Multi-Scale Categorical Object Recognition Using Contour Fragments

Jamie Shotton, Andrew Blake, Roberto Cipolla

Abstract— **Psychophysical studies [9], [17] show that we can recognize objects using fragments of outline contour alone. This paper proposes a new automatic visual recognition system based only on local contour features, capable of localizing objects in space and scale. The system first builds a class-specific codebook of local fragments of contour using a novel formulation of chamfer matching. These local fragments allow recognition that is robust to within-class variation, pose changes, and articulation. Boosting combines these fragments into a cascaded sliding-window classifier, and mean shift is used to select strong responses as a final set of detections. We show how learning can be performed iteratively on both training and test sets to boot-strap an improved classifier. We compare with other methods based on contour and local descriptors in our detailed evaluation over 17 challenging categories, and obtain highly competitive results. The results confirm that contour is indeed a powerful cue for multi-scale and multi-class visual object recognition.**

Index Terms— **Edge and feature detection, feature representation, size and shape, object recognition, computer vision, machine learning.**

I. Introduction

Consider the images in Figure 1, and try to identify the objects present. The object identities are hopefully readily apparent. This simple demonstration confirms the intuition that fragments of contour can be used to successfully recognize objects in images, and detailed psychophysical studies such as [9], [17] bear this out. With this inspiration, we set out to build an automatic object recognition system that uses only the cue of contour. The most significant contribution of this work is the demonstration that such a system can accurately recognize objects from challenging and varied object categories at multiple scales.

Our system aims to learn, from a small set of training images, a class-specific model for classification and detection in unseen test images. The task of classification is to determine the presence or absence of objects of a particular class (category) within an image, answering the question "does this image contain at least one *X*?", while detection aims to localize any such objects in space and scale, answering "how many *X*s are in this image, and where are they?". Systems that can answer these questions are rapidly becoming central to applications such as image search, robotics, vehicle safety systems, and image editing, to name but a few.

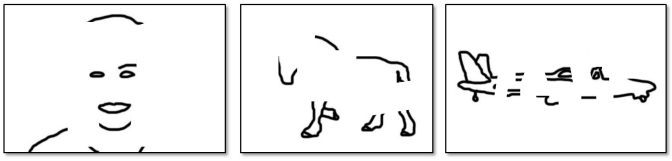

Fig. 1. **Object recognition using contour fragments.** Our innate biological vision system is able to interpret spatially arranged local fragments of contour to recognize the objects present. In this work we show that an automatic computer vision system can also successfully exploit the cue of contour for object recognition.

We define *contour* as the outline (silhouette) together with the internal edges of the object. Contour has several advantages over other cues: it is largely invariant to lighting conditions (even silhouetting) and variations in object color and texture, it can efficiently represent image structures with large spatial extents, and it varies smoothly with object pose change (up to genus changes). It can be matched accurately along the object boundary, where image patches and local descriptor vectors tend to match unreliably due to interaction with the varying background. Note that for the didactic purposes of this work we are deliberately ignoring other useful visual cues, such as color and texture.

The evident power of contour as a recognition cue is somewhat mitigated by practical realities. Contour must be matched against some form of edge map, but reliable edge detection and figure-ground segmentation are still areas of active research [11], [18], [38], [44]. Indeed the problems of edge detection, object detection, and segmentation are intimately bound together: an accurate segmentation mask is useful for recognition, while an object localization

gives an excellent initialization for bottom-up segmentation.

The most significant challenges, therefore, are noisy edge maps and background clutter. Whole object contours are fairly robust to this clutter, but have poor generalization qualities, and for deformable objects therefore require many exemplars that are often arranged hierarchically [28]. Improved models, where whole object templates are divided into *parts*, have recently become prominent in Computer Vision, e.g. [22], [23], [53]. Parts are individually less discriminative, but in ensemble prove robust to clutter and occlusion, and are able to generalize across both rigid and articulated object classes. In this paper, we present a system that learns parts based on *contour fragments* that in combination robustly match both the object outline and repeatable internal edges. Some existing systems, e.g. [23], are computationally limited to a small number of parts, but our technique efficiently copes with larger numbers, of the order of 100. The resulting over-complete model has built-in redundancy with tolerance to within-class variation, different imaging conditions such as lighting, occlusion, clutter, and small pose changes.

The spatial layout of parts is clearly informative, although the degree to which it is modeled varies enormously. The remarkably successful *bag-of-words* model [15], [47], [48] throws away all spatial information and exploits only the repeatable co-occurrence of features to recognize objects or scenes. Alternatively, for a small number of parts, a full joint spatial layout distribution can be learned [23]. Our approach will take a middle ground between these two extremes.

Our preliminary work [46] proved that automatic object recognition was indeed achievable using only contour information. This paper strengthens and extends that thesis with the following contributions: (i) a codebook of scale-normalized contour exemplars, learned automatically from the training images without requiring figure-ground segmentations, (ii) efficient recognition at multiple scales, (iii) a new multi-scale oriented chamfer distance for matching contour fragments, and (iv) a bootstrapping technique that augments the sparse set of training examples used to learn the classifier. The evaluation of classification and detection performance is extended to 17 categories. We introduce a new challenging multi-scale horse dataset, and compare performance with methods based on contours [25], [43] and local descriptors [55].

After the related work immediately below, we begin by defining our object model in Section II, and then our contour fragments in Section III. Section IV presents the object detector, and Section V describes the method for learning the parameters thereof. We present our evaluation in Section VI, and conclude the paper with Section VII.

A. Related Work

We focus this review on techniques that also use contour for recognition. Marr's Primal Sketch [37] already considered contour a powerful cue. Contour was first used for particular objects, matched as complete, rigid templates [30], but later for articulated objects e.g. people in [20], [29], [51], and hands in [49]. Leibe *et al.* [35] used chamfer matched pedestrian outlines in a verification stage. These techniques match whole contours and therefore depend on a large set of templates to represent all joint object configurations. The Generalized Hough Transform [5] is an alternative matching scheme to chamfer or Hausdorff matching. Carmichael and Hebert recognized wiry objects based on edges in [13].

Alternative approaches use fragments of contour. Nelson and Selinger's influential work [40] grouped contour fragments in a multi-level system for recognizing simple 3D objects. Fergus *et al.* [24] augmented the constellation model with contour fragment features, but only exploited fairly clean, planar curves with at least two points of inflection. In [31], contour fragments were arranged in Layered Pictorial Structures and used for detection of articulated objects; good results were obtained although tracked video sequences or manually labeled parts were required for learning. Borenstein & Ullman [10] used image and contour fragments for segmentation, though did not address recognition.

Other methods use local descriptors of contour. Rigid objects were addressed effectively in [39]. Shape contexts [7] describe sampled edge points in a log-polar histogram. The geometric blur descriptor was used in [8] to match deformable objects between pairs of images. More recently, Ferrari *et al.* [25] combined groups of adjacent segments of contour [26] into invariant descriptors, and sliding windows of localized histograms enabled object detection.

Most similar to our work is that of Opelt *et al.* [42], [43]. Their 'boundary fragment model' (BFM) shares much with our earlier work [46]: it uses many fragments of contour arranged in a star constellation learned by boosting and matched with a chamfer distance. Our new work incorporates its advantages of scale invariance, robust detection using mean shift, and reduced supervision (bounding boxes rather than segmentations), but there are important differences. We employ a new chamfer distance that treats orientation in a continuous manner, and show in Section VI-C.1 how this leads to improved recognition accuracy. Contour fragments are matched in local windows relative to the object centroid, rather than across the whole image. The BFM combines several fragments in each weak learner, while our fragments proved sufficiently discriminative individually, reducing training expense. Training from a sparse set of image locations (Figure 7) results in further efficiency. We model scale as an extra dimension in the mean shift mode detection, rather than combining object detections from individual scales post-hoc. Subsequent work [43] showed how to share contour fragments between classes, similar to [50]. We compare against these techniques in Section VI-C.9.

