factor for the Gaussian density. We can see from (2) that we can specify potentials $\phi_{i,j}$ over only singletons and pairs of variables and still manage to reconstruct the full likelihood of the shape. This allows (1) to take an appealing form in which all terms are defined over at most a pair of variables.

As we discuss below in Sect. 6, the procedure to locate the model landmarks in an image first involves discrete global inference using the LOOPS model, followed by a refinement stage that takes local steps. Even if we limit ourselves to pairwise terms, performing discrete inference in a densely connected MRF may be computationally impractical. Unfortunately, a general multivariate Gaussian includes pairwise terms between all landmarks. Thus, during the discrete inference stage, we limit the number of pairwise elements by approximating our shape distribution with a sparse multivariate Gaussian. During the final refinement stage, we use the full multivariate Gaussian distribution.

The sparse shape approximation can be selected in various ways, trading off accuracy of the distribution with computational resources. In our implementation, we include only two types of terms: terms correlating neighboring landmarks along the outline, and a linear number (one) of terms encoding long-range dependencies that promote stability of the shape. We greedily select the latter terms over pairs of landmarks for which the relative location has the least variance (averaged across the x and y coordinates).

Estimation of the maximum likelihood parameters of the full Gaussian distribution may be solved analytically using the corresponded training instances. If we desire a sparse Gaussian distribution, however, there are multiple options for which approximation to use. We use a standard iterative projected gradient approach (Boyd and Vandenberghe 2004) to minimize the Kullback-Leibler divergence (Cover and Thomas 1991) between the sparse distribution and the full maximum likelihood distribution:

$$\begin{aligned} &(\mu_S, \mathbf{\Sigma}_S^{-1}) \\ &\quad = \underset{\mu, \mathbf{\Sigma}^{-1} \in \Delta}{\arg\min}\ KL\left(\mathcal{N}(\mu, \mathbf{\Sigma}^{-1}) \| \mathcal{N}_D(\mu_D, \mathbf{\Sigma}_D^{-1})\right) \end{aligned}$$

where μ_D and $\mathbf{\Sigma}_D$ are the (dense) ML parameters, and Δ is the set of all inverse covariances that meet the constraints defined by our choice of included edges. Note that a removal of a potential (edge) between a pair of landmarks is equivalent to constraining the corresponding entry in $\mathbf{\Sigma}^{-1}$ to be zero.

5.2 Landmark Detector Features

To construct our detector features F^{det}, we build on the demonstrated efficacy of discriminative methods in identifying salient regions (parts) of objects (e.g., Fergus et al. 2003; Hillel et al. 2005; Torralba et al. 2005; Opelt et al. 2006b). The extensive work in this area suggests that the best results are obtained by combining a large number of features. However, incorporating all of these features directly into (1) is problematic because setting such a large number of weights corresponding to these features requires a significant amount of tuning. Even if we chose to learn the weights, this problem would not be alleviated: training a conditional MRF requires that we run inference multiple times as part of the gradient descent process; models with more parameters require many more iterations of gradient descent to converge, making the learning cost prohibitive.

Our strategy builds on the success of boosting in state-of-the-art object detection methods (Opelt et al. 2006b; Torralba et al. 2005). Specifically, we use boosting to learn a strong detector (classifier), H_i for each landmark i. We then define the feature value in the conditional MRF for the assignment of landmark i to pixel l_i to be:

$$F_i^{\text{det}}(l_i; \mathcal{I}) = H_i(l_i).$$

Any set of image features can be used in this approach; we use features that are based on our shape model as well as other features that have proven useful for the task of object detection (see Fig. 5 for examples of each):

- *Shape Templates*. This contour-based feature is a contiguous segment of the mean shape emanating from a particular landmark for a given pixel radius in both directions. This feature is entirely shape dependent and can only be used in the case where training outlines are provided. See Elidan et al. (2006a) for further details on this feature. Because we expect the edges in the image to roughly correspond to the object outline, we match a shape template to actual edges in the image using chamfer distance matching (Borgefors 1988).
- *Boundary Fragments*. This feature is made of randomly selected edge chains (contiguous groups of edge pixels) within the bounding box of the training objects, and are matched using chamfer distance. We construct these using a protocol similar to Opelt et al. (2006b), except that we neither cluster fragments nor combine pairs of fragments into a single feature.
- *Filter Response Patches*. This feature is represented by a patch from a filtered version of the image (filters include bars and oriented Gaussians). Patches are randomly extracted from within the object bounding boxes in the training images. The patch is matched to a new image using normalized cross-correlation. We construct the patches similarly to Torralba et al. (2005).
- *SIFT Descriptors*. Interesting keypoints are extracted from each image, and a SIFT descriptor (Lowe 2003) is computed for each keypoint. In order to limit the number of SIFT features, each descriptor is scored based on how common it is in the training set, and the best scoring descriptors are kept.

Each landmark detector is trained by constructing a strong boosted classifier from a set of the weak single feature detectors described above. To build such detectors, we rely on boosting and adapt the protocol of Torralba et al. (2005). We now provide a brief description of the essentials of the process.

Boosting provides a straightforward way to perform feature selection and learn additive classifiers of the form

$$H_i(p) = \sum_{t=1}^{T} \alpha_t h_i^t(p),$$

where each $h_i^t(p)$ is a weak feature detector for landmark i (described below), T is the number of boosting rounds, and $H_i(p)$ is a strong classifier whose output is proportional to the log-odds of landmark i being at pixel p.

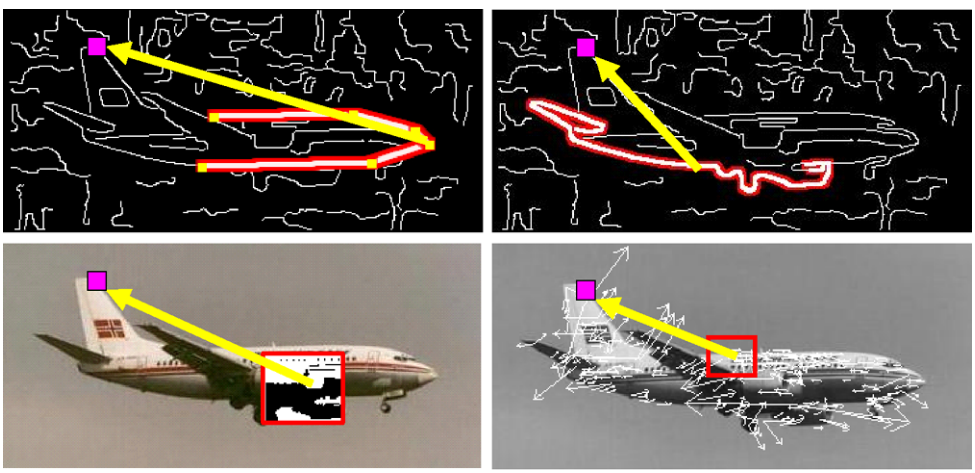


Fig. 5 (Color online) Examples of weak detectors (clockwise from the top left): a shape template feature, representing a segment of the mean contour near the nose of the plane; a boundary fragment generated from an edge chain; a SIFT feature; a filter response patch extracted from a random location within the object bounding box. All features are shown along with their offset vector (*yellow arrow*) that "votes" for the location of a landmark at the tail of the airplane (*pink square*)

A weak detector h_i is a feature of one of the types listed above (e.g., a filter response patch) providing information on the location of landmark i. It is defined in terms of a feature vector of the relevant type and an offset v between the pixel p at which the feature is computed and the landmark position. Thus, if a weak detector h_i produces a high response on a test image at pixel p, then this detector will "vote" for the presence of the landmark at $p + v$. Figure 5 illustrates each type of weak detector together with a sample landmark offset.

A weak detector is applied to an image $\mathcal{I}$ in the following manner. First, the detector's feature is evaluated for each pixel in the image and shifted by the offset for the corresponding landmark i to produce the response image R_i. For our patch features, we make R_i sparse by zeroing out all but the top K (50) responses and blur it to provide a softer voting mechanism, in which small perturbations are not penalized. The sparsification of the response map allows only the most confident feature matches. A more detailed explanation of this process can be found in Murphy et al. (2006). In all experiments, we blur the response image using a Gaussian blur with a standard deviation of 10 pixels. We now let $h_i(p)$ be a regression stump over the response $R_i(p)$ that indicates the probability of the landmark occurring at this pixel.

