From images to shape models for object detection

Vittorio Ferrari, Frédéric Jurie, Cordelia Schmid

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

From images to shape models for object detection

Vittorio Ferrari — Frederic Jurie — Cordelia Schmid

N° 6600

July 2008

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--6600--FR+ENG

From images to shape models for object detection *

Vittorio Ferrari , Frederic Jurie , Cordelia Schmid

Thème COG — Systèmes cognitifs
Projet LEAR

Rapport de recherche n° 6600 — July 2008 — 33 pages

Abstract: We present an object class detection approach which fully integrates the complementary strengths offered by shape matchers. Like an object detector, it can learn class models directly from images, and localize novel instances in the presence of intra-class variations, clutter, and scale changes. Like a shape matcher, it finds the boundaries of objects, rather than just their bounding-boxes. This is made possible by a novel technique for learning a shape model of an object class given *images* of example instances. Furthermore, we also integrate Hough-style voting with a non-rigid point matching algorithm to localize the model in cluttered images. As demonstrated by an extensive evaluation, our method can localize object boundaries accurately, while needing no segmented examples for training (only bounding-boxes)

Key-words: object detection, shape matching, contour features

* This research was supported by the EADS foundation, INRIA and CNRS. Vittorio Ferrari was funded by a postdoctoral fellowship of the EADS foundation.

Centre de recherche INRIA Grenoble - Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

D'images au modèles de forme pour la détection d'objets

Résumé : Nous proposons une nouvelle approche pour la détection de catégories d'objets qui intègre les avantages des méthodes de mise en correspondance de formes. D'une part, comme les détecteurs d'objets, notre approche est capable d'apprendre les modèles de classe directement à partir des images et de localiser de nouvelles instances malgré des variations intra-classe, des changements d'échelle ou la présence d'un fond encombré. D'autre part, tout comme les méthodes de mise en correspondance de formes, elle ne fournit pas simplement une boite englobante, mais parvient à extraire les frontières des objets. Ceci est possible grâce à une nouvelle technique d'apprentissage capable de construire le modèle de forme d'une catégorie d'objets, étant donné des images contenant des instances de cette catégorie. Afin de localiser le modèle dans des images test, nous combinons un algorithme de vote dans un espace de Hough à un algorithme de mise en correspondance de points pour les formes non rigides. Les résultats expérimentaux montrent que les frontières des objets sont localisées avec précision sans utiliser d'images d'apprentissage segmentées, les boites englobants seules suffisent.

Mots-clés : détection d'objets, appariement de formes, caractéristiques de contours

1 Introduction

In the last few years, the problem of learning object class models and localizing previously unseen instances in novel images has received a lot of attention. While many methods use local image patches as basic features [16, 24, 34, 38], recently several approaches based on *contour features* have been proposed [2, 12, 14, 22, 25, 29, 33]. In spite of their potential, even these contour-based methods usually localize objects up to a bounding-box, rather than delineating object outlines. We believe the main reason lies in the nature of the proposed models, and in the difficulty of learning them from *real images* (as opposed to hand-segmented shapes [7, 10, 17, 32]). The models are typically composed of rather sparse collections of contour fragments with a loose layer of spatial organization on top [14, 22, 29, 33]. A few authors even go to the extreme end of using individual edgels as modeling units [2, 25]. In contrast, an *explicit shape model* formed by continuous connected curves completely covering the object outlines is more desirable, as it would naturally support boundary-level localization in test images.

In order to achieve this goal, in this paper we propose an approach which bridges the gap between shape matching and object detection. Classic non-rigid shape matchers [3, 5, 32] produce point-to-point correspondences, but need segmented shapes as input models, and need to be initialized near the object in the test image. In contrast, we build a shape matcher with the input/output behavior of a modern object detector: it learns complete shape models *from images*, and automatically matches them to cluttered images, thereby localizing novel class instances up to their boundaries.

Our approach makes three contributions. First, we introduce a technique for learning the prototypical shape of an object class as well as a statistical model of intra-class deformations, given image windows containing training instances (figure 3a; no segmented shapes are needed). The challenge is to determine which contour points belong to the *class boundaries*, while discarding background and details specific to individual instances (e.g. mug labels). Note how these typically form the majority of points, yielding a poor signal-to-noise ratio. The task is further complicated by intra-class variability: the shape of the object boundary varies across instances.

Second, we localize the boundaries of novel class instances in test images by integrating a Hough-style voting scheme [24, 29, 33] with a non-rigid shape matcher [5]. This combination makes accurate, pointwise shape matching possible even in severely cluttered images, where the object boundaries cover only a small fraction of the contour points (figures 1, 13).

Third, we constrain the shape matcher [5] to only search over transformations compatible with the learned, class-specific deformation model. This ensures output shapes similar to class members, improves accuracy, and helps avoiding local minima.

These contributions result in a powerful system, capable of detecting novel class instances and localizing their boundaries in cluttered images, while training from objects annotated only with bounding-boxes.

After reviewing related work and the local contour features used in the next two sections, we present our shape learning method in section 4, and the scheme for localizing objects in new test images in section 5. Section 6 reports extensive experiments. We evaluate the

Figure 1: *Example object detections returned by our approach (see also figure 13).*

quality of the learned models and quantify localization performance at test time in terms of accuracy of the detected object boundaries. We also compare to previous works for object localization with training on real images [14] and hand-drawings [12]. A preliminary version of this work was published at CVPR 2007 [13].

2 Related works

A major challenge in computer vision is category-level object recognition, the ability of recognizing object instances different from those used to model a category. As pointed out by Hummel [21], shape descriptors are particularly suited for achieving generalization: many categories are characterized by their shape rather than by color or texture. For this reason, many authors have used shape representations [1, 3, 7, 15, 14, 19, 20, 25, 32].

In this section we briefly review some of the most important previous works relevant to this paper, i.e. on shape representation and matching, and their role in the modeling, recognition, and localization of object categories.

Structural descriptors Structural descriptors are one of the earliest representations of object shape. Objects are decomposed into rather complex volumetric parts, such as *generalized cylinders* [27] or *geons* [4], and represented by their spatial relations. However, it is very difficult to generate structural descriptions from images. The resulting descriptions are highly sensitive to the way in which the image is segmented into parts, and the same image may give rise to very different descriptions [36]. As a consequence, recent and successful shape based approaches describe objects by *simpler*, robust and stable features connected in a deformable configuration. These features are generic, such as edge points and lines, and easy to extract, but several of them must be grouped to describe (part of) a given object. A central issue is how simple individual features should be and how to group them. If too simple, they will not be informative enough, and raise the complexity of the matching process. If too complex (like *geons*), they will not be detectable reliably.

Contour features A common framework is to use edge points (*edgels*) grouped into silhouettes. As a silhouette has no internal information nor holes, it can be represented by a single closed curve, and described by simple global descriptors like circularity, eccentricity,

major axis orientation, or bending energy. However, these can only discriminate between very different shapes, so more complex descriptors have been proposed, such as the curvature scale space [28], Fourier descriptors [39], and the medial axis transform [35].

As noticed by Belongie *et al.* [3], silhouettes are fundamentally limited because they ignore internal contours and are difficult to extract. This explains why more recent works represent shapes as loose collections of 2D points [7, 18] or 2D features in general, e.g. [10, 14]. However, more informative structures than individual edgels are preferable, as they simplify matching. Belongie *et al.* [3] propose the *Shape Context*, which captures for each point the spatial distribution of all other points relative to it on the shape. Shape Context is a good semi-local representation and supports recognition by allowing to establish point-to-point correspondences between shapes even under deformations. Leordeanu et al. [25] propose another way to go beyond individual edgels, by encoding relations between all pairs of edgels. Similarly, Elidan *et al.* [10] use pairwise spatial relations between landmark points. Ferrari *et al.* [14] present a family of scale-invariant local shape features formed by short chains of connected contour segments, capable of cleanly encoding pure fragments of an object boundary. They offer an attractive compromise between information content and repeatability, and encompass a wide variety of local shape structures.

Modeling shape categories Edgels and other generic features can be directly used to model any object. An alternative is to learn features adapted to a particular object category. For instance, Shotton *et al.* [33] and Opelt *et al.* [29] automatically learn class-specific boundary fragments, i.e. local edgel groups. Moreover, the fragments are arranged in a star configuration, which is learned as well. In addition to their own local shape, such fragments store a pointer to the object center, enabling object localization in novel images using voting. Other methods [9, 14] achieve the same functionality by encoding spatial organization by tiling object windows, and learning which features and tile combinations discriminate objects from background.

The overall shape model of the above approaches is either (a) a global geometric organization of edge fragments [3, 14, 29, 33]; or (b) an ensemble of pairwise constraints between point features [10, 25]. Global geometric shape models are appealing because of their ability to handle deformations, which can be represented in several ways. The authors of [3] use regularized Thin Plate Splines, found to be effective for modeling changes in biological forms. This generic deformation model can quantify dissimilarity between any two shapes, but does not allow to model shape variations within a specific class. In contrast, Pentland *et al.* [30] learn modes of intra-class deformation of an elastic material using a set of training shapes. The most famous work in this spirit is *Active Shape Models* [7], where the shape model in novel images is constrained to vary only in ways seen in a set of training shapes. The main deformation modes, accounting for most of the total variability over the training set, are learnt using PCA. More generally, non-linear statistics can be used to gain robustness to noise and outliers [8].

A shortcoming of the above methods is the need for clean training shapes. This manual image segmentation quickly becomes intractable when hundreds of categories are considered.

Several researchers have tried to develop *semi-supervised* algorithms not requiring segmented training examples. The key idea is to find combinations of object features repeatedly occurring over the example images. Berg *et al.* [2] suggest to build the model by using pairs of images containing an object instance, by retaining parts matching across several image pairs. A similar strategy is used by [25], which initializes the model using all line segments from a single image and then use many other images to iteratively remove spurious features and add new good features. LOCUS [37] is another model which can be learned in an semi-supervised way, but needs the objects to be aligned in the training images and to occupy most of their surface.

One important limitation common to these approaches is the lack of modeling of intra-class shape deformations, assuming the same shape model is explaining all training images. Moreover, as pointed out by [6, 38], LOCUS is not suited for localizing objects in extensively cluttered test images. Finally, the models learned by [25] are sparse collections of features, rather than explicit shapes formed by continuous connected curves. As a consequence, [25] cannot localize objects up to their (complete) boundaries in test images.