II. Object Model

As motivated in the introduction, we use a parts-based object model, shown in Figure 2. We employ a star constellation in which the parts are arranged spatially about a single fiducial point, the object *centroid*. Each training image contains a number of objects, each of which is labeled with a bounding box $b = (\mathbf{b}_{\text{tl}}, \mathbf{b}_{\text{br}})$ that implicitly defines this centroid $\mathbf{x} = \frac{1}{2}(\mathbf{b}_{\text{tl}} + \mathbf{b}_{\text{br}})$ and also the object *scale* $s = \sqrt{\text{area}(b)}$. The object model is defined at scale $s = 1$, and parts derived from objects in images are *scale-normalized* to this canonical scale. Each scale-normalized part $F = (\bar{T}, \bar{\mathbf{x}}_{\text{f}}, \sigma)$ is a contour fragment $\bar{T}$ with expected offset $\bar{\mathbf{x}}_{\text{f}}$ from the centroid, and spatial uncertainty σ.

III. Contour Fragments

This section defines our novel formulation of chamfer matching, before showing how a class-specific codebook of contour fragments is learned.

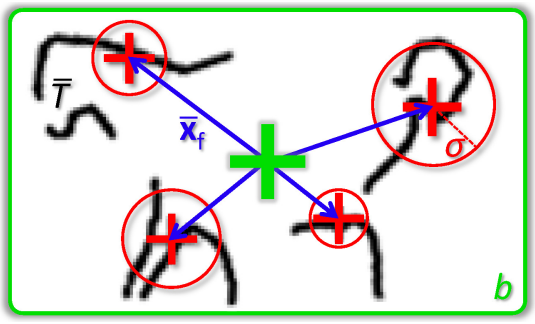

Fig. 2. **Object model.** Contour fragments $\bar{T}$ (black outlines) are arranged about the object centroid (green cross) within the bounding box b (green). Blue arrows show the expected offsets $\bar{\mathbf{x}}_{\text{f}}$ from the centroid, and red circles the spatial uncertainty σ. For clarity, only four parts are drawn; in practice, about 100 parts are used.

A. Chamfer Matching

The chamfer distance function, originally proposed in [6], measures the similarity of two contours. It is a smooth measure with considerable tolerance to noise and misalignment in position, scale and rotation, and hence very suitable for matching our locally rigid contour fragments to noisy edge maps. It has already proven capable of and efficient at recognizing whole object outlines (e.g. [28], [35], [49]), and here we extend it for use in a multi-scale parts-based categorical recognition model.

In its most basic form, chamfer distance takes two sets of edgels (edge points), a template T and an edge map E, and evaluates the asymmetric distance for 2D relative translation $\mathbf{x}$ as:

$$d_{\text{cham}}^{(T,E)}(\mathbf{x}) = \frac{1}{|T|} \sum_{\mathbf{x}_{\text{t}} \in T} \min_{\mathbf{x}_{\text{e}} \in E} \|(\mathbf{x}_{\text{t}} + \mathbf{x}) - \mathbf{x}_{\text{e}}\|_2 \ , \quad (1)$$

where $|T|$ denotes the number of edgels in template T, and $\| \cdot \|_2$ the l_2 norm. The chamfer distance thus gives the mean distance of edgels in T to their closest edgels in E. For clarity, we will omit the superscript (T, E) below where possible.

The distance is efficiently computed via the *distance transform* (DT) which gives the distances of the closest points in E:

$$\text{DT}_E(\mathbf{x}) = \min_{\mathbf{x}_{\text{e}} \in E} \|\mathbf{x} - \mathbf{x}_{\text{e}}\|_2 \ , \quad (2)$$

and hence the min operation in (1) becomes a simple look-up:

$$d_{\text{cham}}(\mathbf{x}) = \frac{1}{|T|} \sum_{\mathbf{x}_{\text{t}} \in T} \text{DT}_E(\mathbf{x}_{\text{t}} + \mathbf{x}) \ . \quad (3)$$

We also compute the *argument* distance transform (ADT) which gives the *locations* of the closest points in E:

$$\mathrm{ADT}_E(\mathbf{x}) = \arg\min_{\mathbf{x}_e \in E} \|\mathbf{x} - \mathbf{x}_e\|_2 \ . \tag{4}$$

The exact Euclidean DT and ADT can be computed simultaneously in linear time [21].

It is standard practice to truncate the distance transform to a value τ:

$$\mathrm{DT}_E^\tau(\mathbf{x}) = \min(\mathrm{DT}_E(\mathbf{x}), \tau) \ . \tag{5}$$

so that missing edgels due to noisy edge detection do not have too severe an effect. Additionally it allows normalization to a standard range $[0, 1]$:

$$d_{\mathrm{cham},\tau}(\mathbf{x}) = \frac{1}{\tau|T|} \sum_{\mathbf{x}_t \in T} \mathrm{DT}_E^\tau(\mathbf{x}_t + \mathbf{x}) \ . \tag{6}$$

1) Edge orientation: Additional robustness is obtained by exploiting edge orientation information. This cue alleviates problems caused by clutter edgels which are unlikely to align in both orientation and position. One popular extension to basic chamfer matching is to divide the edge map and template into discrete orientation channels and sum the individual chamfer scores [49]. However, it is not clear how many channels to use, nor how to avoid artifacts at the channel boundaries.

Building on [41], we instead augment the robust chamfer distance (6) with a continuous and explicit cost for orientation mismatch, given by the mean difference in orientation between edgels in template T and the nearest edgels in edge map E:

$$d_{\mathrm{orient}}(\mathbf{x}) = \frac{2}{\pi|T|} \sum_{\mathbf{x}_t \in T} |\phi(\mathbf{x}_t) - \phi(\mathrm{ADT}_E(\mathbf{x}_t + \mathbf{x}))| \ . \tag{7}$$

The function $\phi(\mathbf{x})$ gives the orientation of edgel $\mathbf{x}$ modulo π, and $|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)|$ gives the smallest circular difference between $\phi(\mathbf{x}_1)$ and $\phi(\mathbf{x}_2)$. Edgels are taken modulo π because, for edgels on the outline of an object, the sign of the edgel gradient is not a reliable signal as it depends on the intensity of the background. The normalization by $\frac{\pi}{2}$ ensures that $d_{\mathrm{orient}}(\mathbf{x}) \in [0, 1]$.

Our improved matching scheme, called *oriented chamfer matching* (OCM), uses a simple linear interpolation between the distance and orientation terms

$$d_\lambda(\mathbf{x}) = (1 - \lambda) \cdot d_{\mathrm{cham},\tau}(\mathbf{x}) + \lambda \cdot d_{\mathrm{orient}}(\mathbf{x}) \ , \tag{8}$$

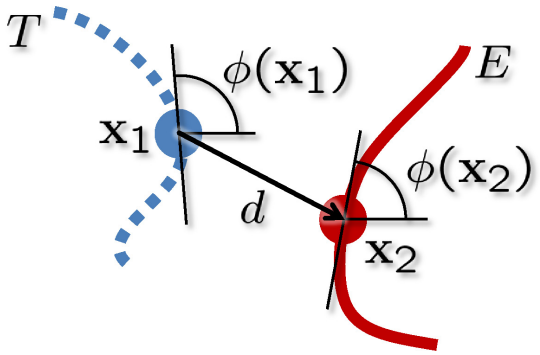

Fig. 3. **Oriented chamfer matching.** For edgel $\mathbf{x}_1$ in template T, the contribution to the OCM distance is determined by the distance d from $\mathbf{x}_1$ to the nearest edgel $\mathbf{x}_2$ in edge map E, and the difference between the edgel gradients at these points, $|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)|$.

with *orientation specificity* parameter λ. As we shall see below, λ is learned for each contour fragment separately, giving improved discrimination power compared with a shared, constant λ. The terms in (8) are illustrated in Figure 3. Note that OCM is considerably more storage efficient than using discrete orientation channels. In Section VI-C.1, we show that the continuous use of orientation information in OCM gives considerably improved performance compared with 8-channel chamfer matching and Hausdorff matching [30] (essentially (1) with the summation replaced by a maximization).