Once we have computed the responses for each weak detector, we learn the strong detector $H_i(p)$ using Real Adaboost (Schapire and Singer 1999) with regression stumps (the h_i's) as in Torralba et al. (2005), Murphy et al. (2006). For each landmark i, our positive training set consists of the response vectors corresponding to assignments of that landmark in each of the training instances. In practice, because our landmark locations are noisy (due to labeling and correspondence error), we add vectors from a small (3×3) grid of pixels around the true assignment. Our negative set is constructed from the responses at random pixels that are at least a minimum distance (10 pixels) away from the true answer.

5.3 Gradient Features

Our model also includes terms that estimate the likelihood of the edges connecting the landmarks. In general, we expect that the image will contain high gradient magnitudes at the object boundary. For neighboring landmarks i and j with pixel assignments (l_i, l_j), we define the feature:

$$F_{ij}^{\text{grad}}(l_i, l_j; \mathcal{I}) = \sum_{r \in \overline{l_i l_j}} |\mathbf{g}(r)^T \mathbf{n}(l_i, l_j)|,$$

where r varies over all of the points along the line segment from l_i to l_j, $\mathbf{g}(r)$ is the image gradient at point r, and $\mathbf{n}(l_i, l_j)$ is the normal to the edge (between l_i and l_j). High values of the feature encourages the boundary of the object to lie along segments of high gradient.

6 Localization of Object Classes

We now address our central computational challenge: assigning the landmarks of a LOOPS model to test image pixels while allowing for large deformations and articulations. Recall that the conditional MRF defines a distribution (1) over assignments of model landmarks to pixels. This allows us to outline objects by using probabilistic inference to find the most probable such assignment:

$$\mathbf{L}^* = \arg\max_{\mathbf{L}} P(\mathbf{L} \mid \mathcal{I}, \mathbf{w}).$$

Because, in principle, each landmark can be assigned to any pixel, finding $\mathbf{L}^*$ is computationally prohibitive. One option is to use an approach analogous to active shape models, using a greedy method to deform the model from a fixed starting point. However, unlike most applications of active shape/appearance models (e.g., Cootes et al. 1998), our images have significant clutter, and such an approach will quickly get trapped in an inferior local maxima. A possible solution to this problem is to consider a series of starting points. Preliminary experiments along these lines (not shown for lack of space), however, showed that such an approach requires a computationally prohibitive number of starting points to effectively localize even rigid objects. Furthermore, large articulations were not captured even with the "correct" starting point (placing the mean shape in the center of the true location).

To overcome these limitations, we propose an alternative two step method, depicted in Fig. 6: we first approximate our problem and find a coarse solution using discrete inference; we then refine our solution using continuous optimization and the full objective defined by (1).

Discrete Inference Obviously, we cannot perform inference over the entire space as even a modestly sized image (200×300 pixels) results in a search space of size $N^{60,000}$ where N is the number of model landmarks. Thus, we start with pruning the domain of each landmark to a relatively small number of pixels. To do so both effectively and efficiently (without requiring inference), we first assume that landmarks will fall on "interesting" points in the image, and consider only points found by the SIFT interest operator (Lowe 2003). We adjust the settings of the SIFT interest operator to produce between 1000 and 2000 descriptors per image (we set the number of scales to 10), allowing us to have a large number of locations to choose from. Following this step, we make use of the MRF appearance based feature functions F_i^{det} to identify the most promising candidate pixel assignments for each landmark. While we cannot expect these features to identify the single correct pixel for each landmark, we might expect the correct answer to lie towards the higher end of the response values. Thus, for each

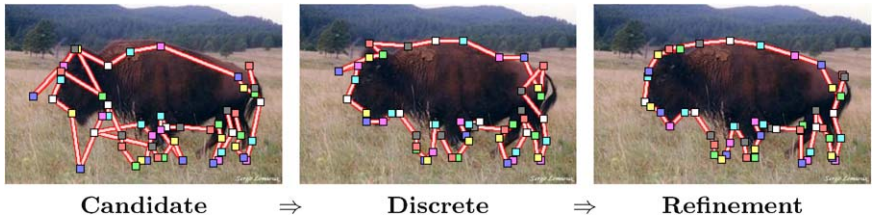


Fig. 6 (**a**) An outline defined by the top detection for each landmark independently, (**b**) an outline inferred by inference over our discrete conditional MRF, and (**c**) a refined outline after coordinate-ascent optimization

landmark, we use the corresponding F_i^{det} to score all the SIFT keypoint pixels in the image and choose the top 25 local optima as candidate assignments. Figure 6(a) shows the top assignment for each landmark according to the detectors for a single example test image.

Given a set of candidates for each landmark, we must now find a single, consistent joint assignment $\mathbf{L}^*$ to all landmarks together. However, even in this pruned space, the inference problem is quite daunting. Thus, we further approximate our objective by sparsifying the multivariate Gaussian shape distribution to include only terms corresponding to neighboring landmarks plus a linear number of terms (see Sect. 5.1). The only pairwise feature functions we use are over neighboring pairs of landmarks (as described in Sect. 5.3), which does not add to the density of the MRF construction, thus allowing the inference procedure to be tractable.

Finally, we find $\mathbf{L}^*$ by performing approximate max-product inference, using the recently developed and empirically successful Residual Belief Propagation (RBP) of Elidan et al. (2006b). Figure 6(b) shows an example of an outlines found after the discrete inference stage.

Refinement Given the best coarse assignment $\mathbf{L}^*$ predicted in the discrete stage, we perform a refinement stage in which we reintroduce the entire pixel domain and use the full shape distribution. This allows us to more accurately adapt to the image statistics while also considering shapes that fit a better distribution than was available at the discrete stage. Refinement is accomplished using a greedy hill-climbing algorithm in which we iterate across each landmark, moving it to the best candidate location using one of two types of moves, while holding the other landmarks fixed. In a *local* move, each landmark can pick the best pixel location in a small window around its current location. In a *global* move, each landmark can move to its mean location given all the other landmark assignments. Determining the conditional mean location is straightforward given the Gaussian form of the joint shape model. If we seek the conditional mean of the location of landmark n, denoted x_n, given the locations of landmarks $1 \ldots n-1$, denoted by $\mathbf{x}^-$, we begin with the joint distribution:

$$\begin{bmatrix} \mathbf{x}^- \\ x_n \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu^- \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma^- & \beta_n^- \\ \beta_n^{-T} & \sigma_n^2 \end{bmatrix} \right).$$

In this case, the conditional mean is given by

$$E[x_n \mid \mathbf{x}^-] = \mu + \beta_n^{-T} \left(\Sigma^-\right)^{-1} \left(\mathbf{x}^- - \mu^-\right).$$

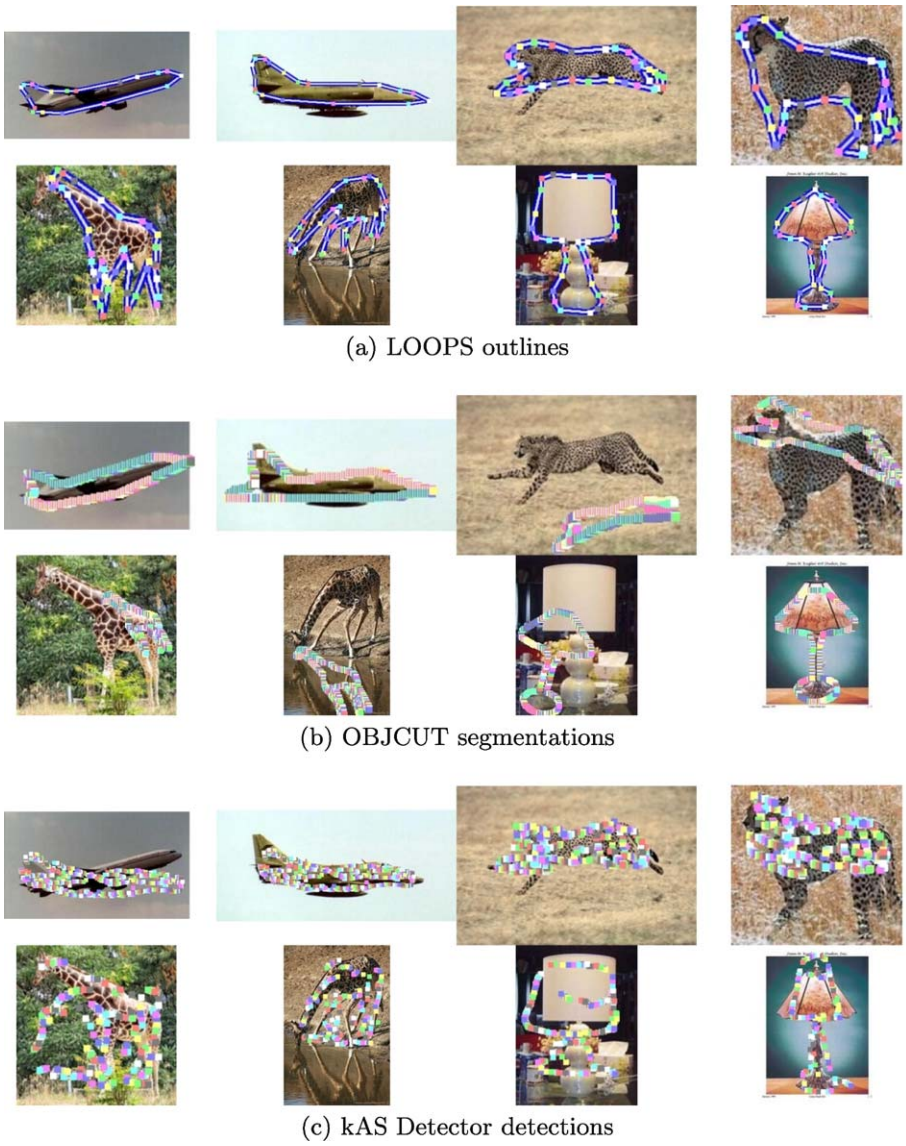
Interestingly, in a typical refinement, the *global* moves dominate the early iterations, correcting large mistakes made by the discrete stage and that resulted in an unlikely shape. In the later iterations, *local* moves do most of the work by carefully adapting to the local image characteristics. Figure 6(c) shows an example of an outline found after the refinement stage.