Shape matching Object recognition using shape can be casted as a matching problem: find correspondences between model features and image features (e.g. edgels, groups of segments, shape contexts). This often costly combinatorial problem can be made tractable by accepting sub-optimal matching solutions.

When the shape is not deformable or we are not interested in recovering the deformation but only in localizing the object up to translation and scale, simple strategies can be applied, such as Geometric Hashing [23], Hough Transform [29], or exhaustive search (typically combined with Chamfer Matching [18] or classifiers [14, 33]). In case of complex deformations, the parameter space becomes too large for these simple strategies. Gold and Rangarajan [20] propose an iterative method to simultaneously find correspondences and the model deformation. The sum of distances between model points and image points is minimized by alternating a step where the correspondences are estimated while keeping the transformation fixed, and a step where the transformation is computed while fixing the correspondences. Chui and Rangarajan [5] put this idea in a deterministic annealing framework and adopts Thin Plate Splines as deformation model (TPS). The great advantage of the deterministic annealing formulation is that it elegantly supports a coarse-to-fine search in the TPS transformation space, while maintaining a continuous soft-correspondence matrix. A related framework is adopted by Belongie *et al.* [3], where matching is supported by shape contexts. Depending on the model structure, optimization scheme can be based on Integer Quadratic Programming [2], spectral matching [25] or graph cuts [37].

Our approach in context In this paper, we present an approach for learning and matching shapes which has several attractive properties, bridging the shortcomings of previous works. First of all, we build explicit shape models based on continuous connected curves, which represent the prototype shapes of categories. The training objects need only be annotated by a bounding-box, i.e. no precise segmentation is necessary. Our learning approach

Figure 2: *Local contour features.* [a] *a) three example PAS.* [b] *b) the 12 most frequent PAS types from 24 mug images.*

avoids the pairwise image matching used in previous approaches, and is therefore computationally cheaper and more robust to clutter edgels due to the 'global view' it has by considering all training images at once. Moreover, we model intra-class deformations and enforce them at test time, when matching the model to novel images. Finally, we introduce a two-stage technique that allows the deformable matching of these explicit shape models to extensively cluttered test images, and therefore to accurately localize the complete boundaries of previously unseen object instances.

3 Local contour features

In order to learn and detect shapes in *cluttered* images of real-world scenes, some kind of local contour features are necessary. Before explaining the contributions of this paper, we briefly introduce here the features we employ.

PAS features. We build on the scale-invariant local contour features recently proposed by [14]. Edgels are found by the excellent Berkeley edge detector [26], and then grouped into pairs of connected, approximately straight segments (figure 2a). Informally, two segments are connected if they are adjacent on the same edgel-chain, or if one is at end of an edgel-chain directed towards the other segment. Each pair of segments forms one feature, called a *PAS*, for *Pair of Adjacent Segments*.

A *PAS* feature has a location (mean over the two segment centers), a scale (distance between the segment centers), a strength (average edge detector confidence over the edgels with values in $[0, 1]$), and a descriptor invariant to translation and scale changes. The descriptor encodes the shape of the PAS, by the segments' orientations θ_1, θ_2 and lengths l_1, l_2, and the relative location vector $\mathbf{r}$, going from the center of the first segment to the center of the second (a stable way to derive the order of the segments in a PAS is given in [14]). Both lengths and relative location are normalized by the scale of the PAS.

PAS features are particularly suited to our needs. First, they are robustly detected because they connect segments even across gaps between edgel-chains. Second, as both PAS and their descriptors cover solely the two segments, they can cover pure portion of an object boundary, without including clutter edges which often lie in the vicinity (as opposed to patch descriptors). Hence, PAS descriptors respect the nature of boundary fragments, to be

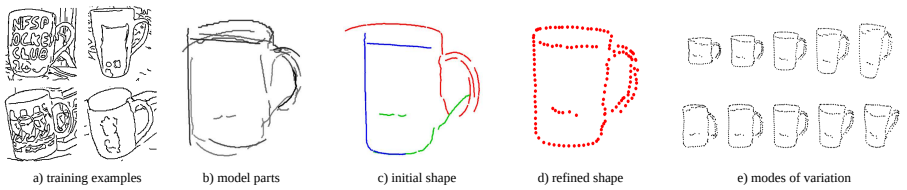


Figure 3: *Learning the shape model. a) 4 out of 24 training examples. b) Model parts. c) Occurrences selected to form the initial shape. d) Refined shape. e) First two modes of variation (mean shape in the middle).*

one-dimensional elements embedded in a 2D image, as opposed to local appearance features, whose extent is a 2D patch. Fourth, PAS have intermediate complexity. As demonstrated in [14], they are complex enough to be informative, yet simple enough to be detectable repeatably across different images and object instances. Finally, since a correspondence between two PAS induces a translation and scale change, they can be readily used within a Hough-style voting scheme for object detection [24, 29, 33].

PAS dissimilarity measure. The dissimilarity $D(P^a, P^b)$ between two PAS P^a, P^b defined in [14] is

$$D(P^a, P^b) = w_r \|\mathbf{r}^a - \mathbf{r}^b\| + w_\theta \sum_{i=1}^{2} D_\theta\left(\theta_i^a, \theta_i^b\right) + \sum_{i=1}^{2} \left|\log\left(l_i^a / l_i^b\right)\right| \quad (1)$$

where the first term is the difference in the relative locations of the segments, $D_\theta \in [0, \pi/2]$ measures the difference between segment orientations, and the last term accounts for the difference in lengths. In all our experiments, the weights w_r, w_θ are fixed to the same values used in [14] ($w_r = 4$, $w_\theta = 2$).

PAS codebook. We construct a codebook by clustering the PAS inside all training bounding-boxes according to their descriptors. For each cluster, we retain the centermost PAS, minimizing the sum of dissimilarities to all the others. The codebook $\mathcal{C} = \{T_i\}$ is the collection of these centermost PAS, the *PAS types* $\{T_i\}$ (figure 2b). A codebook is useful for efficient matching, since all features similar to a type are considered in correspondence.

Note that the codebook is class-specific and built from the same images used later to learn the shape model. In our experiments we obtained better shape models when starting from this codebook, rather than from a universal codebook (as opposed to [14]).

4 Learning the shape model

In this section we present the new technique for learning a prototype shape for an object class and its principal intra-class deformation modes, given image windows with example instances (figure 3a). To achieve this, we propose a procedure for discovering which contour

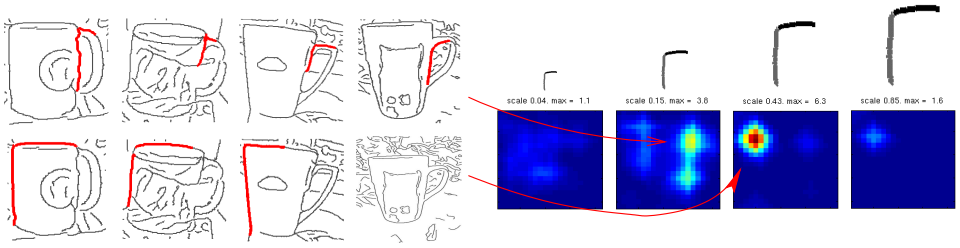


Figure 4: *Finding model parts. On the left, we show a few training instances with two recurring PAS. On the right, we show four slices of the accumulator space for the upper-L shape PAS type (each slice corresponds to a different size). The two recurring PAS form peaks at different locations and sizes. Our method allows for different model parts with the same PAS type.*

points belong to the common *class boundaries*, and for putting them in full point-to-point correspondence across the training examples. For example, we want the shape model to include the outline of a mug, which is characteristic for the class, and not the mug labels, which vary from example to example. The technique is composed of four stages (figure 3b-e):

1. Determine model parts as PAS frequently reoccurring with similar locations, scales, and shapes (subsection 4.1).
2. Assemble an initial shape by selecting a particular feature for each model part from the training examples (subsection 4.2).
3. Iteratively match the shape back onto the training images and refine it using the resulting point-to-point correspondences (subsection 4.3).
4. Learn a statistical model of intra-class deformations from the different shape variations produced by stage 3 (subsection 4.4).

4.1 Finding model parts

The first stage towards learning the model shape is to determine which PAS lie on boundaries common across the object class, as opposed to those on the background clutter and those on details specific to individual training instances. The key insight is that a PAS belonging to the class boundaries will recur consistently across several training instances with a similar location, size, and shape. Although they are numerous, PAS not belonging to the class boundaries are not correlated across different examples. In the following we refer to any PAS or edgel not lying on the class boundaries as *clutter*.

Bounding-box alignment. The first step of the algorithm is to align the training examples by transforming each bounding-box into a zero-centered rectangle with unit height and width equal to the geometric mean of the training aspect-ratios (i.e. width over height).

In addition to removing translation and scale differences, this effectively cancels out shape variations due to different aspect-ratios (e.g tall Starbucks mugs versus coffee cups). This facilitates the learning task, because PAS on the class boundaries are now better aligned.

Voting for parts. The core of the algorithm consists of a voting process. We maintain a separate voting space for each PAS type, in which PAS from all training instances are accumulated. From its limited, local perspective, each PAS votes for the existence of a part of the class boundary with shape, location, and size like its own (figure 4).

More precisely, each PAS P votes as follows. First, it is soft-assigned to all PAS types T_i within a dissimilarity t (equation (1)). Each voting space corresponds to a different PAS type, and has three dimensions: two for location and one for size. For each type the PAS is assigned to, it casts a vote for its own location and size in the corresponding accumulator space. Votes are weighted proportionally to the edge strength e of P, and in inverse proportion to the shape dissimilarity to the type $D(P, T_i)$. The soft-assign to types makes the voting process less sensitive to the exact shape of the PAS and the exact codebook types. Weighting by edge strength allows to take into account the *relevance* of the PAS. It leads to better results over treating edgels as binary features (as also noticed by [9, 12]).

Parts as local maxima. After all PAS have casted their votes, we search for local maxima in location and scale in the accumulator spaces. Each maximum yields a *model part* M, which has a specific location and size relative to the canonical bounding-box, and a specific shape (the codebook type corresponding to the accumulator space where the maximum was found). Moreover, the value of the local maximum provides a measure of the *confidence* that the part really belongs to the class boundaries.