2) Matching at multiple scales: We extend OCM to multiple scales by simply rescaling the templates T. Treating T as now a set of *scale-normalized* edgels, to perform OCM at scale s between T and the original unscaled edge map E, we use the scaled edgel set $sT = \{s\mathbf{x}_t \text{ s.t. } \mathbf{x}_t \in T\}$ and calculate:

$$d_\lambda^{(T,E)}(\mathbf{x}, s) = d_\lambda^{(sT,E)}(\mathbf{x}) \ , \tag{9}$$

rounding scaled edgel positions to the nearest integer.

3) Approximate chamfer matching: For efficiency, one does not need to perform the complete sums over template edgels in (6) and (7). Each sum represents an empirical average, and so one can sum over only a fraction of the edgels, adjusting the normalization accordingly. This provides a good approximation to the true chamfer distance function in considerably reduced time. In practice, even matching only 20% of edgels gave no decrease in detection performance, as demonstrated in Section VI-C.3.

B. Building a Codebook of Contour Fragments

We need now a 'codebook', a set of representative contour fragments, and there is a choice in their class-specificity. One could use completely generic fragments such as lines, corners, and T-junctions and hope that in combination they can be made discriminative [25]. Instead, we create a class-specific codebook so that, for instance, the class horse results in, among others, 'head', 'back', and 'forelegs' fragments, as illustrated in Figure 6. Even individually, these fragments can be indicative of object presence in an image, and in combination will prove very powerful for object detection.

The outline of our codebook learning algorithm is as follows. We start with a large initial set of fragments, randomly chosen from edge maps. These are then clustered based on appearance. Finally, each cluster is subdivided to find fragments that agree in centroid position. The resulting sub-clusters form the codebook.

The initial set of fragments is generated thus. A rectangle $r = (\mathbf{r}_{\text{tl}}, \mathbf{r}_{\text{br}})$ enclosed within bounding box b of a random object is chosen, uniformly at random. We define vector $\mathbf{x}_{\text{f}} = \frac{1}{s}(\mathbf{r}_{\text{cen}} - \mathbf{x})$ as the scale-normalized vector from the object centroid $\mathbf{x}$ to the rectangle center $\mathbf{r}_{\text{cen}} = \frac{1}{2}(\mathbf{r}_{\text{tl}} + \mathbf{r}_{\text{br}})$. Let $E_r = \{\mathbf{x}_{\text{r}}\}$ denote the set of absolute image positions of edgels within rectangle r. The template T used in OCM is then:

$$T = \left\{ \frac{1}{s}(\mathbf{x}_{\text{r}} - \mathbf{r}_{\text{cen}}) \text{ s.t. } \mathbf{x}_{\text{r}} \in E_r \right\} . \qquad (10)$$

To remove overly generic fragments such as small straight lines, fragments with edgel density $\frac{|E_r|}{\text{area}(r)}$ below a threshold η_1 are immediately discarded. Fragments with edgel density above a threshold η_2 are also discarded, since these are likely to contain many background clutter edgels and even if not, will be expensive to match. Edgel sets E_r are computed as $E_r = \{\mathbf{x} \in C \text{ s.t. } \mathbf{x} \in r \text{ and } \|\nabla I\|_{\mathbf{x}} > t\}$. This equation uses the image gradient $\|\nabla I\|$ at the set of edge points C, given by the Canny non-maximal suppression algorithm. Rather than fix an arbitrary threshold t, we choose a random t for each fragment (uniformly, within the central 50% of the range $[\min_{\mathbf{x}} \|\nabla I\|_{\mathbf{x}}, \max_{\mathbf{x}} \|\nabla I\|_{\mathbf{x}}]$), so that at least some initial fragments are relatively clutter-free. As we shall see shortly, the clustering step then picks out these cleaner fragments to use as exemplars.

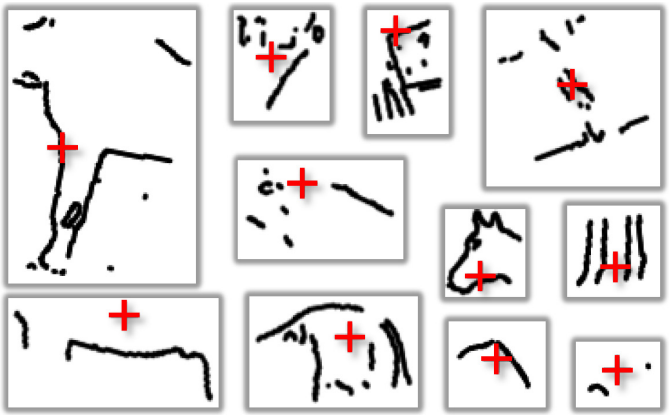

Fig. 4. **Initial set of contour fragments.** Examples of contour fragments extracted at random from the edge maps of horse images. The +s represent the fragment origins, i.e. vectors $(0,0)^T$ in (10). Many fragments are noisy, and so we apply a clustering step to find the cleaner fragments.

Finally, to ensure the initial set of contour fragments covers the possible appearances of an object, a small random transformation is applied to each fragment.[1] Several differently perturbed but otherwise similar fragments are likely to result, given the large number of fragments extracted.

1) Fragment clustering: Figure 4 shows example fragments extracted at random. While many fragments are quite noisy, some fragments are uncluttered, due to particular clean training images and the use of random edge thresholds. A clustering step is therefore employed with the intuition that these uncluttered fragments should lie at the cluster centers.

To this end, all pairs T_i and T_j of fragments in the initial set are compared in a symmetric fashion as follows:

$$d_{i,j} = d_{\lambda}^{(s_j T_i, s_j T_j)}(\mathbf{0}) + d_{\lambda}^{(s_i T_j, s_i T_i)}(\mathbf{0}) , \qquad (11)$$

scaling the fragments (first both to s_j, then both to s_i) and comparing at zero relative offset. Clustering is performed on distances $d_{i,j}$ using the k-medoids algorithm, the analogue of k-means for non-metric spaces. For the experiments in this paper, a constant $\lambda = 0.4$ was used for clustering, chosen to maximize the difference between histograms of distances

[1] The following transformations are chosen uniformly at random: a scaling $\log s \in [-\log s_{\text{r}}, \log s_{\text{r}}]$ and rotation $\theta \in [-\theta_{\text{r}}, \theta_{\text{r}}]$ about the fragment center is applied to the edgels, and the vector $\mathbf{x}_{\text{f}}$ is translated (by $x \in [-x_{\text{r}}, x_{\text{r}}]$ and $y \in [-x_{\text{r}}, x_{\text{r}}]$) and rotated (by $\phi \in [-\phi_{\text{r}}, \phi_{\text{r}}]$) about the object centroid. As we showed in [46], these transformations are crucial to ensure good performance, due to the limited training data and the use of rigid templates.

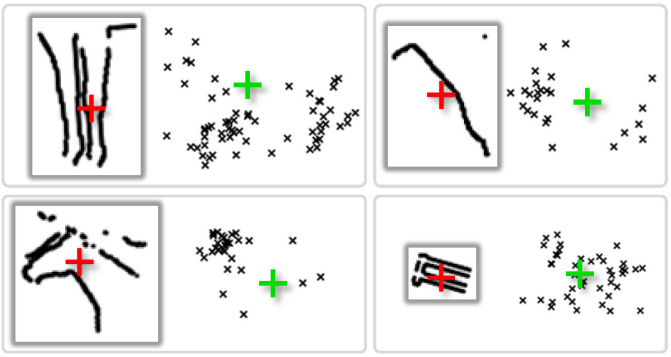

Fig. 5. **Clustering on appearance only.** Four example clusters that have low mutual chamfer distances (11), with (left) the cluster exemplar and (right) the votes (small Xs) of all members for vector $\mathbf{x}_f$ from the object centroid (+). Observe (top left) a 'legs' cluster has resulted in two modes (front and hind) in the voting space. On the bottom row, we see that (left) a very class specific 'head' cluster has highly consistent votes, whereas (right) a background cluster has uniformly scattered votes. To produce a unique centroid vote and remove background fragments, a sub-clustering step is performed.