7 Experimental Evaluation of LOOPS Outlining

We begin with a quantitative evaluation of the accuracy of the loops outlines. In particular, we evaluate the ability of our model to produce accurate outlines in which the model's landmarks are positioned consistently across test images. In the following section, we will demonstrate the range of capabilities of the LOOPS model by showing how these correspondences can be used for a series of descriptive tasks.

As mentioned in Sect. 3, there are existing methods that also seek to produce an accurate outline of object classes in cluttered images. In order to evaluate the LOOPS outlines, we compare to two such state-of-the-art methods. The first is the OBJCUT model of Prasad and Fitzgibbon (2006). Briefly, this method uses an exemplar-based shape model of the object class together with a texture model to find the best match of exemplar to the image. The second method we compare to is that of Ferrari et al. (2008). This approach uses adjacent contour segments as features for a detector. Note that unlike both OBJCUT and our LOOPS method, this method only requires bounding box supervision for the training images rather than full outlines. In order to put this method on equal footing with the others, we also use a version of this method that sees the "ideal" edgemap of the training images, which is derived from the manual outline. Under this setting, the learned contours are clean sections of the true outline. In order to differentiate, we call the original method the *kAS Detector* (bounding box), and the supervised outline version is *kAS Detector* (outline). The matched contour segments for each detection are returned as the detected shape of the object.

We compare LOOPS to these three methods on four object classes: 'giraffe', 'cheetah', 'airplane', and 'lamp'.



(a) LOOPS outlines

(b) OBJCUT segmentations

(c) kAS Detector detections

Fig. 7 Example outlines for the three methods. The full set of resulting outlines for each method can be found at http://ai.stanford.edu/~gaheitz/Research/Loops

Images for each class were downloaded from Google images and were chosen to exhibit a range of articulation and deformation, which will be addressed the experiments below. Randomly selected example outlines are shown in Fig. 7, and the full dataset can be found at http://ai.stanford.edu/~gaheitz/Research/Loops along with all images used in this paper. Both competing methods were updated to fit our data with help from the original code developers (P. Kumar, V. Ferrari; personal communications). The code and parameters were adjusted through rounds of communication until they were satisfied that the methods had achieved reasonable results. We thank them for their friendly and helpful communications.

In all experiments, we trained our loops model on 20 randomly chosen training images and tested on the others. We report average results over 5 random train/test partitions. The features used to construct the boosted classifier for each landmark (see Sect. 5.2) include shape templates of lengths 3, 5, 7, 10, 15, 20, 30, 50, 75 for each landmark as well as 400 boundary fragments, 400 SIFT features and 600 filter features that are shared between all landmarks.

In order to quantitatively evaluate these results, we measured the symmetric root mean squared (rms) distance between the produced outlines and the hand-labeled groundtruth. These numbers are reported in Table 1. Based on this metric, LOOPS clearly produces more accurate outlines than the competing methods. In addition to these evaluations, we include a quantitative analysis for the descriptive experiments in Sect. 8.

In our data, we know the class of the object in each image, and the object is generally in the middle of the image, covering a majority of the pixels. The *kAS Detector*, however, was originally developed for object detection, which is often unnecessary in our data. Due to its success as a de-

tector, its inclusion as a pre-processing stage to the LOOPS outlining is a promising direction for data for which detecting the object is more difficult. Indeed, our experiments have shown that LOOPS struggles with the pure detection task on more difficult detection data for which the *kAS Detector* was designed. We tested LOOPS as a detector on the data of Ferrari et al. (2007), and found that a *kAS Detector* learned from only bounding box annotations outperforms the LOOPS detector by 11%, averaged across the 5 classes. According to Ferrari et al. (2007), the *kAS Detector* achieves detection rates (at 0.4 false positives per image) of 84%, 83%, 58%, 83%, and 75% for the classes 'mug', 'bottle', 'giraffe', 'applelogo', and 'swan', respectively. LOOPS achieves corresponding detection rates of 70%, 61%, 83%, 75%, and 83%. Note that LOOPS does outperform the *kAS Detector* on the 'giraffe' and 'swan' classes, which have the most distinct shape.

While in some cases producing outlines is sufficient, there are also cases where we care about the precision of the localized landmarks. In the next section we show experiments that use the localized landmarks for classification. Towards this goal, we now evaluate the accuracy of the model landmarks in test images. We consider the 'airplane','bass','buddha' and 'rooster' classes from the Caltech 101 dataset (Fei-Fei et al. 2004) as well as the significantly more challenging 'bison', 'deer', 'elephant', 'giraffe', 'llama' and 'rhino' classes from the mammal dataset of (Fink and Ullman 2007).[1] These images have more cluttered backgrounds than the Caltech images as well as foreground objects that blend into these backgrounds.

To measure our ability to accurately localize landmarks, we need a groundtruth labeling for the landmarks that is corresponded with our model. Thus, for the purposes of *this evaluation only*, we did not use our automatic method for corresponding training outlines, but rather labeled the Caltech and Mammal classes with manually corresponded landmarks. We then train a LOOPS model on these highly-supervised instances and evaluate our results using a scale-independent landmark success rate: a landmark is successfully localized if its distance from the manually labeled location is less than 5% of the diagonal of the object's bounding box in pixels.

Figure 8 compares the landmark success rates of our LOOPS model to two baseline approaches. In the first, we pick each landmark location using its corresponding detector by assigning the landmark to the pixel location in the image with the highest detector response. This *Landmark* approach uses no shape knowledge beyond the vote offset that is encoded in the weak detectors. As a second baseline, we create a single boosted *Centroid* detector that attempts to localize the center of the object, and places the landmarks relative to the predicted centroid using the mean object shape.

The evaluation of error on a per landmark basis is particularly biased in favor of the *Landmark* method, which is trained specifically to localize each landmark, without attempting to also fit the shape of the object. Nevertheless, the relative advantage of the LOOPS model, which takes into account the global shape, is evident. We note that while

Table 1 Normalized symmetric root mean squared (rms) distance between the produced outline and the hand-labeled groundtruth, for the three competitors. Outlines are converted into a high-resolution point set, and the number reported is the rms of the distance from each point on the outline to the nearest point on the groundtruth (and vice versa), as a percentage of the groundtruth bounding box diagonal. Note that while the LOOPS and OBJCUT methods require full object outlines for the training images, the kAS Detector can be trained with only bounding boxes supervision (fourth column) or with full supervision (fifth column)

Class	LOOPS	OBJCUT	kAS Detector (bounding box)	kAS Detector (outline)
Airplane	1.9	5.5	3.8	3.6
Cheetah	5.0	12.3	11.7	10.5
Giraffe	2.9	10.5	8.7	8.1
Lamp	2.9	7.3	5.8	5.3

[1] Classes were selected based on the number of images available. For classes with almost enough images, we augmented the dataset with additional ones from Google images. All images can be found at http://ai.stanford.edu/~gaheitz/Research/Loops.