Advantages. The success of this procedure is due in part to adopting PAS as basic shape elements. A simpler alternative would be to use individual edgels. In that case, there would be just one voting space, with two location dimensions and one orientation dimension. In contrast, PAS bring two additional *degrees of separation*: the shape of the PAS, expressed as the assignments to codebook types, and its size (relative to the bounding-box). Individual edgels have no size, and the shape of a PAS is more distinctive than the orientation of an edgel. As a consequence, it is very unlikely that a significant number of clutter PAS will accidentally have similar locations, scales and shapes at the same time. Hence, recurring PAS stemming from the desired class boundaries tend to form peaks in the accumulator spaces, whereas background clutter and details of individual training instances don't.

Intra-class shape variability is addressed partly by the soft-assign of PAS to types, and partly by applying a substantial spatial smoothing to the accumulator spaces before detecting local maxima. This effectively creates wide basins of attraction for PAS from different training examples to accumulate evidence for the same part. We can afford this flexibility while keeping a low risk of accumulating clutter because of the high separability discussed above, especially due to separate accumulator spaces for different codebook types. This

yields the discriminativity necessary to overcome the poor signal-to-noise ratio, while allowing the flexibility necessary to accommodate for intra-class shape variations.

The proposed algorithm sees all training data *at once*, and therefore reliably selects parts and robustly estimates their locations/scales/shapes. In our experiments this was more stable and more robust to clutter than matching pairs of training instances and combining their output a posteriori. As another important advantage, the algorithm has complexity *linear* in the number of training instances, so it can learn from large training sets efficiently.

4.2 Assembling the initial model shape

The collection of parts learned in the previous section captures class boundaries well, and conveys a sense of the shape of the object class (figure 3b). The outer boundary of the mug and the handle hole are included, whereas the label and background clutter are largely excluded. Based on this 'collection of parts' model (COP) one could already attempt to detect objects in a test image, by matching parts based on their descriptor and enforcing their spatial relationship. This could be achieved in a way similar to what earlier approaches do based on appearance features [16, 24], and also done recently with contour features by [29, 33].

However, the COP model is not a shape. It is a loose collection of fragments not connected into a whole shape made of smooth, continuous lines. This can be seen in the multiple stokes along what should be a single line, such as on the top of the mug in figure 3b. Moreover, adjacent parts don't fit together in terms of their relative location and size. The COP model is missing a notion of shape on the global scale. Individual parts are learnt *independently*, each focusing on its own local scale. Going all the way to a proper whole shape is desirable, because it can be used to perform object detection using effective shape-matching techniques. This brings the advantage which motivates this paper: localize objects up to their boundaries, rather than up to a bounding box. In this subsection and the next we describe a procedure for constructing such a shape.

Shape variants. A model part occurs several times on different training images (figure 5a-b). These occurrences offer roughly similar, yet different alternatives for the part's location, size, and shape. Hence, we can assemble several variants for the overall shape by selecting different occurrences for each part. The idea is to select an occurrence for each part so as to form larger aggregates of *connected* occurrences (figure 3c). This can be achieved by assembling occurrences coming from only a few images, since occurrences from the same image fit together naturally.

In this section we cast the problem of assembling a model shape as a combinatorial optimization problem. We introduce the concept of connectedness between model parts and between occurrences, and then select one occurrence for each part so as to obtain a well connected shape. In the following we start with a few definitions necessary to set up the optimization problem, then give the objective function, and present a simple way to maximize it.

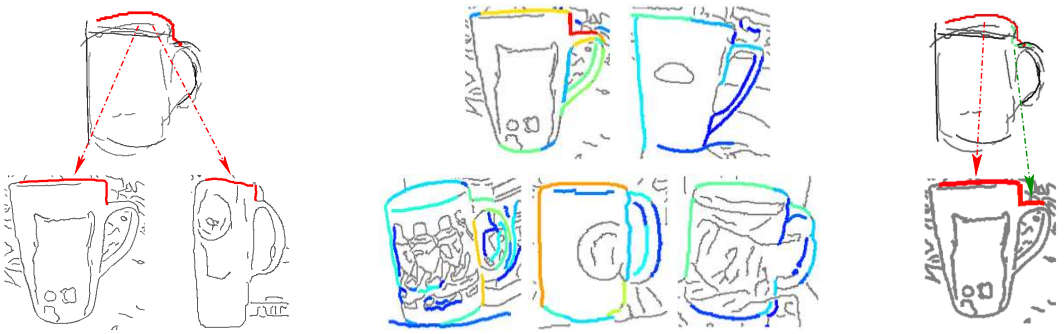


Figure 5: *Occurrences and connectedness. a) A model part (above) and two of its occurrences (below). b) All occurrences of all model parts on a few training images, colored by their confidence (increasing from blue to cyan to green to yellow to red). c) Two model parts with high connectedness (above) and two of their occurrences, which share a common segment (below).*

Occurrences. A PAS P from a specific training image is an occurrence of a model part M if they have similar shape, location, and scale (figure 5a). The following function quantifies this similarity, and expresses the *confidence* that P is an occurrence of M (denoted $M \rightarrow P$):

$$\text{conf}(M \rightarrow P) = P_{str} \cdot \text{D}(M,P) \cdot \min\left(\frac{M_{scale}}{P_{scale}}, \frac{P_{scale}}{M_{scale}}\right) \cdot e^{\left(-\frac{1}{2\sigma_{loc}^2}\|M_{loc}-P_{loc}\|^2\right)} \quad (2)$$

Essentially, the occurrences of M are the PAS that voted for it in section 4.1. The confidence (2) takes into account P's relevance (edge strength, first factor), as well as how far P is from the peak in the voting space (second to last factors). Function (2) ranges in $[0,1]$, and we consider P an occurrence of M if $\text{conf}(M \rightarrow P) > thresh$.

Connectedness. Recall from section 4.1 that a model part M has a specific PAS type and a location and size within the canonical bounding-box. Hence, M has two contour segments M_1, M_2. An important observation is that two occurrences of different model parts might have a segment in common (figure 5c). This provides valuable evidence that the two parts explain connected pieces of the true class boundaries and hence should be connected in the model shape. Since every model part occurs in several training images, we can robustly estimate how likely they are to be connected, by counting how frequently their occurrences share segments. More formally, let the equivalence between two segments M_i, N_j from different model parts M, N be

$$\text{eq}(M_i, N_j) = \sum_{\{s | M_i \rightarrow s, N_j \rightarrow s\}} \text{conf}(M_i \rightarrow s) + \text{conf}(N_j \rightarrow s) \quad (3)$$

with $i,j \in \{1,2\}$ and s a segment on which both M_i, N_j occur (figure 5c). Two model segments have high equivalence if they frequently occur on the same training segments. Based on this, we define the connectedness between two model parts M, N as

$$\text{conn}(M,N) = \max(\text{eq}(M_1,N_1) + \text{eq}(M_2,N_2), \text{eq}(M_1,N_2) + \text{eq}(M_2,N_1)) \quad (4)$$

which is the combined equivalence of their segments (for the best of the two possible segment matchings). Two parts have high connectedness (4) if their occurrences frequently share a segment. Two parts can even share *both* their segments. In this case, their connectedness is even higher, suggesting they explain the same portion of the class boundaries. Equivalent model segments cause the multiple strokes in the COP model (figure 3b). Equation (4) estimates part connectedness robustly because it integrates evidence over all training images.

Objective function. The occurrence selection task can now be formulated precisely: find the assignment $\mathcal{A}(M) = P$ of occurrences to parts that maximizes the following objective function:

$$\sum_{M} \text{conf}\left(M \to \mathcal{A}(M)\right) + \lambda_{conn} \sum_{M,N} \text{conn}(M,N) \cdot \mathbf{1}_{img}\left(\mathcal{A}(M), \mathcal{A}(N)\right) - \lambda_{img} N_{img} \qquad (5)$$

where the indicator function $\mathbf{1}_{img}$ takes value 1 if two occurrences come from the same image, and 0 otherwise. N_{img} is the number of images contributing occurrences to $\mathcal{A}$, and $\lambda_{img}, \lambda_{conn}$ are predefined weights. Exactly one occurrence is assigned to each part.

The first term of the objective function prefers high confidence occurrences. The second favors assigning parts with high connectedness to occurrences from the same image. This pushes towards assigning connected occurrences to connected parts, because occurrences of parts with high connectedness are likely to be connected themselves if they lie in the same image (by construction of function 4). The last term discourages scattering occurrences across many images, as occurrences from the same image fit together naturally. This holds even for occurrences which are not directly connected, but lie on different portions of the shape. Hence, the last term encourages assignments concentrating on a few training images.

Overall, function (5) encourages the formation of aggregates of good confidence *and* properly connected occurrences. In addition, the second and third terms push equivalent model segments to be assigned to the same actual image segments, hence suppressing multiple strokes.

Optimization. It is computationally expensive to optimize the objective function (5) exactly, as the space of all possible assignments is huge. In practice, the following approximation algorithm brings satisfactory results. We start by assigning the part with the single most confident occurrence. Next, we iteratively consider the part most connected to those assigned so far, and assign it to the occurrence which maximizes (5). The algorithm iterates until all parts have been assigned to an occurrence

Figure 3c shows the selected occurrences for our running example. The shape is composed of three blocks, each from a different training image. Within each block, segments fit well together and form continuous lines. Nevertheless, there are discontinuities between blocks, and some redundant strokes remains (lower half of the handle).