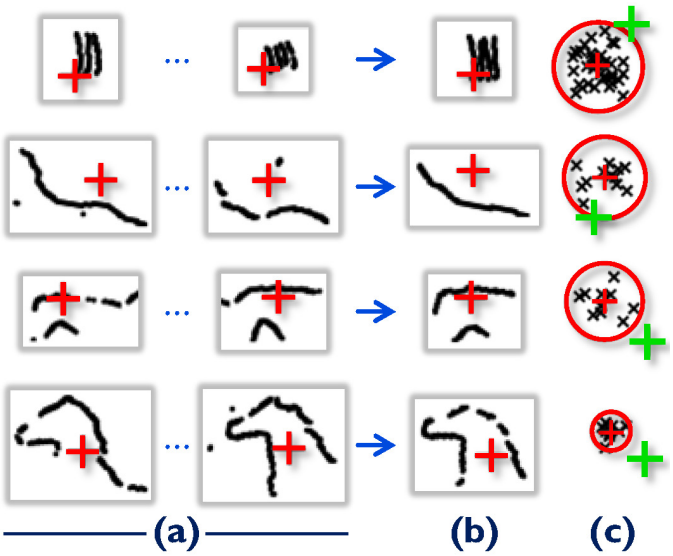

Fig. 6. **Clustering on appearance and centroid location.** Example sub-clusters that have low mutual chamfer distances (11) and agree on centroid location. From top to bottom: 'front legs', 'back', 'neck', and 'head'. **(a)** Example members of the sub-cluster. **(b)** Exemplar contour fragments (centers of the sub-clusters). **(c)** Votes (Xs) from the centroid (+), with their mean $\bar{\mathbf{x}}_f$ (+) and radial uncertainty σ (red circle). Note that we obtain uncluttered, class-specific exemplars, with an accurate estimate of location and uncertainty relative to the object centroid.

$d_{i,j}$ for within-cluster and between-cluster fragment pairs.

Example fragment clusters are shown in Figure 5. Clusters contain relatively uncluttered contour fragments of similar appearance. However, this purely appearance-based clustering does not take the vectors $\mathbf{x}_f$ from the object centroid into account. We desire each fragment to give a unique estimate of the object centroid, and so split each cluster into sub-clusters which agree on $\mathbf{x}_f$. Each fragment casts a vote for the object centroid, and modes in the voting space are found using mean shift mode estimation [14]. Each mode defines a sub-cluster, containing all fragments within a certain radius. To ensure high quality sub-clusters, only those with a sufficient number of fragments are kept (for our experiments, five fragments were required). Mode detection is iterated for unassigned fragments until no new sub-clusters are generated.

Contour fragments within each sub-cluster now agree both in appearance (11) and location $\mathbf{x}_f$ relative to the object centroid, shown in Figure 6. From noisy edge maps, our algorithm has selected uncluttered and class specific fragments, since random background fragments are highly unlikely to agree in position as well as appearance. Within each sub-cluster, the central fragment $\bar{T}$ with lowest average distance to the other fragments is used as an exemplar, together with the mean $\bar{\mathbf{x}}_f$ and radial variance σ of the centroid votes $\mathbf{x}_f$ (cf. Figure 2). We show below how boosting selects particular sub-clusters to use as the parts $F = (\bar{T}, \bar{\mathbf{x}}_f, \sigma)$ in the model.

The clustering step is somewhat similar to that used in [33], except that we cluster contour fragments rather than image patches, and each resulting sub-cluster has only one particular location relative to the centroid. Also observe that we have taken a rather unconstrained approach to choosing contour fragments. Research from psychology [17] analyzed theories of how to split outline contours into fragments for optimal recognition by humans, for example at points of extremal curvature. It would be interesting future work to investigate such ideas applied in a computer-based system.

IV. Object Detection

In this section, we describe how contour exemplars are combined in a boosted sliding window classifier. Parts are matched to an edge map using OCM with priors on their spatial layout. The classifier is evaluated across the scale-space of the image, and mean shift produces a final set of confidence-valued object detections. The only image information used by the detector is the edge map E, computed using the Canny edge detector [12].

For an object centroid hypothesis with location $\mathbf{x}$ and scale s, part F is expected to match the edge map E at position $\hat{\mathbf{x}} = \mathbf{x} + s\bar{\mathbf{x}}_{\mathrm{f}}$, with spatial uncertainty $s\sigma$. The chamfer distance is weighted with a cost that increases away from the expected position, and minimizing this weighted distance gives a degree of spatial flexibility, allowing parts to 'snap' into place. The location of the minimum is given by

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}'} \left(d_\lambda^{(\bar{T},E)}(\mathbf{x}', s) + w_{s\sigma}(\|\mathbf{x}' - \hat{\mathbf{x}}\|_2) \right) , \tag{12}$$

where $w_\sigma(x)$ is the radially symmetric spatial weighting function[2] for which we use the quadratic

$$w_\sigma(x) = \begin{cases} \frac{x^2}{\sigma^2} & \text{if } |x| \leq \sigma \\ \infty & \text{otherwise.} \end{cases} \tag{13}$$

The *part response* v at centroid hypothesis $(\mathbf{x}, s)$ is defined as the chamfer distance at the best match $\mathbf{x}^\star$

$$v_{[F,\lambda]}(\mathbf{x}, s) = d_\lambda^{(\bar{T},E)}(\mathbf{x}^\star, s) , \tag{14}$$

and this is used in the classifier, described next.

A. Detecting Objects

Sliding window classification [4], [25], [52] is a simple, effective technique for object detection. A probability $P(\mathrm{obj}_{(\mathbf{x},s)})$ of object presence at location $(\mathbf{x}, s)$ is calculated across scale-space using a boosted classifier which combines multiple part responses v (14). These probabilities are far from independent: for example, the presence of two distinct neighboring detections is highly unlikely. Hence a mode detection step selects local maxima as the final set of detections.

One must choose a set $\mathcal{X}$ of centroid scale-space location hypotheses, sampled frequently enough to allow detection of all objects present. We use a fixed number of test scales, equally spaced logarithmically to cover the range of scales in the training data. Space is sampled over a regular grid with spacing $s\Delta_{\mathrm{grid}}$ for constant Δ_{grid} (optimized by holdout validation). Increasing the spacing with scale is possible since the search window in (12) is proportionally enlarged.

[2]The hard cut-off at σ limits the search range and thus improves efficiency. In practice, increasing the cut-off radius did not appear to improve performance.

1) Classifier: We employ a boosted classifier to compute probabilities $P(\mathrm{obj}_{(\mathbf{x},s)})$. This combines part responses v (14) for parts $F_1, \ldots, F_M$ as

$$H(\mathbf{x}, s) = \sum_{m=1}^{M} a_m [v_{[F_m,\lambda_m]}(\mathbf{x}, s) > \theta_m] + b_m , \tag{15}$$

where $[\cdot]$ is the zero-one indicator, and (λ, a, b, θ) are learned parameters (see ahead to Section V). Each term in the sum corresponds to a part in the model, and is a decision stump which assigns a weak confidence value according to the comparison of part response $v_{[F_m,\lambda_m]}$ to threshold θ_m. The weak decision stump confidences are summed to produce a strong confidence H, which is then interpreted as a probability using the logistic transformation [27]:

$$P(\mathrm{obj}_{(\mathbf{x},s)}) = [1 + \exp(-H(\mathbf{x}, s))]^{-1} . \tag{16}$$

2) Mode detection: We employ the powerful technique of mean shift mode estimation [14] on the hypothesized locations $(\mathbf{x}, s) \in \mathcal{X}$, weighted by their scaled posterior probabilities $s^2 P(\mathrm{obj}_{(\mathbf{x},s)})$, similarly to [34]. Multiplying by s^2 compensates for the proportionally less dense hypotheses at larger scales. The algorithm models the non-parametric distribution over the hypothesis space with the kernel density estimator

$$P(\mathbf{x}, s) \propto \sum_{(\mathbf{x}_i,s_i)\in\mathcal{X}} s_i^2 P(\mathrm{obj}_{(\mathbf{x}_i,s_i)}) \; K\left(\frac{x^{\mathrm{x}} - x_i^{\mathrm{x}}}{h_{\mathrm{x}}}, \frac{x^{\mathrm{y}} - x_i^{\mathrm{y}}}{h_{\mathrm{y}}}, \frac{\log s - \log s_i}{h_{\mathrm{s}}} \right) , \tag{17}$$

where Gaussian kernel K uses bandwidths h_{x}, h_{y} and h_{s} for the x, y, and scale dimensions respectively (the scale dimension is linearized by taking logarithms). Mean shift efficiently locates modes (local maxima) of the distribution which are used as the final set of detections. The density estimate at each mode is used as a confidence value for the detection.