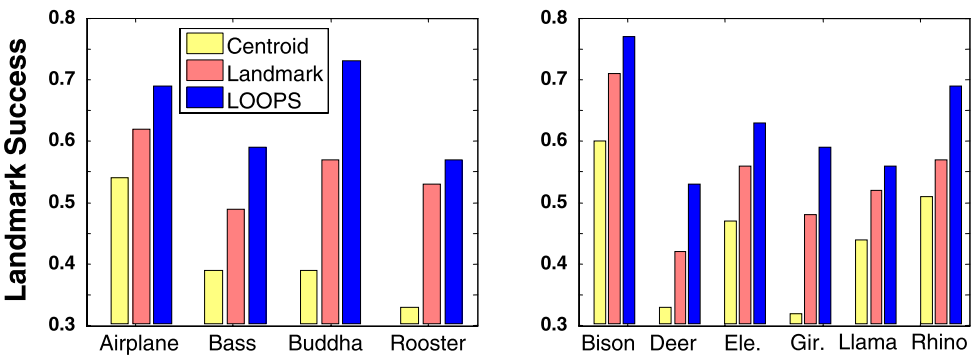


Fig. 8 Landmark success rates for the Caltech and Mammal classes. Shown is the average fraction of landmarks that were localized within 5% of the bounding box diagonal from the groundtruth for our *LOOPS* method as well as the *Centroid* and *Landmark* baselines

Table 2 Normalized symmetric root mean squared (rms) distance between the LOOPS outline and the hand-labeled groundtruth

Caltech	Centroid	Landmark	LOOPS	
			Discrete	Refined
Airplane	2.6	2.0	1.6	1.5
Bass	5.9	4.3	4.0	4.1
Buddha	7.5	4.8	4.1	4.0
Rooster	6.5	4.7	3.8	3.8
Mammals				
Bison	2.7	2.7	2.0	2.0
Deer	5.9	6.6	4.5	4.4
Elephant	3.3	3.3	2.5	2.5
Giraffe	7.1	4.5	3.4	3.3
Llama	3.7	3.2	3.5	3.1
Rhino	3.3	3.1	2.6	2.4

other works quantitatively evaluate the accuracy of their *outlines* (e.g., Ferrari et al. 2008), our evaluation explicitly measures the accuracy of the localized correspondences.

Figure 6 provides a representative example of the localization at each stage of our algorithm. It illustrates how the discrete inference prediction (b) builds on the landmark detectors (a) but also takes into account global shape. Following a reasonable rough landmark-level localization, our refinement is able to further improve the accuracy of the outline (c).

To get an absolute sense of the quality of our corresponded localizations, Table 2 shows the symmetric root mean squared distance between the predicted and groundtruth outlines. The improvement over the baselines is obvious and most errors are on the order of 10 pixels or less, a sufficiently precise localization for images that are 300 pixels in width. We note that our worst performance for the 'buddha' class is in part due to inconsistent human labeling that includes the base of the statue in some images but not in others. Figure 9 provides a qualitative sense of the outlines predicted by LOOPS, showing several examples from each of the Mammal classes and Fig. 10 shows several examples from each of the Caltech classes.

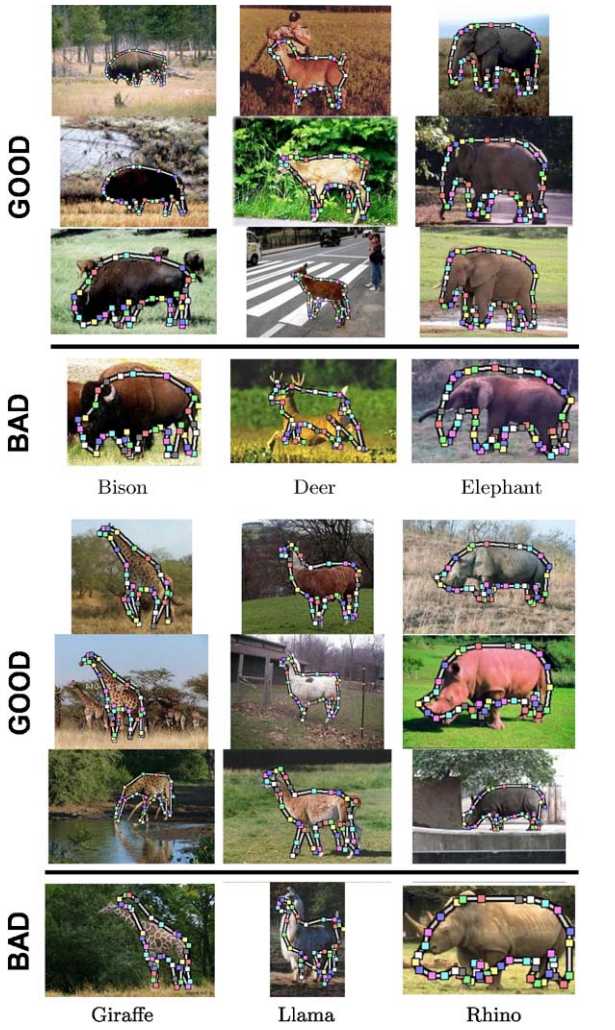


Fig. 9 Example LOOPS localizations from the Mammal classes. The *top* three for each class are successful localizations, while the *fourth* (*below the line*) is a failure. The full set of resulting outlines can be found at http://ai.stanford.edu/~gaheitz/Research/Loops

8 Descriptive Queries with LOOPS Outlines

Recall that our original goal was to perform descriptive queries on the image data. In this section we consider tasks that attempt to explore the space of possible applications facilitated by the LOOPS model. Broadly speaking, this space varies along two principal directions. The first direction concerns variation of the machine learning application. We will present results for classification, search (ranking), and a clustering application. The second direction varies the components that are extracted from the LOOPS outlines for these tasks. We will show examples that use the entire object shape, those that use a part of the object shape, and those that use the local appearance of a shape-localized part of the object. While each particular example shown below might be accomplished by other methods, the LOOPS framework allows us to approach any of these tasks with the same underlying object model.

8.1 User-Defined Landmark Queries

We begin by allowing the user to query the objects in our images using refined shape-based queries. In particular, we use an interface in which the user selects one or more landmarks in the model and provides a query about those landmarks in test images. For example, to answer the question "Where is the giraffe's head?" the user might select a landmark in the

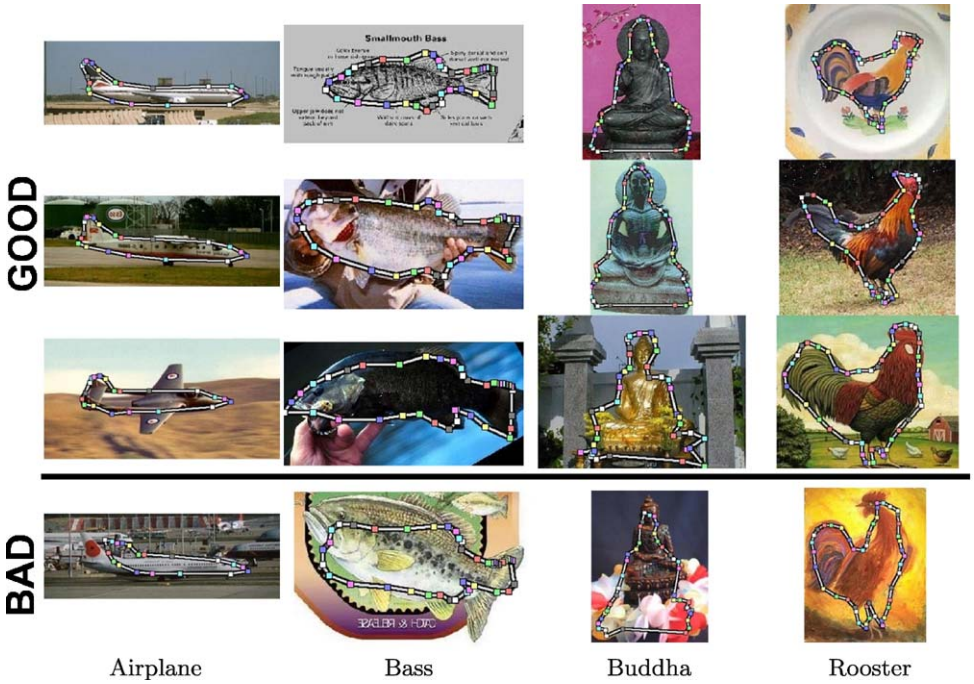


Fig. 10 Example LOOPS localizations for the Caltech classes. The *top* three for each class are successful localizations, while the *fourth* (*below the line*) is a failure. The full set of resulting outlines can be found at http://ai.stanford.edu/~gaheitz/Research/Loops