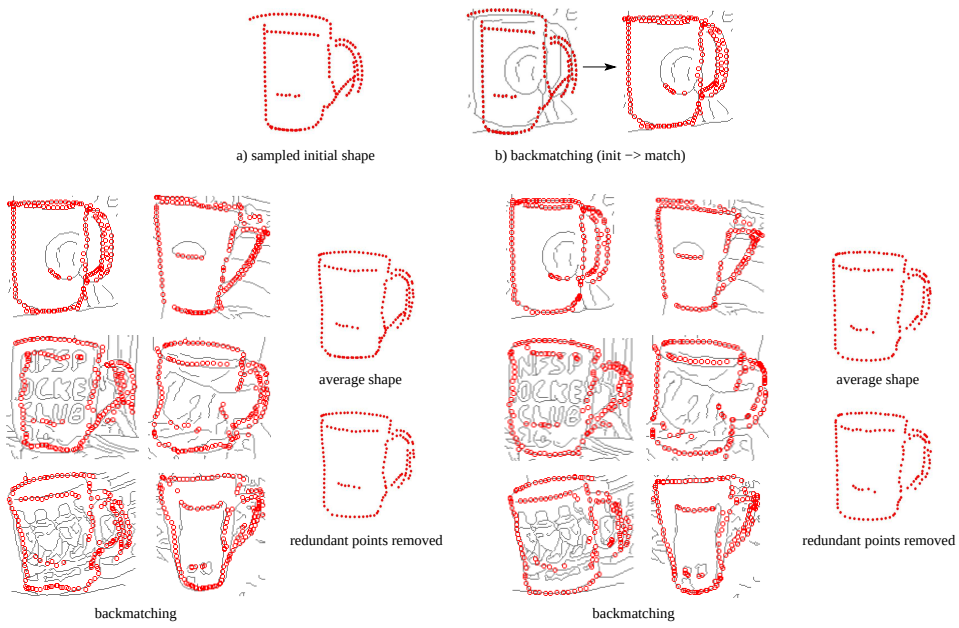


Figure 6: *Model shape refinement. a) sampled points from the initial model shape. b) after initializing backmatching by aligning the model with the image bounding-box (left), it deforms it so as to match the image edgels (right). c) the first iteration of shape refinement. d) the second iteration.*

4.3 Model shape refinement

In this subsection we refine the initial model shape constructed above. This is achieved by treating it as a deformable point set and applying a non-rigid point matching algorithm to *match it back* onto the training images. For each image, we *deform* the model by establishing point correspondences with the edgels within the object's bounding-box (figure 6b). An improved model shape is obtained by averaging these backmatched shapes. The updated shape is then matched back on the training images, and the process iterates. This results in a succession of better models and better matches to the training data, as the point correspondence cover more of the true class boundaries of the training objects (figure 6c+d).

The proposed technique can be seen as searching for the model shape that best explains the training data, under the rather general assumption that smooth deformations account for the difference between the model shape and the class boundaries of the training objects.

Before explaining the algorithm in detail, notice that we work now at the level of individual image edgels, no longer with PAS features as in the previous stages. As a preprocessing step, we sample the initial model shape, thus obtaining a point set (figure 6a). The main algorithm iterates over the following three steps:

1. Backmatching. The current model shape is matched back to each training image in turn. We use an extension of the excellent non-rigid robust point matcher by Chui and Rangarajan [5]. Following their naming, we refer to the algorithm by *TPS-RPM* (*Thin-Plate Spline Robust Point Matcher*).

TPS-RPM estimates a Thin-Plate Spline transformation between the model points and the image edgels while at the same time rejecting edgels not corresponding to any model point. This is an important ability, as only some of the image edgels lie on the object boundaries.

As with any shape matching technique, in order to work with noisy point sets TPS-RPM needs to start from a reasonable initial transformation. We initialize it by transforming the model shape so that its bounding-box aligns with the training bounding-box (figure 6b). This strong initialization makes TPS-RPM very likely to succeed. Subsection 5.2 presents this technique in more detail, as it will be used again for localizing object boundaries in test images.

2. Average shape. An important benefit of backmatching is that it generates shapes in *full point-to-point correspondence*, because they are all smooth deformations of the same initial shape (figure 6c). These correspondences allow us to manipulate the shapes as a set, and to analyze the variations in the point locations. We apply Cootes' variant of Procustes analysis [7] to align them with respect to translation, scale, and orientation. The remaining differences are due to true (non-rigid) shape variations, which we will learn in the next subsection.

The model shape is now updated by setting it to the average of the aligned shapes (i.e. average coordinates of corresponding points; figure 6c). The combined effects of backmatching and computing the average shape are very beneficial (figure 6c). Segments of the model shape are moved, bent, and stretched so as to form smooth, connected lines, thus recovering the shape of the class well on a *global* scale (e.g. topmost and leftmost segments in figure 6c). The reason for these improvements is that backmatching deforms the initial shape onto the class boundaries of the training images, thus delivering a set of natural, well formed shapes. The averaging step then integrates them all into a generic-looking shape, and smoothes out occasional inaccuracies of the individual backmatches.

3. Remove redundant points. As another effect, the previous steps tend to crush multiple strokes (and other clutter points) onto the same points along the correct class boundaries. This results in redundant points, roughly coincident with other segments (figure 6c). We remove them by deleting a point if it lies very close to another point from a part with higher confidence (see subsection 4.1 for definition of the confidence of a part). If a significant proportion of points ($> 10\%$) are removed, the procedure iterates to point 1 (figure 6d). Otherwise, it is completed.

Impact. As shown in figure 6d, the running example improves further during the second (and final) iteration. For example, the handle arcs becomes more continuous. The final shape

Figure 7: *Evolution of shape models over the three stages of learning. Top row: model parts (section 4.1). Second row: initial shape (section 4.2). Bottom row: refined shape (section 4.3).*

is overall smooth and well connected, includes no background clutter and very little interior clutter, and, as desired, represents an average class member (a *prototype shape*). Notice how both large scale (the external frame) *and* fine scale structures (the double handle arc) are correctly recovered. The backmatched shapes also improve in the second iteration, because shape matching is easier given a better model. In turn, the better backmatches yield a better average shape (i.e. a better model). The mutual help between backmatching and updating the model is key for the success of the procedure.

In figure 7, we show examples of other models evolving over the three stages (sections 4.1 to 4.3). Notice the large positive impact of model shape refinement. Furthermore, to demonstrate that the proposed techniques consistently produce good quality models, we show many of them in the result section (figure 11).

Discussion. Our idea for shape refinement is related to a general design principle which recently emerged in different areas of vision. It involves going *back to the image* after building some intermediate representation from initial low-level features, to refine and extend it. This differs from the conventional way of building layers of increasing abstraction, involving representations of higher and higher level, progressively departing from the original image data. The traditional strategy suffers from two problems: errors accumulate from a layer to the next, and relevant information missed by the low-level features is never recovered. Going back to the image enables to correct both problems, and it has good chances to succeed since a rough model has already been built. Several different algorithms are instances of this strategy and have led to excellent results in various areas: human pose estimation [31], appearance-based top-down segmentation [24], and recognition of specific objects [11].

4.4 Learning shape deformations

The previous subsection matches the model shape to each training image, and thus provides examples of the variations within the object class we want to learn. Since these examples are in full point-to-point correspondence, we can learn a compact model of the intra-class variations using the statistical shape analysis technique by Cootes [7].

The idea is to consider each example shape as a point in a $2p$-D space (with p the number of points on each shape), and model their distribution with Principal Component Analysis (PCA). The eigenvectors returned by PCA represent modes of variation, and the associated eigenvalues λ_i their importance (how much the example shapes deform along them, figure 3e). By keeping only the n largest eigenvectors $E_{1:n}$ representing 95% of the total variance, we can approximate the region in which the training examples live by $\bar{x} + E_{1:n}b$, where $\bar{x}$ is the mean shape, b is a vector representing shapes in the subspace spanned by $E_{1:n}$, and b's i^{th} component is bound by $\pm 3\sqrt{\lambda_i}$. This defines the *valid region* of the shape space, containing shapes similar to the example ones. Typically, $n < 15$ eigenvectors are sufficient (compared to $2p \simeq 200$).

Figure 3e shows the first two deformation modes for our running example. The first mode spans the spectrum between little coffee cups and tall Starbucks-style mugs, while the handle can vary from pointed down to pointed up within the second mode. In subsection 5.3, we exploit this deformation model to constrain the matching of the model to novel test images. Notice that previous works on these deformation models require at least the example shapes as input [17], and many also need the point-to-point correspondences [7]. In contrast, we automatically learn shapes, correspondences, and deformations given just *images*.

5 Object detection

In this section we describe how to localize the boundaries of previously unseen object instances in a test image. To this end, we match the shape model learnt in the previous section to the test image edges. This task is very challenging, because 1) the image can be extensively cluttered, with the object covering only a small proportion of its edges (figure 8a-b); and 2) to handle intra-class variability, the shape model must be *deformed* into the shape of the particular instance shown in the test image.

We decompose the problem into two stages. We first obtain rough estimates for location and scale of the object based on a Hough-style voting scheme (subsection 5.1). This greatly simplifies the subsequent shape matching, as it approximately lifts three degrees of freedom (translation and scale). The estimates are then used to initialize the non-rigid shape matcher [5] (subsection 5.2). This combination enables shape matching in cluttered images, and hence localization of object boundaries. Furthermore, in subsection 5.3, we constrain the matcher to explore only the region of shape space spanned by the training examples, thereby ensuring that output shapes are similar to class members.

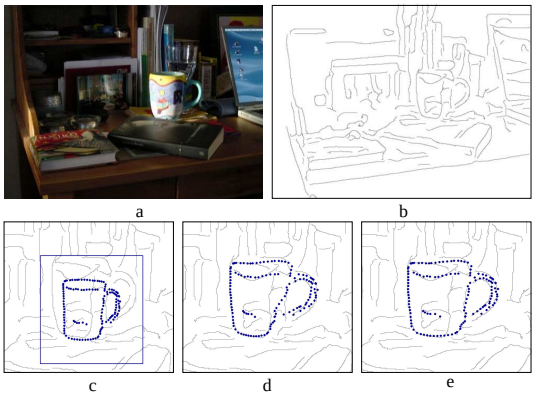


Figure 8: *a) A challenging test image and its edgemap b). The object covers only about 6% of the image surface, and only about 1 edgel in 17 belongs to its boundaries. c) Initialization with a local maximum in Hough space. d) Output shape with unconstrained TPS-RPM. It recovers the object boundaries well, but on the bottom-right corner, where it is attracted by the strong-gradient edgels caused by the shading inside the mug. e) Output of the shape-constrained TPS-RPM. The bottom-right corner is now properly recovered.*

5.1 Initialization by Hough voting

In subsection 4.1 we have represented the shape of a class as a set of PAS parts, each with a specific shape, location, size, and confidence. Here we match these parts to PAS from the test image, based on their shape descriptors. More precisely, a model part is deemed matched to an image PAS if their dissimilarity (1) is below a threshold t (this is the same as used in section 4.1). Since a pair of matched PAS induces a translation and scale transformation, each match votes for the presence of an object instance at a particular location (object center) and scale (in the same spirit as [24, 29, 33]). Votes are weighed by the shape similarity between the model part and test PAS, the edge strength of the PAS, and the confidence of the part. Local maxima in the voting space define rough estimates of the location and scale of candidate object instances (figure 8c).