V. Learning

We describe in this section how the classifier H (15) is learned using the Gentle AdaBoost algorithm [27]. This takes as input a set of training examples i, each consisting of feature vector $\mathbf{f}_i$ paired with target value $z_i = \pm 1$, and iteratively builds the classifier.

For our purposes, training example i represents location $(\mathbf{x}_i, s_i)$ in one of the training images. The

target value z_i specifies the presence ($z_i = +1$) or absence ($z_i = -1$) of the object class. The feature vector $\mathbf{f}_i$ contains the responses $v_{[F,\lambda]}(\mathbf{x}_i, s_i)$ (14) for all codebook entries F, and all OCM orientation specificities λ from a fixed set Λ. A given dimension d in the feature vector therefore encodes a pair (F, λ). The decision stump parameters a, b, and θ are learned as described in [50].

We are free to choose the number, locations, and target values of the training examples. One could densely sample each training image, computing feature vectors for examples at every point on a grid in scale-space. This is however unnecessarily inefficient because the minimization in (12) means that neighboring locations often have near identical feature vectors.

Instead, we use the sparse pattern of examples shown in Figure 7. For a training object at location $(\mathbf{x}, s)$, positive examples are taken at the 3x3x3 scaled grid locations $\mathbf{x}' = \mathbf{x} + (z_\mathrm{x} s' \delta_1, z_\mathrm{y} s' \delta_1)^T$ for scales $s' = s\gamma_1^{z_\mathrm{s}}$, where $(z_\mathrm{x}, z_\mathrm{y}, z_\mathrm{s}) \in \{-1, 0, +1\}^3$. The grid is spaced by δ_1 (scale-normalized) and scaled by γ_1. The positive examples ensure a strong classification response near the true centroid, wide enough that the sliding window classifier need not be evaluated at every pixel. To ensure the response is localized, negative examples are taken at positions $\mathbf{x}' = \mathbf{x} + (z_\mathrm{x} s' \delta_2, z_\mathrm{y} s' \delta_2)^T$ for scales $s' = s\gamma_2^{z_\mathrm{s}}$, with a larger spacing $\delta_2 > \delta_1$ and scaling $\gamma_2 > \gamma_1$, and the same $(z_\mathrm{x}, z_\mathrm{y}, z_\mathrm{s})$ but now excluding $(0, 0, 0)$. This particular pattern results in a total of 53 examples for each object, which is vastly less than the total number of scale-space locations in the image. For training images not containing an object, we create all negative examples in the same pattern, at a number of random scale-space locations.

Feature vectors are pre-computed for all examples, usually taking less than an hour on a modern machine. Boosting itself is then very quick, taking typically less than a minute to converge, since the weak learners are individually quite powerful. A cascade [52] is also learned, which resulted in a five-fold reduction in the average number of response calculations at test time.

A. Retraining on Training Data

It is unclear how to place the sparse negative training examples optimally throughout the training images, and hence they are initially placed at random. However, once a detector is learned from these examples, a retraining step is used to boot-strap the set of training examples [54]. We evaluate the detector on the training images, and record any false positives or negatives (see ahead to Section VI-A). The classifier is then retrained on the original example set, augmented with new negative examples at the locations of false positives [16], and duplicate positive examples to correct the false negatives. We demonstrate in Section VI-C.2 that this procedure allows more parts to be learned without over-fitting.

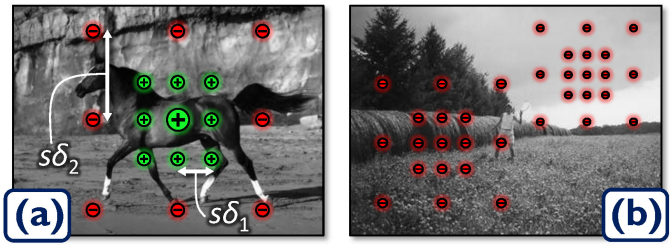

Fig. 7. **Training examples. (a)** A pattern of positive (⊕) and negative (⊖) examples are arranged about the true object centroid (the central, larger ⊕). The positive and negative examples are spaced on grids of size δ_1 and δ_2 respectively, scaled by the ground-truth object scale s. The boosting algorithm trains from feature vectors of part responses (14) computed at these examples. **(b)** For images with no objects present, all negative copies of the same pattern are placed at a number of random scale-space locations. For clarity, only one scale $z_\mathrm{s} = 0$ is shown (see text).

B. Retraining on Test Data

The same idea can be put to work on the *test* data, if one assigns a degree of trust to the output of the classifier. One can take a fixed proportion ξ (e.g. $\xi = 10\%$) of detections with strongest confidence and assume these are correct, positive detections, and the same proportion of detections with weakest confidence and assume there are no objects present at those locations. The boosted classifier is retrained with the new positive and negative training examples further augmenting the training set.

VI. Evaluation

We present a thorough evaluation of the classification and detection performance of our technique on several challenging datasets, investigating different aspects of our system individually, and comparing against other state-of-the-art methods. The standard experimental procedure is detailed in Section VI-A, the description of the datasets in Section VI-B, and the results in Section VI-C.

A. Procedure

The image datasets are split into training and test sets. Each model is learned from the training set with only ground-truth bounding boxes provided. At test time, the bounding boxes are used only for evaluating accuracy.

Mode detection results in a set of centroid hypotheses and confidences of object presence at these points. We assign a scaled bounding box centered on each detection, with aspect-ratio proportional to that of the average training bounding box. For a detection to be marked as correct, its inferred bounding box b_{inf} must agree with the ground truth bounding box b_{gt} based on an overlap criterion as $\frac{\mathrm{area}(b_{\mathrm{inf}} \cap b_{\mathrm{gt}})}{\mathrm{area}(b_{\mathrm{inf}} \cup b_{\mathrm{gt}})} > 0.5$ (from [2]). Each b_{gt} can match to only one b_{inf}, and so spurious detections of the same object count as false positives. For image classification, we use the confidence of the single most confident detection within each image.

The receiver operating characteristic (ROC) curve is used to measure classification accuracy. This plots the trade-off between false positives and false negatives as a global confidence threshold is varied. The equal error rate (EER) gives an easily interpretable accuracy measure, while the area under the curve (AUC) takes the whole curve into account and so gives a better measure for comparison purposes.

For detection we use two closely related measures. The first, the recall-precision (RP) curve, plots the trade-off between recall and precision as one varies the global threshold. For comparison with previous work we quote the EER measure on the RP curve, but for new results we report the more representative AUC measure. The second measure plots recall against the average number of false positives per image (RFPPI) as the detection threshold is varied [25]. The RFPPI curve seems more natural than RP for human interpretation since it is monotonic and stabilizes as more negative images are tested (the RP curve can only deteriorate). However it gives no overall quantitative score, and so the legends in Figures 8 and 11 contain RP AUC figures even though the graphs show RFPPI.

B. Datasets

1) Weizmann Horses [10]: This is a challenging set of side-on horse images, containing different breeds, colors, and textures, with varied articulations, lighting conditions, and scales. While nominally viewed side-on, considerable out-of-plane rotation is evident. We paired this with the difficult Caltech 101 background set [3], [19]. While these images have different textural characteristics, they contain many clutter edges that pose a hard challenge to our contour-only detector. Images were down-sampled to a maximum dimension of 320 pixels where necessary. The resulting objects have a scale range of roughly 2.5x from smallest to largest. The first 50 horse and background images were used for training, the next 50 for holdout validation, and the final 228 as the test set. We also compare against our earlier work [46] using a single-scale horse database. The datasets are available at [1].

2) Graz 17: We compare against [43] on their 17 class database (listed in Table II). As closely as possible, we use the same training and test sets. Images are down-sampled to a maximum dimension of 320 pixels. For some classes, the resulting scale range is more than 5x. We test each class individually, paired with an equal number of background test images.