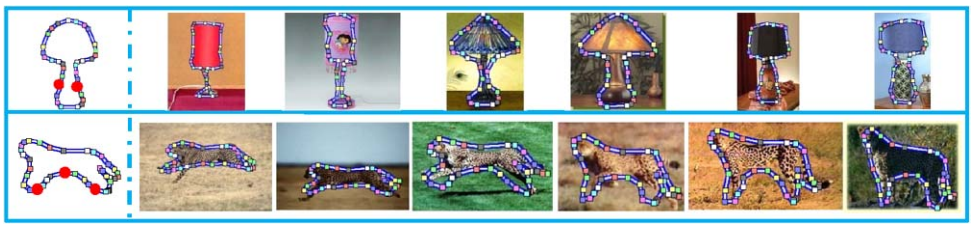


Fig. 11 Refined object descriptions using pairs and triplets of landmarks. In the *top row*, the user selects two landmarks from the model on either side of the lamp base (*highlighted on the left*) and asks for the distance between them in test images. Results are displayed in order of this distance increasing from left to right. In the *bottom row*, the user selects a landmark on the front and hind legs of the cheetah as well as a landmark on the stomach in between them and asks for the angle formed by the three points. Test images are displayed in order from widest angle on the left to smallest angle on the right

head and ask for its location in test images. The user might also ask for the distance between two landmarks, or the angle between three landmarks. Figure 11 shows how this can be used to find images of lamps whose bases vary in width, as well as cheetahs with varying angles of their legs. While one might be able to engineer a task-specific solution for any particular query using another method, the localization of corresponded landmarks of the LOOPS model allows a range of such queries to be addressed in a natural and simple manner.

8.2 Shape-Based Classification

Our goal is to use the predicted LOOPS outlines to distinguish between configurations of an object. To accomplish this, we first train the joint shape and appearance model and perform inference to localize outlines in the test images, all *without* knowledge of the task at hand or any labels. We then incorporate knowledge of the desired *descriptive* task, such as class labels, and perform the task using the corresponded localizations in the test images.

We now show how to perform descriptive classification with minimal supervision in the form of example images. Rather than specifying relationships between particular landmarks, the user simply provides classification labels for a small subset of the instances used to train the LOOPS model (as few as one per class) and the entire outline is used. We present three descriptive classification tasks for three quite different object classes (below we will also consider multiple classification tasks applied to the same object class):

1. Giraffes standing vs. bending down.
2. Cheetahs running vs. standing.
3. Airplanes taking off vs. flying horizontally.

The first two of these tasks depend on the *articulation* of a target object, while the third task demonstrates the ability of LOOPS to capture the *pose* of an object. With these experiments we hope to demonstrate two points. The first is that an accurate shape allows for better classification than using image features alone. To show this, we compare to generative and discriminative classifiers that operate directly on the image features. Our second point is that the LOOPS model in particular produces outlines that are better suited to these tasks than existing methods that produce object outlines. We compare to two existing methods and show that the classifications produced by LOOPS are superior.

We begin our comparison with two baseline methods that operate directly on the image features. The first baseline is a generative *Naive Bayes* classifier that uses a SIFT dictionary for constructing features. For each SIFT keypoint for each image, we create a vector by concatenating the location, saliency score, scale, orientation and SIFT descriptor. Across the training images, we cluster these vectors into bins using K-means clustering, thus creating a SIFT-based dictionary (we chose 200 bins as this achieves the best overall classification results). We then train a generative Naive-Bayes model, where, for each label (e.g., standing = 0, bending down = 1), we learn a multinomial over the SIFT descriptor dictionary entries. When all of the training data is labeled, the multinomial for each label may be learned in closed form; when some of the training data is unlabeled, the EM algorithm is used to train the model by filling in the missing labels.

Our second baseline uses a discriminative approach, based on the *Centroid* detector described above, that is similar to the detector used by Torralba et al. (2005). After predicting the centroid of the object, we use the vector of feature responses at that location in order to classify the object. Specifically, we train a second boosted classifier to differentiate between the response vector of the two classes using the labeled training instances. We note that, unlike the generative baseline, this discriminative method is unable to make use of the unsupervised instances in the training set.

When using the LOOPS model, we train the joint shape and appearance model independently of the classification tasks and *without* knowledge of the labels. We then use the labels to train a classifier for predicting the label given a corresponded localization. To do this in a manner that is invariant to scaling, translation and rotation, we first align the training outlines to the mean using the procrustes method (Dryden and Mardia 1998). (In the airplane task, where the rotation is the goal itself, we normalize only for the translation and scale of the instances, while keeping their original orientations.) Once our instances are aligned, we use principal component analysis to represent our outlines as a vector of coefficients that quantify the variation along the most prominently varying axes. We then learn a standard logistic regression classifier over these vectors; this classifier is trained to maximize the likelihood of the training labels given the training outlines. To classify a test image, we align the corresponded outline produced by LOOPS to the mean training outline and apply the classifier to the resulting PCA vector.

In order to get a sense of what contributes to the errors made by the LOOPS method, we also include classification results using the groundtruth outlines of the test instances. In this method, called GROUND in graphs below, the training and test groundtruth outlines are mutually corresponded using our automatic landmark selection algorithm (see Sect. 4). These corresponded outlines are then used in the same way as the LOOPS outputs. This measure shows the signal present in the "ideal" localizations (according to human observers), and serves as a rough indication of how well the LOOPS method would perform if its localizations matched human annotations.

Figure 12 (left column) shows the classification results for the three tasks as a function of the number N of supervised training instances per class (x-axis). For all three tasks, LOOPS + *Logistic* (solid blue) outperforms both baselines. Importantly, by making use of the shape of the outline predicted in a cluttered image, we surpass the fully supervised baselines (rightmost on the graphs) with as little as a single supervised instance (leftmost on the graphs). Furthermore, our classification results were similarly impressive when using both a nearest-neighbor classifier and support vector machine. This indicates that once the data instances have been projected into the correct space (that of the corresponded outline), many simple machine learning algorithms can easily solve this problem. For the airplane and giraffe classes, the LOOPS + *Logistic* method perfectly classifies all test instances, even with a single training instance per class. Our advantage relative to the *Naive Bayes* classifier is least pronounced in the case of the cheetah task. This is not surprising as most running cheetahs are on blurry or blank backgrounds, and a SIFT descriptor based approach is expected to perform well.

Convinced that outlines are superior to image features for these descriptive classification tasks, we now turn to the question of how LOOPS compares to existing methods in producing outlines that will be used for these tasks. We compare to the two existing methods described above, OBJCUT and the *kAS Detector* (both versions). Because these the OBJCUT method has no notion of correspondence between outline points, we will use a classification technique that ignores the correspondences of the *kAS* and LOOPS outlines. In particular, for each produced shape (by all four methods) we rescale the shape to have a bounding-box diagonal of 100 pixels, and center the shape around the origin. We can then compute the chamfer distance between any pair of shapes normalized in this manner.

We build nearest-neighbor classifiers for the descriptive task in question using this chamfer distance as our distance