The above voting procedure delivers 10 to 40 local maxima in a typical cluttered image, as the local features are not very distinctive on their own. The important point is that a few tens is *far less* than the number of possible location and scales the object could take in the image, which is in the order of the thousands. Thus, Hough voting acts as a focus of attention mechanism, drastically reducing the problem complexity. We can now afford to run a full-featured shape matching algorithm, starting from each of the initializations. Note that running shape matching directly, without initialization, is likely to fail on very cluttered images, where only a small minority of edgels are on the boundaries of the target object.

5.2 Shape Matching by TPS-RPM

For each initial location l and scale s found by Hough voting, we obtain a point set V by centering the model shape on l and rescaling it to s, and a set X which contains all image edge points within a larger rectangle of scale $1.8s$ (figure 8c). This larger rectangle is designed to contain the whole object, even when s is under-estimated. Any point outside this rectangle is ignored by the shape matcher.

Given the initialization, we want to put V in correspondence with the subset of X lying on the object boundary. We estimate the associated non-rigid transformation, and reject image points not corresponding to any model point with the Thin-Plate Spline Robust Point Matching algorithm (TPS-RPM [5]). In this subsection we give a brief summary of TPS-RPM, and we refer to [5] for details.

TPS-RPM matches the two point sets $V = \{v_a\}_{a=1..K}$ and $X = \{x_i\}_{i=1..N}$ by applying a non-rigid TPS mapping $\{d, w\}$ to V (d is the affine component, and w the non-rigid warp). It estimates both the correspondence matrix $M = \{m_{ai}\}$ between V and X, and the mapping $\{d, w\}$ that minimize an objective function including 1) the distance between points of X and their corresponding points of V after mapping them by the TPS, and 2) the regularization terms for the affine and warp components of the TPS. In addition to the inner $K \times N$ part, M has an extra row and an extra column which allow to reject points as unmatched.

Since neither the correspondence M nor the TPS mapping $\{d, w\}$ are known beforehand, TPS-RPM iteratively alternates between updating M, while keeping $\{d, w\}$ fixed, and updating the mapping with M fixed. M is a continuous-valued soft-assign matrix, allowing the algorithm to evolve through a continuous correspondence space, rather than jumping around in the space of binary matrices (hard correspondence). It is updated by setting m_{ai} as a function of the distance between x_i and v_a, after mapping by the TPS (details below). The update of the mapping fits a TPS between V and the current estimate $Y = \{y_a\}_{a=1..K}$ of the corresponding points. Each point y_a in y is a linear combination of all image points $\{x_i\}_{i=1..N}$ weighted by the soft-assign values $M(a, i)$:

$$y_a = \sum_{i=1}^{N} m_{ai} x_i \tag{6}$$

The TPS fitting maximizes the proximity between the points Y and the model points V after TPS mapping, under the influence of the regularization terms, which penalize local warpings w and deviations of d from the identity. Fitting the TPS to $V \leftrightarrow Y$ rather than to $V \leftrightarrow X$, allows to harvest the benefits of mantaining a full soft-correspondence matrix M.

The optimization procedure of TPS-RPM is embedded in a deterministic annealing framework by introducting a temperature parameter T, which decreases at each iteration. The entries of M are updated by the following equation:

$$m_{ai} = \frac{1}{T} \exp\left(\frac{(x_i - f(v_a, d, w))^T (x_i - f(v_a, d, w))}{2T} \right) \tag{7}$$

where $f(v_a, d, w)$ is the mapping of point v_a by the TPS $\{d, w\}$. The entries of M are then iteratively normalized to ensure the rows and columns sum to 1 [5]. Since T is the bandwidth

of the Gaussian kernel in equation (7), as it decreases M becomes less fuzzy, progressively approaching a hard correspondence matrix. At the same time, the regularization terms of the TPS is given less weight. Hence, the TPS is rigid in the beginning, and gets more and more deformable as the iterations continue. These two phenomena enable TPS-RPM to find a good solution even when given a rather poor initialization. At first, when the correspondence uncertainty is high, each y_a essentially averages over a wide area of X around the TPS-mapped point and the TPS is constrained to near-rigid transformations. This can be seen as a large T in equation (7) generates similar-valued m_{ai}, which are then averaged by equation (6). As the iterations continue and the temperature decreases, M looks less and less far, and pays increasing attention to the differences between matching options from X. Since the uncertainty diminishes, it is safe to let the TPS looser, freer to fit the details of X more accurately. Figure 9 illustrates TPS-RPM on our running example.

We have extended TPS-RPM by adding two terms to the objective function: the orientation difference between corresponding points (minimize), and the edge strength of matched image points (maximize). In our experiments, these extra terms made TPS-RPM more accurate and stable, i.e. it succeeds even when initialized farther away from the best location and scale.

5.3 Constrained shape matching

TPS-RPM treats all shapes according to the same *generic* TPS deformation model, simply preferring smoother transformations (in particular, low 2D curvature w, and low affine skew d). Two shapes with the same deformation energy are considered equivalent. This might result in output shapes unlike any of the training examples. In this section, we extend TPS-RPM with the *class-specific* deformation model learned in subsection 4.4. We constrain the optimization to explore only the valid region of the shape space, containing shapes plausible for the class (defined by $\bar{x}, E_{1:n}, \lambda_i$ from subsection 4.4).

At each iteration of TPS-RPM we project the current shape estimate Y (equation (6)) inside the valid region, just before fitting the TPS. This amounts to:

1) align Y on $\bar{x}$ w.r.t. to translation/rotation/scale
2) project Y on the subspace spanned by $E_{1:n}$:
   $b = E^{-1} \cdot (Y - \bar{x})$, $b_{(n+1):2p} = 0$
3) bound the first n components of b by $\pm 3\sqrt{\lambda_i}$
4) transform b back into the original space: $Y^c = \bar{x} + E \cdot b$
5) apply to Y^c the inverse of the transformation used in 1)

The assignment $Y \leftarrow Y^c$ imposes *hard constraints* on the shape space. While this guarantees output shapes similar to class members, it might sometimes be *too* restrictive. To match a novel instance accurately, it could be necessary to move a little along some dimensions of the shape space not recorded in the deformation model. The training data cannot be assumed to present all possible intra-class variations.

To tackle this issue, we propose a *soft-constrained* variant, where Y is *attracted* by the valid region, with a force that diminishes with temperature: $Y \leftarrow Y + \frac{T}{T_{init}}(Y^c - Y)$. This

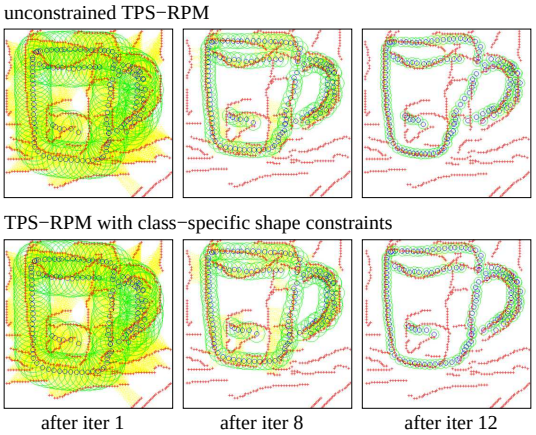


Figure 9: *Three iterations of TPS-RPM initialized as in figure 8c. The image points X are shown in red, and the current shape estimate Y in blue. The green circles have radius proportional to the temperature T, and give an indication of the range of potential correspondence considered by M. This is fully shown by the yellow lines joining all pairs of points with non-zero m_{ai}. Top: unconstrained TPS-RPM. Bottom: TPS-RPM with the proposed class-specific shape constraints. The two processes are virtually identical until iteration eight, when the unconstrained matcher diverges towards interior clutter. The constrained version instead, sticks to the true object boundary.*

causes TPS-RPM to start fully constrained, and then, as temperature decreases and M looks for correspondences closer to the current estimates, later iterations are allowed to apply small deformations beyond the valid region (typically along dimensions not in $E_{1:n}$). As a result, output shapes fit the image data more accurately, while still resembling class members. Notice how this behavior is fully in the spirit of TPS-RPM, which also lets the TPS more and more free as T decreases.

The proposed extension to TPS-RPM reaches deep into its heart as it *alters the search* through the transformation and correspondence spaces. Beside improving accuracy, it can help TPS-RPM to avoid local minima far from the correct solution, thus avoiding gross failures.

Figure 8e shows the improvement brought by the proposed constrained shape matching, compared to TPS-RPM with just the generic TPS model (figure 8d). On the running example, the two versions of TPS-RPM diverge crucially after the eight iteration, as shown in figure 9.

5.4 Detections

Every local maximum in Hough space constitutes an initialization for the shape matching, and results in different shapes (detections) localized in the test image. In this section we *score* the detections, making it possible to reject detections and to evaluate the detection rate and false-positive rate of our system.

	apple-logo	bottle	giraffe	mug	swan	INRIA
Training	20	24	44	24	16	50
Test	20+215	24+207	43+167	24+207	16+223	120+170

Table 1: *Number of training images and of positive+negative test images for all datasets.*

We score each detection by a weighted sum of four terms:

1) the number of matched model points, i.e. for which a corresponding image point has been found with good confidence. Following [5], these are all points v_a with $\max_{i=1..N}(m_{ai}) > 1/N$.
2) the sum of squared distances from the TPS-mapped model points to their corresponding image points. This measure is made scale-invariant by normalizing by the squared range r^2 of the image point coordinates (width or height, whichever is larger). Only matched model points are considered.
3) the deviation $\sum_{i,j\in[1,2]} \left(I(i,j) - d(i,j)/\sqrt{|d|}\right)^2$ of the affine component d of the TPS from the identity I. The normalization by the determinant of d factors out deviations due to scale changes.
4) the amount of non-rigid warp w of the TPS $\text{trace}(w^T \Phi w)/r^2$,
where $\Phi(a,b) \propto ||v_a - v_b||^2 \log ||v_a - v_b||$ is the TPS kernel matrix [5].