C. Results

1) Matching measures: First, we compare the performance of the object detector using several different matching measures: our proposed OCM with learned λ and with constant $\lambda \in \{0, 0.5, 1\}$, standard 8-channel chamfer matching, and Hausdorff matching. The experiment was performed against 100 images in the Weizmann test set using 100 parts without retraining (other parameter settings are specified below).

Figure 8 superimposes the RFPPI curves for each matching measure, and the legend reports the corresponding RP AUC statistics. Observe that with no orientation information ($\lambda = 0$, identical to 1-channel, non-oriented chamfer matching), performance is very poor. The Hausdorff distance also fails to work well, since it too does not use orientation information. The 8-channel chamfer matching performs fairly well, but by modeling orientation continuously, our OCM (for $\lambda > 0$) performs as well or better, even if λ is kept constant. The RFPPI curve for $\lambda = 1$ appears almost as good as the learned λ curve, although the AUC numbers confirm that learning λ per part is noticeably better. However, the extra expense of learning per-part λ values may mitigate its quantitative advantages in some applications.

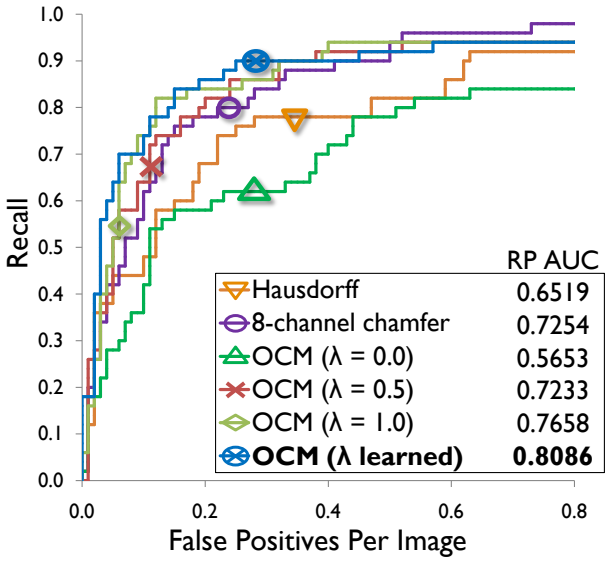

Fig. 8. **Detection performance of different contour matching measures.** Recall is plotted as a function of the number of false positives per image averaged over the Weizmann test subset. The best performance is obtained by our OCM technique with learned λ parameter, although fixed $\lambda = 1$ also performs well.

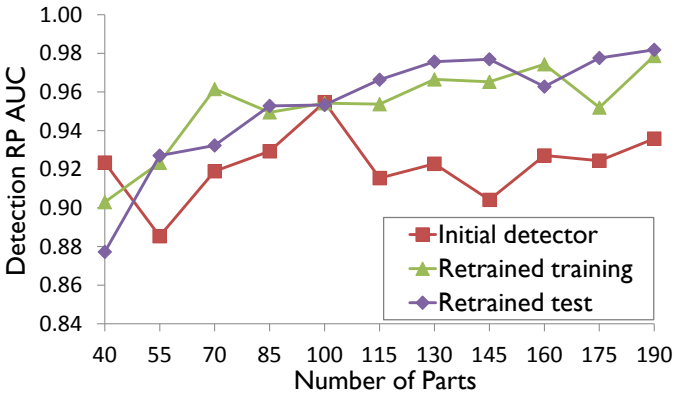

Fig. 9. **Effect of retraining.** Detection performance is graphed as a function of number of parts (rounds of boosting). The initial detector starts to over-fit as the number of parts is increased above 100. Retraining prevents this over-fitting allowing an overall performance improvement at the expense of more parts.

2) Retraining: As described in Sections V-A and V-B, one can boot-strap the detector by retraining. For this experiment on the Weizmann validation set, we recorded the RP AUC against the number of parts: (i) without retraining, (ii) retraining only on the training data ('retrained training' in Figures 9 and 11), and (iii) retraining both on the training and test data ('retrained test'). The confidence parameter was set to $\xi = 10\%$.

We can draw several conclusions from the results in Figure 9. Adding more parts helps performance on the test data up to a point, but eventually the detector starts to over-fit to the training data and generalization decreases. By providing more training examples by retraining on the training data, we can use more parts without over-fitting. Retraining on the test data maintains the additional accuracy, and gives a further improvement on the full test set, as described below. With only 40 parts, retraining on the test data decreases performance, since the strongest and weakest detections are not sufficiently reliable. Note that retraining does entail significant extra effort for a relatively modest performance gain.

3) Approximate chamfer matching: All results in our evaluation make use of the approximation of Section III-A.3, whereby only a subset of fragment edgels are used for chamfer matching. We used only every fifth edgel (scan-line order) in each fragment, which gave a commensurate speed improvement. We compared detection performance with and without the approximation on the Weizmann validation set, using 100 features. With the approximation, 0.9547 RP AUC was achieved, whereas without the approximation (matching every edgel) only 0.9417 was obtained. We conclude that the approximation can improve speed without degrading detection performance. The slight improvement in performance may even be significant, since the variance of the training part responses is increased slightly, which may prevent over-fitting.

4) Multi-scale Weizmann horses: We now evaluate on the full Weizmann dataset, showing example detections in Figure 10 and quantitative results in Figure 11.

We draw several conclusions. Firstly, we have shown that retraining on both the training and test sets not only helps generalization, but actually considerably improves performance. Turning to Figure 10, we observe that the detector works very well on the challenging horse images, despite wide within-class variation, considerable background clutter and even silhouetting. Missed detections (false negatives) tend to occur when there is significant pose change or out-of-plane rotation beyond the range for which we would expect our side-on detector to work. Training explicitly for these poses or rotations, perhaps sharing features between views [50], would allow detection of these objects. False positives occur when the pattern of clutter edgels is sufficiently similar to our model, as for

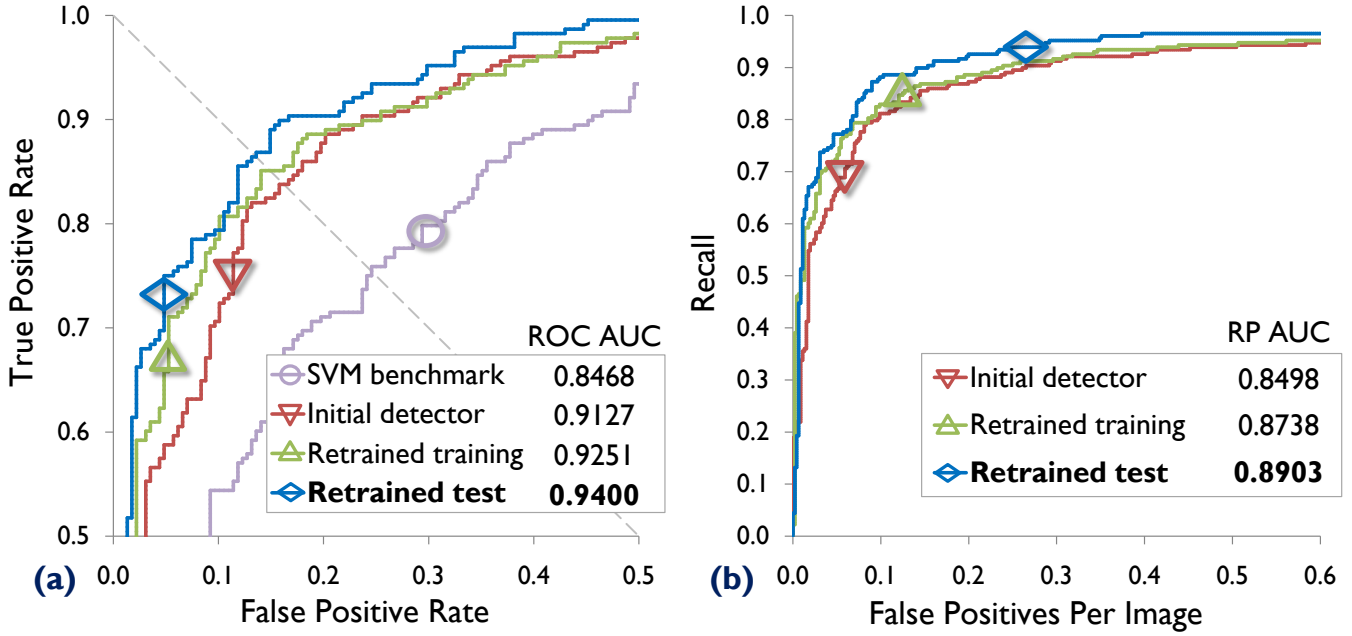

Fig. 11. **Performance on the Weizmann test set. (a)** ROC curves showing classification performance, with the curve for the SVM benchmark included (Section VI-C.7). **(b)** RFPPI curves showing detection performance. Note how both stages of retraining improve both classification and detection performance.