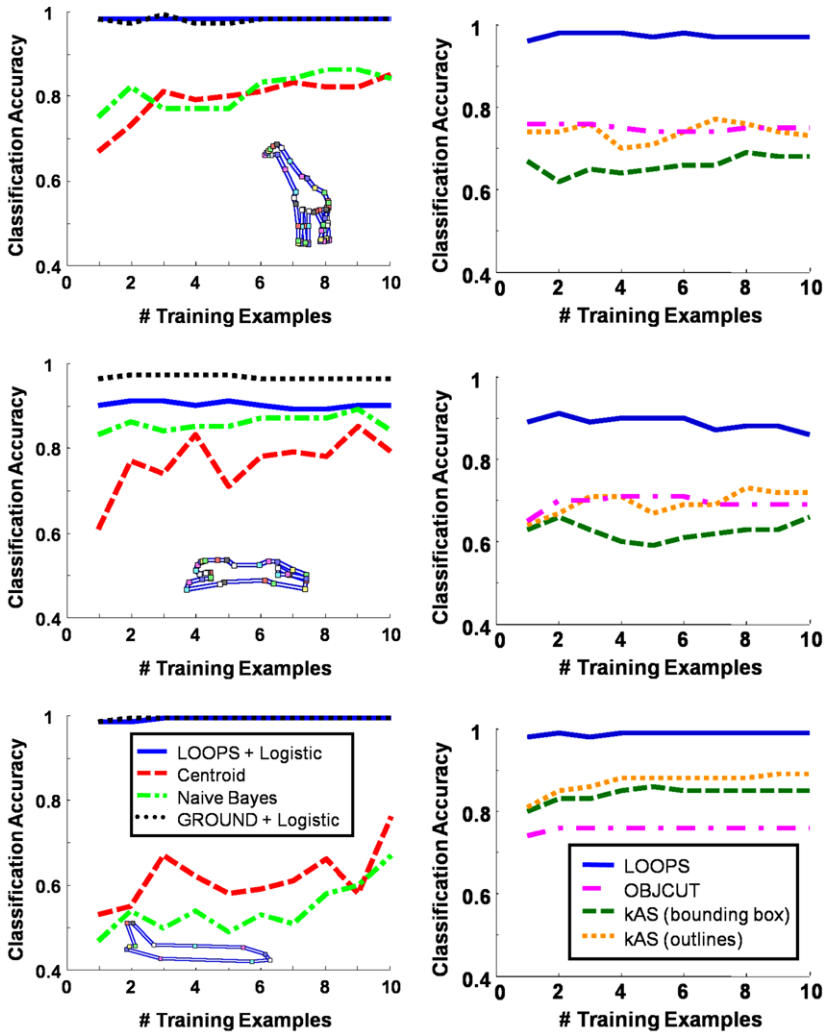


Fig. 12 (*Left column*) Results for three single-axis descriptive classification tasks. Compared are the LOOPs outlining method joined by a logistic regression classifier LOOPS + *Logistic*, a *Naive Bayes* classifier and a discriminative appearance based *Centroid* classifier. We also include the GROUND + *Logistic* results, in which manually labeled outlines from the training and test set are used and that approximately upper bounds the achievable performance when relying on outlines. On both the giraffe and airplane tasks, the combination of LOOPS outlining and a logistic regression classifier (LOOPS + *Logistic*) achieves perfect performance (GROUND + *Logistic*), even with a single training instance for each class. For the cheetah task where the images are blurred, the LOOPS method only comes near to perfect performance and its advantage over the *Naive Bayes* classifier is less pronounced. (*Right column*) A comparison of LOOPS to three other shape-based object detectors. For the purposes of these tasks, the LOOPS outlines are far superior

metric. Classification accuracy for these three methods as a function of the number N of supervised training instances is shown in Fig. 12 (right column). We can see that LOOPS significant outperforms both the OBJCUT and *kAS Detector* for these tasks. For this data, OBJCUT is generally equally as good as the *kAS Detector* with outline supervision, and both tend to outperform the *kAS Detector* with only bounding box supervision.

Once we have our output outlines, an important benefit of the LOOPS method is that we can in fact perform multiple descriptive tasks with the same object model. We demonstrate this with a pair of classification tasks for the lamp object class. The tasks differ in which "part" of the object we consider for classification. In particular, we learn a LOOPS model for table lamps, and consider the following two descriptive classification tasks:

1. Triangular vs. Rectangular Lamp Shade.
2. Thin vs. Fat Lamp Base.

While we do not explicitly annotate a subset of the landmarks to consider, we show that by including a few examples in the labeled set, our classifiers can learn to consider only the relevant portion of the shape. The setup for each task is the same as described in the previous section. We stress that the test localizations predicted by LOOPS are the same for both tasks. The only things that change are the la-

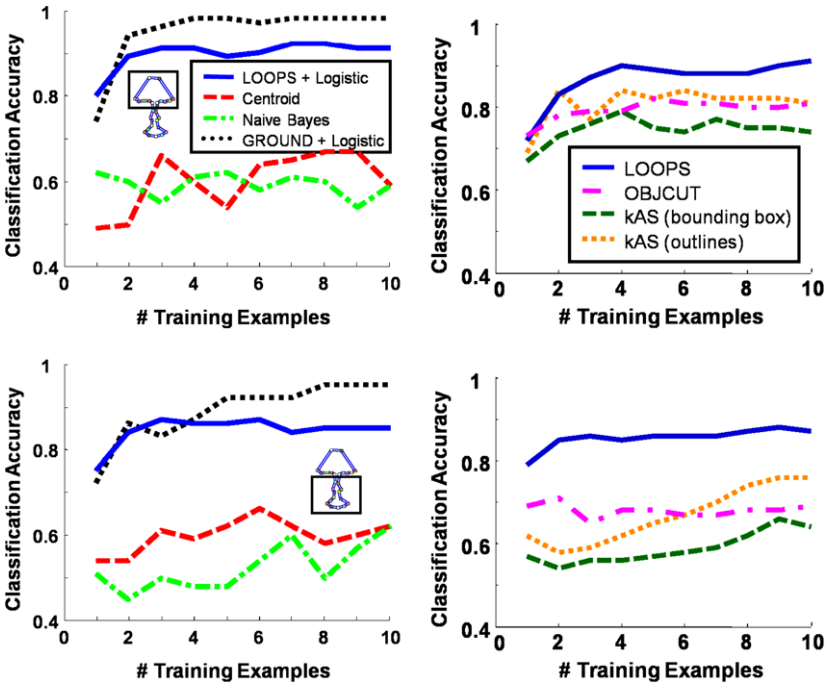


Fig. 13 (*Left column*) Classification results for the two descriptive tasks for the table lamp object class. Compared are the LOOPS outlining method joined by a logistic regression classifier LOOPS + *Logistic*, a *Naive Bayes* classifier and a discriminative appearance based *Centroid* classifier. We also include the GROUND + *Logistic* results, in which manually labeled outlines from the training and test set are used and that approximately upper bounds the achievable performance when relying on outlines. In both tasks, the LOOPS model is clearly superior to the baselines and is not far from perfect (GROUND + *Logistic*) performance. (*Right column*) A comparison of LOOPS to three other shape-based object detectors. For the purposes of these tasks, the LOOPS outlines are far superior

bel set and the logistic regression classifier that attempts to predict labels based on the output localization. In this case, the baseline methods must be completely retrained for each task, which requires EM learning in the case of *Naive Bayes*, and boosting in the case of *Centroid*.

As can be seen in Fig. 13 (left column), along both axes we are able to achieve effective classification, while both feature-based baselines perform barely above random. In addition, we again outperform the shape-based baselines, as can be seen in the right column of Fig. 13. The consequences of this result are promising: we can do most of the work (learning a LOOPS model and predicting corresponding outlines) once, and then easily perform several descriptive classification tasks. All that is needed for each task is a small number of labels and an extremely efficient training of a simple logistic regression classifier. Figure 14 shows several qualitative examples of successes and failures for the above tasks.

We note that in the lamp tasks, the LOOPS + *Logistic* method sometimes outperforms the GROUND + *Logistic* "upper bound" for a small number of training instances. While this seems counterintuitive, in practice a different choice of landmarks or inaccuracies in localizations can lead to fluctuations in the classification accuracy.

8.3 Shape Similarity Search

The second application area that we consider is similarity search, which involves the ranking of test instances on their similarity to a search query. A shopping website, for example, might wish to allow a user to organize the examples in a database according to similarity to a query product. The similarity measure can be any feature that is easily extracted from the LOOPS outline or from the image based on the corresponded outline. We demonstrate similarity using the entire shape, a user-specified component of the shape, and an appearance feature (color) localized to a certain part of the object.

The experimental setup is as follows: offline, we train a LOOPS model for the object class and localize corresponded outlines in each of the test set images. Recall that in LOOPS this is done once and all tasks are carried out given the *same* localized outlines. Online, a user indicates an instance from the test set to serve as a "query" image and a similarity metric to use. We search our database for the images that are most similar based on the specified similarity metric, and return the ranked list back to the user.

As an example, we consider the case of the lamp dataset used above. Users select a particular lamp instance, a subset of landmarks (possibly all), and whether to use shape or

Fig. 14 Example classifications for our five tasks, for the case of one labeled training instance per label. Underneath each image we report the label produced by LOOPS + *Logistic*, and for errors we give the groundtruth label in parentheses. The *first two columns* show a correct labeling for each label, and the *third column* shows a mistake for each task (where available). In *rows three and five* (cheetah and lamp bases), we can see that the mistake is caused by a failure of LOOPS to correctly outline the object. In the *fourth row* (lamp shades), we see that despite a very accurate LOOPS localization, we still incorrectly classify the lamp shade. This is due to the inherent limitations of the logistic approach with a single training instance

color. Each instance in the dataset is then ranked based on Euclidean distance to the query in shape PCA space.