This score integrates the information a matched shape provides. It is high when the TPS fits *many* (term 1) points *well* (term 2), without having to *distort* much (terms 3 and 4). In our current implementation, the relative weights between these terms have been selected manually, they are the same for all classes, and remain fixed in all experiments.

As a final refinement, if two detections overlap substantially, we remove the lower scored one. Notice that the method can detect multiple instances of the same class in an image. Since they appear as different peaks in the Hough voting space, they result in separate detections.

6 Experiments

We present an extensive experimental evaluation involving six diverse object classes from two existing datasets [12, 22]. After introducing the datasets in the next subsection, we evaluate our approach for learning shape models in subsection 6.2. The ability to localize objects in novel test images, both in terms of traditional bounding-boxes and precise object boundaries, is measured in subsection 6.3. All experiments are run with exactly the same parameters (no class-specific nor dataset-specific tuning is applied).

	apple-logo	bottle	giraffe	mug	swan
Full system	90.2 / 90.6	96.2 / 92.7	70.8 / 74.3	93.9 / 83.6	90.0 / 80.0
No assembly	91.2 / 92.7	96.8 / 88.1	70.0 / 72.6	92.6 / 82.9	89.4 / 79.2

Table 2: *Accuracy of learned models. Each entry is the average coverage/precision over trials and training instances.*

6.1 Datasets and protocol

ETHZ shape classes [12]. This dataset features five diverse classes (bottles, swans, mugs, giraffes, apple-logos) and contains a total of 255 images collected from the web by Ferrari et al. [12]. It is highly challenging, as the objects appear in a wide range of scales, there is considerable intra-class shape variation, and many images are severely cluttered, with objects comprising only a fraction of the total image area (figures 16, 13).

For each class, we learn 5 models, each from a different random sample containing half of the available images (there are 40 for apple-logos, 48 for bottles, 87 for giraffes, 48 for mugs and 32 for swans). Learning models from different training sets allows to evaluate the stability of the proposed learning technique (subsection 6.2). Notice that our method does not require any negative image for training, i.e. images not containing any instance of the class.

The test set for a model consists of *all* other images in the dataset. Since this includes about 200 negative images, it allows to properly estimate false-positive rates. Table 6 gives an overview of the composition of all training and testing sets. We refer to learning and testing on a particular split of the images as a *trial*.

INRIA horses [22]. This challenging dataset consists of 170 images with one or more horses viewed from the side and 170 images without horses. Horses appear at several scales, and against cluttered backgrounds.

We train 5 models, each from a different random subset of 50 horse images. For each model, the remaining 120 horse images and all 170 negative images are used for testing, see table 6.

6.2 Learning shape models

Evaluation measures. We assess the performance of the learning procedure of section 4 in terms of how accurately it recovers the true class boundaries of the training instances. For the purpose of this evaluation, we have manually annotated the boundaries of all object instances in the ETHZ shape classes dataset.

Let B_{gt} be the ground-truth boundaries, and B_{model} the backmatched shapes output by the model shape refinement algorithm of subsection 4.3. The accuracy of learning is quantified by two measures. *Coverage* is the percentage of points from B_{gt} closer than a threshold t from any point of B_{model}. We set t to 4% of the diagonal of the bounding-box of B_{gt}. Conversely, *precision* is the percentage of B_{model} points closer than t from any point of

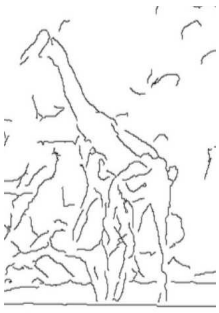


Figure 10: *A typical edgemap for a Giraffe training window is very cluttered and edges are broken along the animal's outline, making it difficult to learn clean models.*

B_{gt}. The measures are complementary. Coverage captures how much of the object boundary has been recovered by the algorithm, whereas precision reports how much of the algorithm's output lies on the object boundaries.

Models from the full algorithm. Table 2 shows coverage and precision averaged over training instances and trials, for the complete learning procedure described in section 4. With the exception of giraffes, the proposed method achieves very high coverage (above 90%), demonstrating its ability to discover which contour points belong to the class boundaries. The precision of apple-logos and bottles is also excellent, thanks to the remarkably clean prototype shapes learned by our approach (figure 11). Interestingly, the precision of mugs is somewhat lower, because the learned shapes include a detail not present in the ground-truth annotations, although it is arguably part of the class boundaries: the inner half of the opening on top of the mug. A similar phenomenon penalizes the precision of swans, where our method sometimes includes a few water waves in the model. Although they are not part of the swan boundaries, waves accidentally occurring at a similar position over many training images are picked up by the algorithm. A larger training set might lead to the suppression of such artifacts, as waves have less chances of accumulating accidentally (we only used 16 images). The modeling performance for giraffes is lower, due to the extremely cluttered edgemaps arising from their natural environment, and to the camouflage texture which tends to break edges along the body outlines (figure10).

Models without assembling the initial shape. One could design a simpler scheme for learning shape models by skipping the procedure for assembling the initial shape (section 4.2). An alternative initial shape could be obtained directly from the COP model (section 4.1) by picking the highest confidence occurrence for each part independently (i.e. the single occurrence maximizing equation (2)). This initial shape could then be passed on to the shape refinement stage as usual (section 4.3).

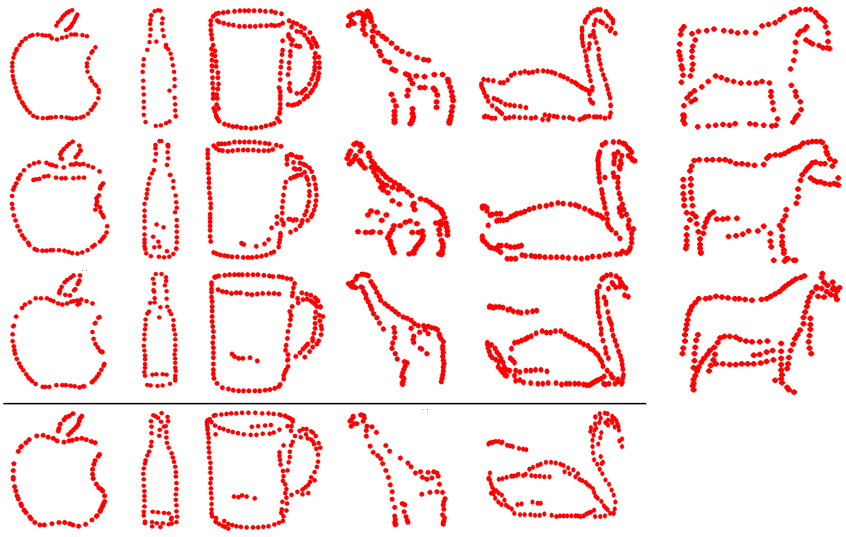


Figure 11: *Learned shape models (three out of total five per class).* **Top three rows:** *models learnt using the full method presented in section 4. The first 5 columns show models learned from the ETHZ shape classes dataset, while the last column models from the INRIA horses dataset.* **Last row:** *models learnt using the same training images used in row 3, but skipping the procedure for assembling the initial shape (subsection 4.2; done only for ETHZ shape classes).*

For each object class and trial we have rerun the learning algorithm without the assembly stage, but otherwise keeping identical conditions (including using exactly the same training images). Many of the resulting prototype shapes are moderately worse than those obtained using the full learning scheme (figure 11 bottom row). However, the lower model quality only results in slightly lower average coverage/accuracy values (table 2). These results suggest that while the initial assembly stage does help getting better models, it is not a crucial step. Interestingly, this suggests that the shape refinement stage proposed in section 4.3 is robust to large amounts of noise, and delivers good models even when starting from poor initial shapes.

6.3 Object detection

Detection up to a bounding-box. We first evaluate the ability of the object detection procedure of section 5 to localize objects in cluttered test images up to a bounding-box (i.e. the traditional detection task commonly defined in the literature).

Figure 12 reports detection-rate against the number of false-positives averaged over *all* test images (FPPI) and averaged over the 5 trials. As discussed above, this includes mostly negative images. We adopt the strict standards of the PASCAL Challenge criterion (dashed lines in the plots): a detection is counted as correct only if the intersection-over-union ratio

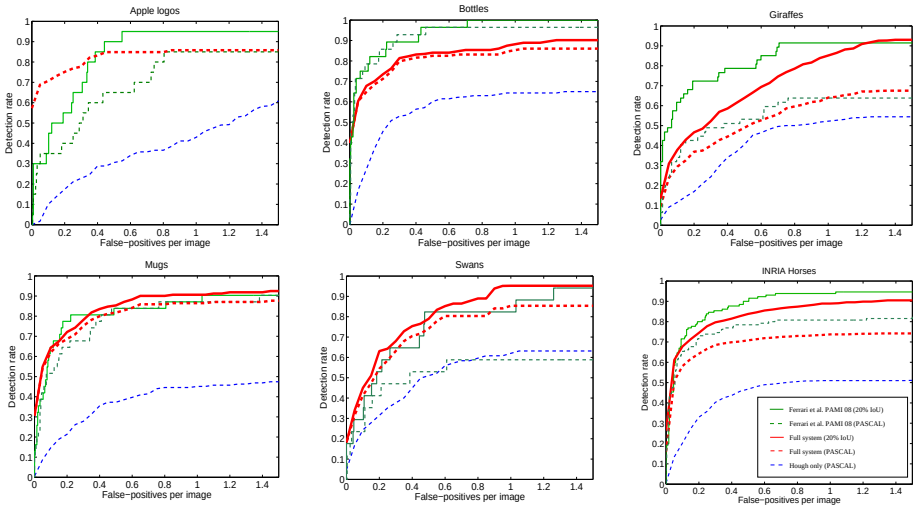


Figure 12: *Object detection performance (models learnt from real images). Each plot shows five curves: the full system evaluated under the PASCAL criterion for a correct detection (dashed, thick, red), the full system under the 20%-IoU criterion (solid, thick, red), the Hough voting stage alone under PASCAL (dashed, thin, blue), [14] under 20%-IoU (solid, thin, green) and under PASCAL (dashed, thin, green). The curve for the full system under PASCAL in the apple-logo plot is identical to the curve for 20%-IoU.*

(IoU) with the ground-truth bounding-box is greater than 50%. All other detections are counted as false-positives. In order to compare to [12, 14], we also report results under their somewhat softer criterion: a detection is counted as correct if its bounding-box overlaps more than 20% with the ground-truth one, and vice-versa (we refer to this criterion as *20%-IoU*).