Fig. 10. **Example detections for the Weizmann test set.** Bounding boxes around objects indicate detections: green represents correct detections, red false positives, and yellow the ground-truth for false negatives. The final column visualizes the contour fragments for the neighboring detections. Note accurate scale-space localization in the presence of variable object appearance, background clutter, silhouetting, articulation, and pose changes.

example the case (third column, last row) of the man standing at the front of the horse, where in terms of image edges the man's legs look sufficiently similar to a horse's front legs. An investigation in [45] shows how cues based on texture and color can be combined with contour fragments to remove such false positives and improve overall performance.

Our C# implementation on a 2.2 GHz machine takes approximately 2 hours to train and 10 seconds per image to test. For these and all experiments, unless stated otherwise, the following parameters were used. The distance transform truncation was $\tau = 30$, and fragments were randomly chosen with the following transformation parameters: scaling $s_{\mathrm{r}} = 1.2$, rotation about the fragment center $\theta_{\mathrm{r}} = \frac{\pi}{8}$, (scale-normalized) translation $x_{\mathrm{r}} = 0.05$, and rotation about the centroid $\phi_{\mathrm{r}} = \frac{\pi}{16}$. To learn the dictionary, 10000 raw fragments, with edgel density bounded as $(\eta_1, \eta_2) = (1\%, 5\%)$, were clustered to produce 500 exemplars. To learn the classifier, examples were taken with grid spacings $(\delta_1, \delta_2) = (0.03, 0.25)$ and scalings $(\gamma_1, \gamma_2) = (1.1, 1.4)$. Three patterns of negative examples were used for background images, and λ was allowed values in $\Lambda = \{0, 0.2, \ldots, 1\}$. Evaluation used a grid spacing of $\Delta_{\mathrm{grid}} = 0.07$ scaled by each of 6 test scales over $M = 100$ rounds. The top and bottom $\xi = 10\%$ of detections were used for retraining on the test set.

5) Training from segmented data: To investigate the ability of the codebook learning algorithm to extract clean exemplars from unsegmented images, we repeated the detection experiment on the

Weizmann dataset, with the codebook now learned from segmented training data. We obtained 0.8637 RP AUC, slightly worse than the performance on unsegmented images (0.8903). This slight drop in performance is not particularly surprising given that no interior edges are present, but it does confirm the power of the codebook learning algorithm.

6) Learned edge detection: The Canny edge detector [12] has thus far proved a capable basis for our features. However, recent developments such as the Berkeley edge detector [38] and boosted edge learning (BEL) [18] take a more modern approach to edge detection, whereby a model of edges is learned from training data. We compared performance between the three edge detectors on the Weizmann test set, without using retraining. Two BEL models [18] were trained: one using natural image boundaries, the second using segmented horse images. The results are summarized in Table I.

TABLE I
PERFORMANCE USING DIFFERENT EDGE DETECTORS.

	Classification ROC AUC	Detection RP AUC
Canny	0.9127	0.8498
Berkeley [38]	0.9275	0.8871
BEL [18] Natural	0.9029	0.8354
BEL [18] Horse	0.9518	0.8976

The Berkeley detector performs considerably better than Canny, especially for detection. While the BEL trained on natural images gave no improvement, the BEL trained on segmented horse images performs the best of all detectors. Note that current implementations of both Berkeley and BEL are very much slower than Canny, and so these advances may not yet be useful in certain applications. This experiment has shown that a modern, learned edge detector complements our object detection system; future work remains to extend this evaluation to the other datasets in this paper.

7) SVM classification benchmark: To compare contour fragments against interest point based features, and to determine the challenge that the Weizmann horse dataset poses, we evaluated a benchmark using an SVM built on a bag-of-words representation [55]. In our experiment, SIFT [36] were extracted and clustered into a number of 'words'. Histograms of word counts for each image were computed, and a radial basis function SVM was trained to discriminate between class and background images. SVM parameters were optimized using cross validation, as were the numbers of clusters.

The ROC curve for the SVM benchmark is shown in Figure 11(a). We observe considerably worse performance than our contour based classifiers achieve. This suggests that the varied textures of the objects in this dataset cannot be characterized well by local descriptors. Our contour-based detector is however able to exploit the characteristic outline of the objects.

8) Single-scale Weizmann horses: Using the single-scale Weizmann horse dataset, we compare against [46], where 92.1% RP EER was achieved (using some segmented data). Experiments in [25] improved accuracy to 94.2% RP EER using contour-based features, and to 95.7% by combining contour features with local descriptors. Our method using no segmented training data and only contour features obtained 95.68% RP EER and 0.9496 RP AUC. This is as good as [25], but without needing the additional feature type.

9) Graz 17: We conclude our evaluation by evaluating on the Graz 17 class dataset. In Table II we compare our results to [43] (which subsumes the results of [42]), and in Figure 12 show example detections. Parameter values were unchanged from the previous multi-scale Weizmann experiments, although the number of parts and number of scales were optimized against the training data.

There are several conclusions to draw. Firstly, for most classes we perform comparably to [43], and for the larger (admittedly slightly more straightforward) datasets we show a significant improvement, with almost perfect performance on motorbikes. Classification proves easier than detection in most cases, since strong but poorly localized detections contribute positively to classification but negatively to detection. Performance is worse for a few classes, such as cars ($\frac{2}{3}$ rear) and cars (front), and poor for both techniques for bikes (front) and people. There are few training images for these classes, and objects exhibit considerably more out-of-plane rotation. Also, the small number of test images means that even one missed detection has a very large effect on the RP EER (up to $\frac{100}{N}\%$ for N test images). Much more significant therefore is our sustained improvement for classes with more test

TABLE II
CLASSIFICATION AND DETECTION PERFORMANCE ON THE GRAZ 17 DATASET, WITH COMPARISON TO [43].

	Number of images		Classification (ROC)		Detection (RP)		
Class	Training	Test	AUC	EER	AUC	**EER**	[43] EER
Airplanes	100	400	0.9953	3.4%	0.9310	**6.8%**	7.4%
Cars (rear)	100	400	0.9992	1.5%	0.9912	**1.8%**	2.3%
Motorbikes	100	400	1.0000	0.4%	1.0000	**0.3%**	4.4%
Faces	100	217	0.9966	2.4%	0.9850	**2.8%**	3.6%
Bikes (side)	90	53	0.9366	13.2%	0.6959	**32.1%**	28.0%
Bikes (rear)	29	13	0.9172	15.4%	0.6398	**26.7%**	25.0%
Bikes (front)	19	12	0.9375	16.7%	0.6344	**41.7%**	41.7%
Cars ($\frac{2}{3}$ rear)	32	14	0.9000	20.9%	0.6925	**30.0%**	12.5%
Cars (front)	34	16	0.9727	12.5%	0.7233	**29.4%**	10.0%
Bottles	54	64	0.9802	7.8%	0.9468	**9.4%**	9.0%
Cows (side)	45	65	0.9992	1.7%	0.9975	**1.5%**	0.0%
Horses (side)	55	96	0.9816	6.3%	0.9680	**6.3%**	8.2%
Horses (front)	44	22	0.9566	13.6%	0.7852	**27.3%**	13.8%
Cows (front)	34	16	0.9727	6.3%	0.8575	**18.8%**	18.0%
People	39	18	0.9321	16.7%	0.4271	**47.6%**	47.4%
Mugs	30	20	0.9600	5.0%	0.9035	**10.0%**	6.7%
Cups	31	20	0.9825	5.0%	0.9158	**15.0%**	18.8%

images.