Figure 15 shows some example queries. In the top row we show a full-shape search, where the first (left-most) instance is the query instance. The results are shown left to right in increasing distance from the query instance. For the full-shape query, we see that both triangular lampshades and wide bases are returned. Suppose that the user decides to focus only on the lampshade; the second row shows the results of a search using only the landmarks in the selected region. This search returns lamps with triangular shades, but with bases of varying width. The third row displays the results of a search where the user instead decides to select only the base of the lamp, which returns lamps with wide bases. Finally, the bottom row shows a search based on the color of the shade. In this case we compute the similar-

Fig. 15 (Color online) Object similarity search using the LOOPS output. In the top row we show the top matches in the test database using the full shape of the query object (*left column*). Similarity is computed as the negative sum of the squared distances between the corresponding PCA coefficients of the procrustes aligned shapes. In the *second row* we refine our search by using only the landmarks that correspond to the lamp shade. The results now show only triangular lamp shades. The *third row* shows a search based on the lamp base, which returns wide bases. In the *bottom row*, we use color similarity of the lamp shade to rank the search results. The *top matches* show lamps with faded *yellow* shades

ity between two instances as the negative distance between the mean *LAB* color vectors[2] in the region outlined by the selected landmarks. The top-ranked results all contain off-white shades similar in color to the query image.

Similarity search allows users to browse a dataset with the goal of finding instances with certain characteristics. In all of the examples considered above, by projecting the images into LOOPS outlines, similarity search desiderata were easily specified and effectively taken into account. Importantly, the similarity of interest in all of these cases is hard to specify without a predicted outline.

8.4 Descriptive Clustering

Finally, we consider clustering a database by leveraging the LOOPS predicted outlines. As an example, suppose we have a large database of airplane images, and we wish to group our images into "similar looking" groups of airplanes. Clustering based on shape might produce clusters corresponding to passenger jets, fighter jets, and small propeller airplanes. In this section, we consider an even more interesting outline *and* appearance based clustering where the feature vector for each airplane includes the mean color values in the *LAB* color space for all pixels inside the airplane boundary. We cluster the database of examples on this feature vector using K-means clustering. Figure 16(a) shows one cluster that results from this procedure for a database of 770 images from the Caltech airplanes image set (Fei-Fei et al. 2004). Such results might be obtained from any method that produces a precise outline (or segmentation) of the object.

Imagine, however, that instead of clustering using the appearance for the entire airplane, we are instead interested in the appearance of the tail. This may be a useful scenario because patterns and colors on the tail are generally more informative about the airline or country of origin. In order to focus on the airplane tails, we can specify a subset of the landmarks in the LOOPS model to indicate the region of interest (as in the lamp example above). Since a localization is a correspondence between the model's landmarks and a test image, this will automatically allow us to zoom in on the airplane tails as shown in Fig. 16(b) for the same cluster

[2]The Lab color space represents colors as three real numbers, where the *L* component indicates "lightness" and the *A* and *B* dimensions represent the "color". The space is derived from a nonlinear compression of the XYZ color space.

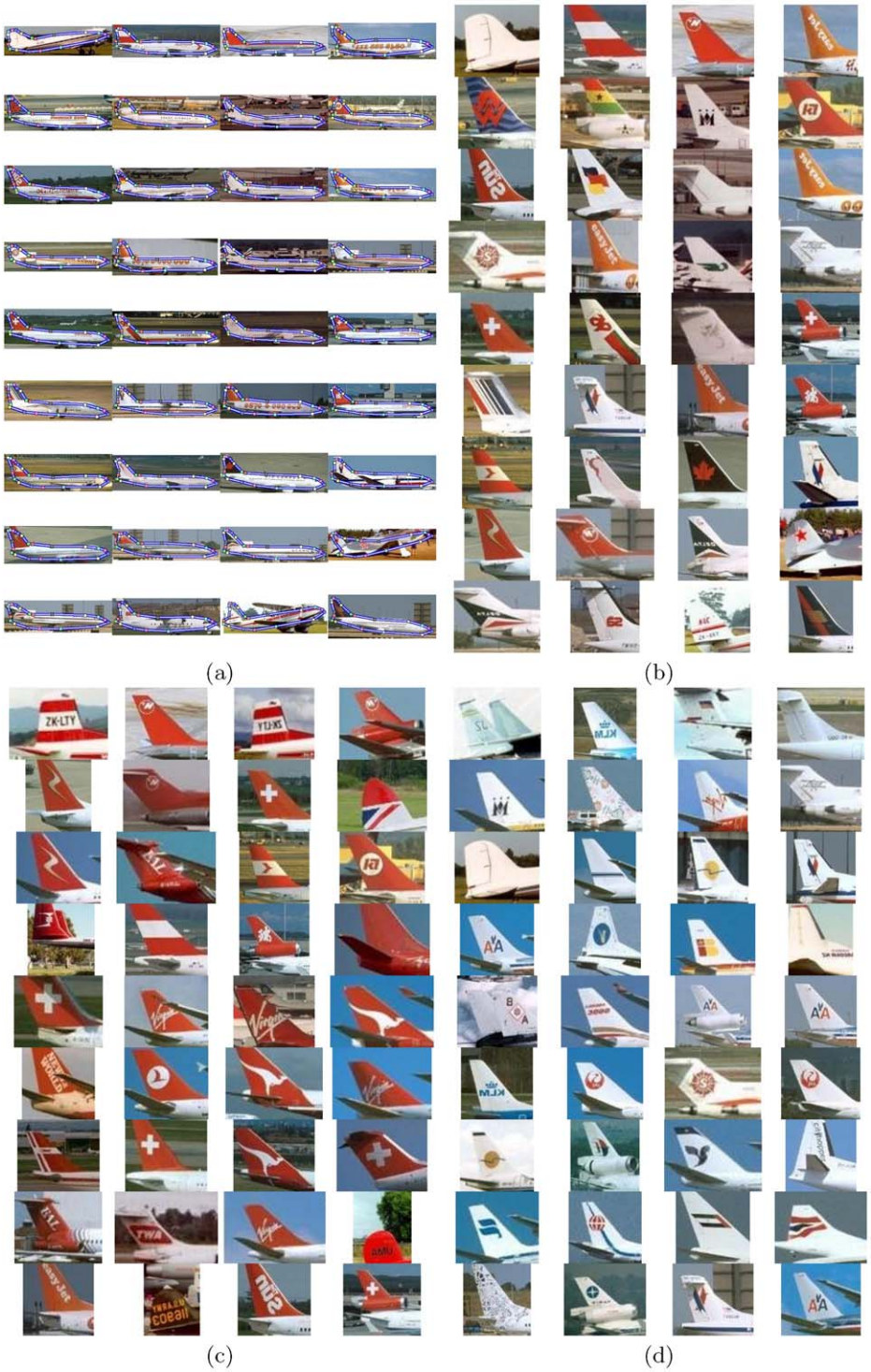


Fig. 16 (Color online) Clustering of a large airplane database. For each cluster we show the top 36 instances based on the distance to the cluster centroid. (**a**) shows an example cluster when using the color features averaged across the entire airplane. (**b**) depicts the zoomed-in tails of the cluster in (a). If we instead cluster using the color of the tails alone, we obtain more coherent groups of airplane tails. (**c**) and (**d**) show the tail clusters that contain the first two examples of (a) and (b)

of Fig. 16(a). Despite the fact that the cluster looks coherent when considering the whole plane, the tails are very heterogeneous in appearance.

The fact that the outlines produced by LOOPS are consistently corresponded allows us to cluster the appearance of the tail itself. That is, LOOPS allows us to rely on the

localized appearance of different parts of the object. Specifying the tail as the landmark subset of interest, we compute the same feature vectors only for the pixels contained in the hull of the tail. We then re-cluster our database using these tail feature vectors. Figures 16(c) and (d) shows the zoomed in tails for the two clusters that contain the first two instances from Fig. 16(a). The coherence of the tail appearance is much more apparent in this case, and both clusters group many tails from the same airlines.

In order to perform such coherent clustering of airplane tails, one needs first to accurately localize the tail in test images. Even more than the table lamp ranking task presented above, this example highlights the ability of LOOPS to leverage *localized appearance*, opening the door for many additional shape and appearance based descriptive tasks.

9 Discussion and Future Work

In this work we presented the Localizing Object Outlines using Probabilistic Shape (LOOPS) approach for obtaining accurate, corresponded outlines of objects in test images, with the goal of performing a variety of descriptive tasks. Our LOOPS approach relies on learning a probabilistic model in which shape is combined with discriminative detectors into a coherent model. We overcome the computational challenge of localizing such a model in cluttered images and under large deformations and articulation by making use of a hybrid discrete-then-continuous optimization approach. We directly and quantitatively evaluated the ability of our method to consistently localize constituent elements of the model (landmarks) and showed how such a localization can be used to perform descriptive classification, shape and localized appearance based search, and clustering that focuses on a particular part of the object. Importantly, we showed that by relying on a model-to-image correspondence, our performance is superior to discriminative and generative competitors often with as little as a single labeled example.