As the plots show, our method performs well on all classes but giraffes, with detections-rates around 80% at the moderate false-positive rate of 0.4 FPPI (this is the reference point for all comparisons). The lower performance on giraffes is mainly due to the difficulty of building shape models from their extremely noisy edge maps.

It is interesting to compare against the detection performance obtained by the Hough voting stage alone (subsection 5.1), without the shape matcher on top (subsections 5.2, 5.3). The full system performs substantially better: the difference under PASCAL criterion is about 30% averaged over all classes. This shows the benefit of treating object detection fully as a shape matching task, rather than simply matching local features, which is one of the principal points of this paper. Moreover, the shape matching stage also makes it possible to localize complete object boundaries, rather than just bounding-boxes (figure 13).

The difference between the curves under the PASCAL criterion and the 20%-IoU criterion of [12, 14] is small for apple-logos, bottles, mugs and swans (0%, 1.6%, 3.6%, 4.9%), indicating that most detections have accurate bounding-boxes. For horses and giraffes the decrease is more significant (18.1%, 14.1%), because the legs of the animals are harder to detect and

	apple-logo	bottle	giraffe	mug	swan
Full system	91.6 / 93.9	83.6 / 84.5	68.5 / 77.3	84.4 / 77.6	77.7 / 77.2
No learned deform	91.3 / 93.6	82.7 / 84.2	68.4 / 77.7	83.2 / 75.7	78.4 / 77.0
Ground-truth BB	42.5 / 40.8	71.2 / 67.7	26.7 / 29.8	55.1 / 62.3	36.8 / 39.3

Table 3: *Accuracy of localized object boundaries at test time. Each entry is the average coverage/precision over trials and correct detections at* 0.4 *FPPI.*

cause the bounding-box to shift along the body. On average over all classes, our method achieves 78.1% detection-rate at 0.4 FPPI under 20%-IoU and 71.1% under PASCAL.

For reference, the plots also show the performance of [14] on the same dataset, using the same number of training and test images. An exact comparison is not possible, as [14] reports result based on only one training/testing split, whereas we average over 5 random splits. Under the rather permissive 20%-IoU criterion, [14] performs a little better than our method on average over all classes. Under the strict PASCAL criterion instead, our method performs substantially better than [14] on two classes (apple-logos, swans), moderately worse on two (bottles, horses), and about the same on two (mugs, giraffes), thanks to the higher accuracy of the detected bounding-boxes. Averaged over all classes, under PASCAL our method reaches 71.1% detection-rate at 0.4 FPPI, comparing well against the 68.5% of [14]. Notice how our results are achieved even without the beneficial discriminative learning of [14], where a SVM learns which PAS types at which relative location within the training bounding-box best discriminate between instances of the class and background image windows. Our method instead trains only from positive examples. Beyond this evaluation, the method presented in this paper offers two important advantages over [14]. It localizes object boundaries, rather than just bounding-boxes, and can also detect objects starting from a single hand-drawing as a model (see below).

Localizing object boundaries. The most interesting feature of our approach is the ability to localize object boundaries in novel test images. This is shown by several examples in figure 13, where the method succeeds in spite of extensive clutter, a large range of scales, and intra-class variability. In the following we quantify how accurately the output shapes match the true object boundaries. We use the coverage and precision measures defined above. In the present context, coverage is the percentage of ground-truth boundary points recovered by the method and precision is the percentage of output points that lie on the ground-truth boundaries.

Table 3 shows coverage and precision averaged over trials and correct detections at 0.4 FPPI. Coverage ranges in $78 - 92\%$ for all classes but giraffes, demonstrating that most of the true boundaries have been successfully detected. Moreover, precision values are similar, indicating that the method returns only a small proportion of points outside the true boundaries. Performance is lower for giraffes, due to the more noisy models and difficult edgemaps derived from the test images.

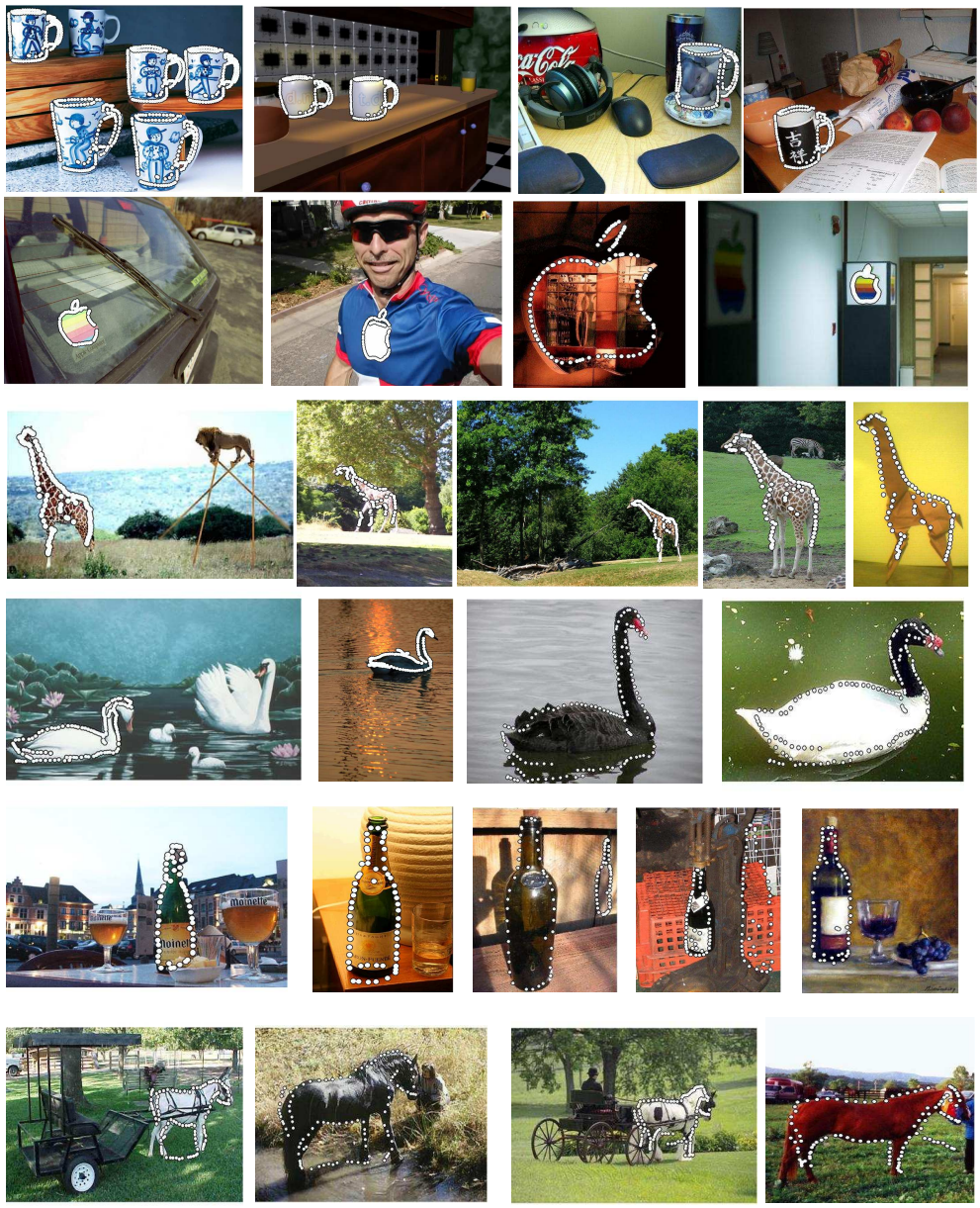
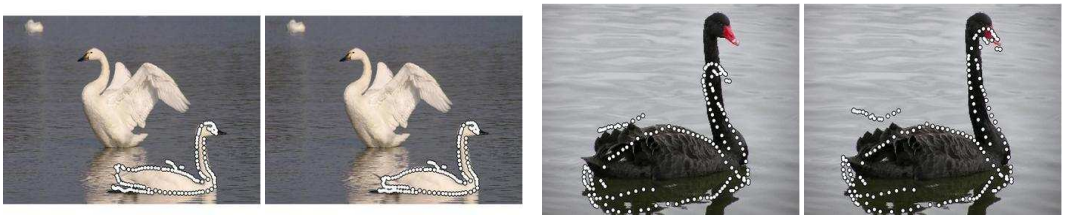


Figure 13: *Example detections (models learnt from images). Notice the large scale variations (especially in apple-logos, swans), the intra-category shape variability (especially in swans, giraffes), and the extensive clutter (especially in giraffes, mugs). The method works for photographs as well as paintings (first swan, last bottle). Two bottle cases show also false-positives. In the first two horse images, the horizontal line below the horses' legs is part of the model and represents the ground. Interestingly, the ground line systematically reoccurs over the training images for that model and gets learned along with the horse.*



default TPS | shape constrained | default TPS | shape constrained

Figure 14: *(left) typical improvement brought by constrained shape matching over simply using the TPS deformation model. As the improvement is often a refinement of a local portion of the shape (the swan's tail in this case), the numerical differences in the evaluation measures is only modest (in this case less than* 1%*). (right) an infrequent case, where constrained shape matching fixes the entirely wrong solution delivered by standard matching. The numerical difference in such cases is noticeable (about 6%).*

Although it uses the same evaluation metric, the experiment carried out at training time in subsection 6.2 differs substantially from the present one, because at testing time the system is *not* given ground-truth bounding-boxes. In spite of the important additional challenge of having to determine the object's location and scale in the image, the coverage/precision scores in table 3 are only moderately lower than those achieved during training (table 3; the average difference in coverage and precision is 7.1% and 2.1% respectively). This demonstrates that our detection approach is highly robust to clutter.