VII. CONCLUSIONS AND FUTURE WORK

Our thorough evaluation has demonstrated that contour can be used to successfully recognize objects from a wide variety of object classes at multiple scales. Our new approximate oriented chamfer matching outperformed existing contour matching methods, and enabled us to build a class-specific codebook of uncluttered contour fragments from noisy training data. We observed that retraining on both the training and test data can improve generalization and test performance. Finally, we showed how modern, learned edge detection gave an improvement over the traditional Canny edge detector.

A. Future Work

We are interested in developing a more principled method to combine the classification probabilities from multiple sliding windows. We plan also to investigate further our codebook of contour fragments. The clustering algorithm is slightly inefficient, and perhaps agglomerative clustering would be faster. The codebook might also be used in a bag-of-words model. Our investigation of modern edge detection algorithms is also preliminary and more work is desirable here.

An eventual goal of object detection is both localization and segmentation of the object from the background. Preliminary segmentation results using our inferred object bounding rectangles as initialization to GrabCut [44] show promise. Individually segmented fragments could serve as a segmentation prior, similarly to [32]. An alternative method proposed in [56] is to learn to segment directly from the image. Eventually, edge detection, segmentation, and recognition should be combined at a fundamental level.

Acknowledgements: The authors are very grateful to the anonymous reviewers for their excellent input, to P. Dollár, G. Brostow, A. Zisserman, V. Ferrari, A. Opelt, and J. Winn, and to Microsoft Research Cambridge for funding.

REFERENCES

[1] `http://jamie.shotton.org/work/`.
[2] `http://www.pascal-network.org/challenges/VOC/`.
[3] `http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html`.
[4] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In A. Heyden, G. Sparr, and P. Johansen, editors, *Proc. European Conf. on Computer Vision*, volume LNCS 2353, pages 113–130. Springer, May 2002.
[5] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

Fig. 12. **Example detections and visualizations for the Graz 17 test set.** Note accurate localization even with wide intra-class appearance variation, significant scale and pose changes, partial occlusion, background clutter, and multiple objects.

[6] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.

[8] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 26–33, June 2005.

[9] I. Biederman and G. Ju. Surface vs. edge-based determinants of visual recognition. *Cognitive Psychology*, 20(1):38–64, January 1988.

[10] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentations. In *IEEE Workshop on Perceptual Organization in Computer Vision*, volume 4, page 46, 2004.

[11] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. Int. Conf. on Computer Vision*, volume 1, pages 105–112, Vancouver, Canada, July 2001.

[12] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[13] O. Carmichael and M. Hebert. Shape-based recognition of wiry objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(12):1537–1552, December 2004.

[14] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[15] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 886–893, June 2005.

[17] J. De Winter and J. Wagemans. Contour-based object identification and segmentation: stimuli, norms and data, and software tools. *Behavior Research Methods, Instruments, and Computers*, 36(4):604–624, November 2004.

[18] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 1964–1971, 2006.

[19] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006.

[20] P.F. Felzenszwalb. Learning models for object recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1056–1062, December 2001.

[21] P.F. Felzenszwalb and D.P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell, 2004.

[22] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Computer Vision*, 61(1):55–79, January 2005.

[23] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 264–271, June 2003.

[24] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In T. Pajdla and J. Matas, editors, *Proc. European Conf. on Computer Vision*, volume LNCS 3021, pages 242–256. Springer, May 2004.

[25] V. Ferrari and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2007.

[26] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proc. European Conf. on Computer Vision*, volume LNCS 3953, pages 14–28. Springer, May 2006.

[27] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[28] D.M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 439–444, August 1998.

[29] D.M. Gavrila. Pedestrian detection from a moving vehicle. In D. Vernon, editor, *Proc. European Conf. on Computer Vision*, volume LNCS 1843, pages 37–49. Springer, June 2000.

[30] D.P. Huttenlocher and W.J. Rucklidge. A multi-resolution technique for comparing images using the hausdorff distance. Technical Report TR 92-1321, Department of Computer Science, Cornell University, December 1992.

[31] M.P. Kumar, P.H.S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *Proc. British Machine Vision Conference*, 2004.

[32] M.P. Kumar, P.H.S. Torr, and A. Zisserman. OBJ CUT. In

Proc. IEEE Conf. Computer Vision and Pattern Recognition, volume 1, pages 18–25, June 2005.

[33] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *Proc. British Machine Vision Conference*, volume II, pages 264–271, 2003.

[34] B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *DAGM'04: 26th Pattern Recognition Symposium*, pages 145–153, June 2004.

[35] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 878–885, June 2005.

[36] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, November 2004.

[37] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. W. H. Freeman, 1982.

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

[39] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proc. British Machine Vision Conference*, volume 2, pages 779–788, September 2003.

[40] R.C. Nelson and A. Selinger. A Cubist approach to object recognition. In *Proc. Int. Conf. on Computer Vision*, pages 614–621, Bombay, India, January 1998.

[41] C.F. Olson and D.P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Trans. on Image Processing*, 6(1):103–113, January 1997.

[42] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proc. European Conf. on Computer Vision*, volume 2, pages 575–588, Graz, Austria, May 2006. Springer.

[43] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 3–10, June 2006.

[44] C. Rother, V. Kolmogorov, and A. Blake. GrabCut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.

[45] J. Shotton. *Contour and Texture for Visual Recognition of Object Categories*. PhD thesis, University of Cambridge, March 2007.

[46] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proc. Int. Conf. on Computer Vision*, volume 1, pages 503–510, Beijing, China, October 2005.

[47] J. Sivic, B.C. Russel, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their localization in images. In *Proc. Int. Conf. on Computer Vision*, volume 1, pages 370–377, Beijing, China, October 2005.

[48] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1470–1477, Nice, France, October 2003.

[49] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384, September 2006.

[50] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):854–869, May 2007.

[51] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. J. Computer Vision*, 48(1):9–19, June 2002.

[52] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 511–518, December 2001.

[53] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 37–44, June 2006.

[54] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[55] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classificaiton of texture and object categories: A comprehensive study. *Int. J. Computer Vision*, 73(2):213–238, 2007.

[56] S. Zheng, Z. Tu, and A.L. Yuille. Detecting object boundaries using low-, mid- and high-level information. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

Jamie Shotton graduated from the University of Cambridge with the BA degree (computer science) in 2002, and the PhD (computer vision) in 2007. He is currently a research fellow at the Toshiba Corporate R&D Center in Kawasaki, Japan. His research interests include computer vision, object recognition, machine learning, computational photography, and robotics.

Andrew Blake has been on the faculty of computer science at the University of Edinburgh and also a Royal Society Research Fellow, until 1987, then at the Department of Engineering Science in the University of Oxford, where he became a professor in 1996, and a Royal Society Senior Research Fellow in 1998. In 1999, he was appointed senior research scientist at Microsoft Research, Cambridge, while continuing as visiting professor at Oxford. His research interests are in computer vision, signal processing, and learning. He has published a number of papers in vision, a book with A. Zisserman (Visual Reconstruction, MIT Press), edited Active Vision with Alan Yuille (MIT Press), and the book (Active Contours, Springer-Verlag) with Michael Isard. He was elected fellow of the Royal Academy of Engineering in 1998. He is a member of the IEEE and the IEEE Computer Society.

Roberto Cipolla received the BA degree (engineering) from the University of Cambridge in 1984 and the MSE degree (electrical engineering) from the University of Pennsylvania in 1985. In 1991, he was awarded the D.Phil. degree (computer vision) from the University of Oxford. His research interests are in computer vision and robotics and include the recovery of motion and 3D shape of visible surfaces from image sequences, visual tracking and navigation, robot hand-eye coordination, algebraic and geometric invariants for object recognition and perceptual grouping, novel man-machine interfaces using visual gestures, and visual inspection. He has authored three books, edited six volumes, and coauthored more than 200 papers. He is a member of the IEEE and the IEEE Computer Society.