In theory, some existing detection methods (e.g., Ferrari et al. 2008; Berg et al. 2005; Opelt et al. 2006b; Shotton et al. 2005) lend themselves to some of the descriptive tasks described above, by producing outlines during the detection process. However, in experiments above, we demonstrated shortcomings with two state-of-the-art methods in this regard. Because our LOOPS model was targeted towards producing these types of outlines rather than detecting the presence or absence of the object, it was able to obtain significantly more accurate shape-based classifications. Furthermore, in practice, no other work that considered object classes in cluttered images demonstrated a combination of accurate localization and shape analysis that would solve these problems.

Our contribution is thus threefold. First, we design our LOOPS model with the primary goal of corresponded localization in cluttered scenes. The landmarks are chosen automatically so that they will both be salient and appear consistently across instances, and both the training and correspondence steps are geared specifically toward the goal of accurate localization. Second, we present a hybrid global-discrete then local-continuous approach to the computational challenge of corresponding a model to a novel image. This allows us to achieve consistent correspondence in cluttered images even in the face of large deformations that will hinder alternatives such as active shape or appearance models. Third, we demonstrate that accurate localization is of value for a range of descriptive tasks, including those that are based on appearance.

There are several interesting directions in which our LOOPS approach can be extended. We would like to automatically learn coherent parts of objects (e.g., the neck of the giraffe) as a set of landmarks that articulate together, and accurately localize both landmarks and part joints in test images. Intuitively, learning a distribution over part articulation (e.g., the legs of a running mammal are synchronized) can help localization. More importantly, localizing parts opens the door for novel descriptive tasks that consider variation in the number of parts (e.g., ceiling fan with 4 or 5 blades) or in their presence/absence (e.g., cup with and without a handle). Equally exciting is the prospect of prediction at the scene level. A natural extension of our model is a hierarchical variant that views each object detected as a landmark in the higher level scene model. One can imagine how the geometry of such a model could capture relative spatial location and orientations so that we can answer questions such as whether a man is walking the dog, or whether the dog is chasing the man.

Acknowledgements This work was supported by the DARPA Transfer Learning program under contract number FA8750-05-2-0249. We would also like to thank Vittorio Ferrari and Pawan Kumar for providing us code and helping us to get their methods working on our data.

References

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., & Davis, J. (2005). Scape: shape completion and animation of people. In *SIGGRAPH '05: ACM SIGGRAPH 2005 papers* (pp. 408–416). New York: ACM. doi:http://doi.acm.org/10.1145/1186822.1073207.

Basri, R., Costa, L., Geiger, D., & Jacobs, D. (1998). Determining the similarity of deformable shapes. *Vision Research*, *38*, 2365–2385.

Belongie, S., Malik, J., & Puzicha, J. (2000) Shape context: A new descriptor for shape matching and object recognition. In *Neural Information Processing Systems* (pp. 831–837).

Berg, A., Berg, T., & Malik, J. (2005). Shape matching and object recognition using low distortion correspondence. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.

Borenstein, E., Sharon, E., & Ullman, S. (2004). Combining top-down and bottom-up segmentation. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (p. 46). Los Alamitos: IEEE Computer Society. ISBN 0-7695-2158-4.

Borgefors, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(6), 849–865. ISSN 0162-8828. doi:10.1109/34.9107.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Caselles, V., Kimmel, R., & Sapiro, G. (1995). Geodesic active contours. In *International conference on computer vision* (pp. 694–699).

Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models: their training and application. *Computer Vision and Image Understanding*, *61*(1), 38–59. ISSN 1077-3142. doi:10.1006/cviu.1995.1004.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998). Active appearance models. In *European conference on computer vision* (vol. 2, pp. 484–498).

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Crandall, D. J., & Huttenlocher, D. P. (2006). Weakly supervised learning of part-based spatial models for visual object recognition. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Lecture notes in computer science: Vol. 3951. European conference on computer vision* (Vol. 1, pp. 16–29). Berlin: Springer.

Crandall, D., Felzenszwalb, P., & Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In *Proceedings of the 2005 IEEE Computer Society conference on computer vision and pattern recognition (CVPR'05)* (vol. 1).

Cremers, D., Tischhäuser, F., Weickert, J., & Schnörr, C. (2002). Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, *50*(3), 295–313. ISSN 0920-5691. doi:10.1023/A:1020826424915.

Dryden, I., & Mardia, K. (1998). *Statistical shape analysis*. New York: Wiley.

Elidan, G., Heitz, G., & Koller, D. (2006a). Learning object shape: From cartoons to images. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.

Elidan, G., McGraw, I., & Koller, D. (2006b). Residual belief propagation: Informed scheduling for asynchronous message passing. In *Uncertainty in artificial intelligence*.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2000). Efficient matching of pictorial structures. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (pp. 66–73).

Felzenszwalb, P. F., & Schwartz, J. D. (2007). Hierarchical matching of deformable shapes. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 264–271)

Fergus, R., Perona, P., & Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition, San Diego* (Vol. 1, pp. 380–397).

Ferrari, V., Tuytelaars, T., & Van Gool, L. (2006). Object detection by contour segment networks. In *European conference on computer vision (ECCV)*.

Ferrari, V., Jurie, F., & Schmid, C. (2007). Accurate object detection with deformable shape models learnt from images. In *IEEE conference on computer vision and pattern recognition*. IEEE, June 2007. New York: IEEE.

Ferrari, V., Fevrier, L., Jurie, F., & Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(5), 36–51.

Fink, M., & Ullman, S. (2007). From aardvark to zorro: A benchmark for mammal image classification. *International Journal of Computer Vision*, *77*, 143–156.

Grauman, K., & Darrell, T. (2005). Pyramid match kernels: Discriminative classification with sets of image features. In *International conference on computer vision*, October 2005.

Hill, A., & Taylor, C. (1996). A method of non-rigid correspondence for automatic landmark identification. In *Proceedings of the British machine vision conference*.

Hillel, A. B., Hertz, T., & Weinshall, D. (2005). Efficient learning of relational object class models. In *International conference on computer vision* (pp. 1762–1769), Washington, DC, USA. Los Alamitos: IEEE Computer Society. ISBN 0-7695-2334-X.

Kumar, M. P., Torr, P. H. S., & Zisserman, A. (2005). OBJ CUT. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 workshop on statistical learning in computer vision* (pp. 17–32), Prague, Czech Republic, May 2004.

Leordeanu, M., Hebert, M., & Sukthankar, R. (2007). Beyond local appearance: Category recognition from pairwise interactions of simple features. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.

Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *20*, 91–110.

Murphy, K. P., Torralba, A., Eaton, D., & Freeman, W. T. (2006). Object detection and localization using local and global features. In J. Ponce, M. Hebert, C. Schmid, & A. Zisserman (Eds.), *Toward category-level object recognition*. Cambridge: MIT Press.

Opelt, A., Pinz, A., & Zisserman, A. (2006a). Fusing shape and appearance information for object category detection. In *Proceedings of the British machine vision conference*.

Opelt, A., Pinz, A., & Zisserman, A. (2006b). Incremental learning of object detectors using a visual shape alphabet. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 3–10).

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo: Morgan Kaufmann.

Prasad, M., & Fitzgibbon, A. (2006). Single view reconstruction of curved surfaces. In *Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR '06)*, Washington, DC, USA (pp. 1345–1354). Los Alamitos: IEEE Computer Society. ISBN 0-7695-2597-0. doi:10.1109/CVPR.2006.281.

Schapire, R. E., & Singer, Y. (1999). Improved boosting using confidence-rated predictions. *Machine Learning*, *37*(3), 297–336.

Sebastian, T. B., Klein, P. N., & Kimia, B. B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis Machine Intelligence*, *26*(5), 550–571. ISSN 0162-8828. doi:10.1109/TPAMI.2004.1273924.

Sethian, J. (1998). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge: Cambridge University Press.

Shotton, J., Blake, A., & Cipolla, R. (2005). Contour-based learning for object detection. In *International conference on computer vision*.

Thayananthan, A., Stenger, B., Torr, P., & Cipolla, R. (2003). Shape context and chamfer matching in cluttered scenes. In *IEEE Com-*

puter Society conference on computer vision and pattern recognition (CVPR).

Torralba, A., Murphy, K. P., & Freeman, W. T. (2005). Contextual models for object detection using boosted random fields. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. *17*, pp. 1401–1408). Cambridge: MIT Press.

Winn, J., & Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceedings of the 2006 IEEE Computer Society conference on computer vision and pattern recognition (CVPR '06)*, Washington, DC, USA (pp. 37–44). Los Alamitos: IEEE Computer Society. ISBN 0-7695-2597-0.