As a baseline, table 3 also reports coverage/precision results when using the *ground-truth bounding-boxes as shapes*. The purpose of this experiment is to compare the accuracy of our method to the maximal accuracy that can be achieved when localizing objects up to a bounding-box. As the table clearly shows, the shapes returned by our method are substantially more accurate than the best bounding-box, thereby proving one of the principal points of this paper. While the average difference is about 35%, it is interesting to observe how the difference is greater for less rectangular objects (swans, giraffes, apple-logos) than for bottles and mugs. Notice also how our method is much more accurate than the ground-truth bounding-box even for giraffes, the class where it performs the worst.

Finally, we investigate the impact of the constrained shape matching technique proposed in subsection 5.3, by re-running the experiment without it, simply relying on the deformation model implicit in the thin-plate spline formulation (table 3, second row). The coverage/precision values are very similar to those obtained through constrained shape matching. The reason is that most cases are either already solved accurately without learned deformation models, or they do not improve when using them because the low accuracy is due to particularly bad edgemaps. In practice, the difference made by constrained shape matching is visible in about one case every six, and it is localized to a relatively small region of the shape (figure 14). The combination of these two factors explains why constrained shape

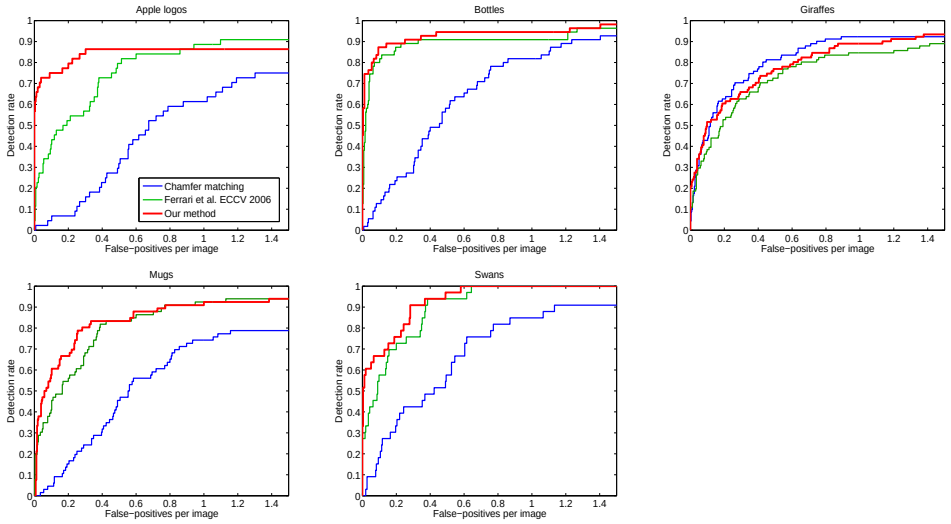


Figure 15: *Object detection performance (hand-drawn models). To facilitate comparison, all curves have been computed using the 20%-IoU criterion of [12].*

matching appears to make little quantitative difference, although in many cases the localized boundaries improve visibly.

Detection from hand-drawn models. A useful characteristic of the proposed system is that, unlike most existing object detection approaches, it can take a *hand-drawing* as a model, as well as training from real images. In the hand-drawing case, the modeling stage simply decomposes the hand-drawing into PAS. Object detection then uses these PAS for the Hough voting stage, and the hand-drawing itself for the shape matching stage. As no deformation model can be learnt from a single example, our method naturally switches to the standard deformation model implicit in the Thin-Plane Spline formulation.

Figure 15 compares our method to [12] using their exact setup, with a single hand-drawing per class as model and *all* 255 images of the ETHZ shape classes as test set. Our method performs better than [12] on all 5 classes, especially in the low FPPI range, and substantially outperforms the oriented chamfer matching baseline (details in [12]).

Beside this quantitative evaluation, our approach offers three additional advantages over [12]: it can train from real images, it supports branching and self-intersecting input shapes, and it does not need the test image to contain long chains of contour segments around the object (and is therefore more robust).

Interestingly, hand-drawings lead to a moderately better detection results than when learning models from images. This is less suprising when considering that hand-drawings are essentially the prototype shapes the system tries to learn.

Figure 16: *Example detections. Results are based on hand-drawings from [12] as models.*

7 Conclusions and future work

As confirmed by the experiments, we have proposed an approach capable of learning class-specific explicit shape models from images annotated by bounding-boxes, and then localize the boundaries of novel class instances in the presence of extensive clutter, scale changes, and intra-class variability. In addition, the approach operates effectively also when given hand-drawings as models. Interestingly, the ability to input both images and hand-drawings as training data is a consequence of the fundamental design of our approach, which is meant to bridge the gap between shape matching and object detection.

The presented approach can be extended in several ways. First, the training stage models only positive examples. This could be extended by learning a classifier to distinguish positive and negative examples, which might reduce false-positives. One possibility could be to train both our shape models and the discriminative models of [14]. At detection time, we could then use the bounding-box delivered by [14] to initialize shape matching based on our models. Moreover, the current method exploits only image contours, discarding valuable information such as texture and color. To further improve discriminative power, the representation could be augmented with appearance features. Finally, our approach assumes that all training objects for a class are seen from approximately the same viewpoint. It would be interesting to add a clustering stage to automatically group objects by viewpoint, and learn separate shape models.

References

[1] R. Basri, L. Costa, D. Geiger, and D. Jacobs, *Determining the Similarity of Deformable Shapes*, Vision Research, vol. 38, pp. 2365-2385, 1998.

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

[3] S. Belongie and J. Malik, *Shape Matching and Object Recognition using Shape Contexts*, PAMI, 24(4):509-522, 2002.

[4] I. Biederman, *Recognition-by-components: A theory of human image understanding*, Psychological Review, 94(2):115-147.

[5] H. Chui and A. Rangarajan, *A new point matching algorithm for non-rigid registration*, CVIU, 2003.

[6] O. Chum and A. Zisserman, *An Exemplar Model for Learning Object Classes*, CVPR, 2007.

[7] T. Cootes, C. Taylor, D. Cooper, and J. Graham, *Active Shape Models: Their Training and Application*, CVIU, 61(1):38-59, 1995.

[8] D. Cremers, T. Kohlberger, and C. Schnor, *Nonlinear Shape Statistics in Mumford-Shah Based Segmentation* ECCV, 2002.

[9] N. Dalal and B. Triggs, *Histograms of Oriented Gradients for Human Detection*, CVPR, 2005.

[10] G. Elidan, G. Heitz, D. Koller, *Learning Object Shape: From Drawings to Images*, CVPR, 2006.

[11] V. Ferrari, T. Tuytelaars, and L. van Gool, *Simultaneous Object Recognition and Segmentation by Image Exploration*, ECCV, 2004.

[12] V. Ferrari, T. Tuytelaars, and L. Van Gool, *Object Detection with Contour Segment Networks*, ECCV, 2006.

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

[14] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, *Groups of Adjacent Contour Segments for Object Detection*, PAMI, (30)1:36-51, 2008.

[15] P. Felzenswalb, *Representation and Detection of Deformable Shapes*, PAMI, 27(2):208-220, 2005.

[16] R. Fergus, P. Perona, and A. Zisserman, *Object Class Recognition by Unsupervised Scale-invariant Learning*, CVPR, 2003.

[17] A. Hill and C. Taylor, *A Method of Non-Rigid Correspondence for Automatic Landmark Identification*, BMVC, 1996.

[18] D. Gavrila, *Multi-Feature Hierarchical Template Matching Using Distance Transforms*, ICPR, 1998.

[19] Y. Gdalyahu and D. Weinshall, *Flexible Syntactic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes*, PAMI, 21(12):1312-1328, 1999.

[20] S. Gold and A. Rangarajan, *Graduated Assignment Algorithm for Graph Matching*, PAMI, 18(4):377-388, 1996.

[21] J. Hummel, *Where view-based theories break down: The role of structure in shape perception and object recognition*, In E. Dietrich and A. Markman (Eds.) Cognitive Dynamics: Conceptual Change in Humans and Machines, pp. 157-185, Hillsdale, NJ: Erlbaum, 2000.

[22] F. Jurie and C. Schmid *Scale-invariant Shape Features for Recognition of Object Categories*, CVPR, 2004.

[23] Y. Lamdan, J. Schwartz, and H. Wolfson, *Affine invariant model-based object recognition*, IEEE Transactions on Robotics and Automation, 6(5):578-589, 1990.

[24] B. Leibe and B. Schiele, *Scale-Invariant Object Categorization using a Scale-Adaptive Mean-Shift Search*, DAGM, 2004.

[25] M. Leordeanu, M. Hebert, and R. Sukthankar, *Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features*, CVPR, 2007.

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

[27] D. Marr and H.K. Nishihara, *Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes*, Proc. Royal Soc. London, Series B, Biological Sciences, (200):269-294, 1978.

[28] F. Mokhtarian and A. Mackworth, *Scale-based description and recognition of planar curves and two-dimensional shapes*, PAMI, 8(1):34-43, 1986.

[29] A. Opelt, A. Pinz, and A. Zisserman, *A Boundary-Fragment-Model for Object Detection*, ECCV, 2006.

[30] A. Pentland, A.; Sclaroff, S.; *Closed-form solutions for physically based shape modeling and recognition*, PAMI, 13(7):715-729, 1991.

[31] D. Ramanan, *Learning to parse images of articulated bodies*, NIPS, 2006.

[32] T. Sebastian, P. Klein, B. Kimia, *Recognition of shapes by editing their shock graphs*, PAMI, 26(5):550-571, 2004.

[33] J. Shotton, A. Blake, and R. Cipolla, *Contour-Based Learning for Object Detection*, ICCV, 2005.

[34] A. Torralba, K. Murphy, and W. Freeman, *Sharing Features: efficient boosting procedures for multiclass object detection*, CVPR, 2004.

[35] D. Sharvit, J. Chan, H. Tek, B Kimia, *Symmetry-Based Indexing of Image Databases*, IEEE Workshop on Content-Based Access of Image and Video Libraries, 1998.

[36] S. Ullman, *Aligning pictoral descriptions: An approach to object recognition*, Cognition, 32(3):193-254, 1989.

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

[38] J. Winn and J. Shotton, *The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects*, CVPR, 2006.

[39] C. Zahn, and R. Roskies, *Fourier Descriptors for Plane Closed Curves*, IEEE Transactions on Computer, (21)3:269-281, 1972.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399