in Proc. 13th International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011

From Contours to 3D Object Detection and Pose Estimation

Nadia Payet and Sinisa Todorovic
Oregon State University
Corvallis, OR 97331, USA
payetn@onid.orst.edu, sinisa@eecs.oregonstate.edu

Abstract

This paper addresses view-invariant object detection and pose estimation from a single image. While recent work focuses on object-centered representations of point-based object features, we revisit the viewer-centered framework, and use image contours as basic features. Given training examples of arbitrary views of an object, we learn a sparse object model in terms of a few view-dependent shape templates. The shape templates are jointly used for detecting object occurrences and estimating their 3D poses in a new image. Instrumental to this is our new mid-level feature, called bag of boundaries (BOB), aimed at lifting from individual edges toward their more informative summaries for identifying object boundaries amidst the background clutter. In inference, BOBs are placed on deformable grids both in the image and the shape templates, and then matched. This is formulated as a convex optimization problem that accommodates invariance to non-rigid, locally affine shape deformations. Evaluation on benchmark datasets demonstrates our competitive results relative to the state of the art.

1. Introduction

We study multi-view object detection and pose estimation in a single image. These problems are challenging, because appearances of 3D objects may differ significantly within a category and when seen from different viewpoints. A majority of recent work resorts to the object-centered framework, where statistical generative models [16, 22, 17, 1, 10, 7], discriminative models [6], or view-independent implicit shape models [15, 18] are used to encode how local object features (e.g. points or edgeless), and their spatial relationships vary in the images as the camera viewpoint changes. They strongly argue against certain limitations of viewer-centered approaches that apply several single-view detectors independently, and then combine their responses [21, 13]. In the light of the age-long debate whether viewer- or object-centered representations are more suitable for 3D object recognition [4, 20], the recent trend to readily dismiss viewpoint-dependent approaches seems too hasty.

In this paper, we revisit the viewer-centered framework. We are motivated by two widely recognized findings in psychophysics and cognitive psychology that: (i) shape is one of the most categorical object properties [3], and (ii) viewpoint-dependent object representations generalize well across members of perceptually-defined classes [20]. These findings motivate our new approach that uses a number of viewpoint-specific shape representations to model an object category. Shape is typically more invariant to color, texture, and brightness changes in the image than other features (e.g., interest points), and thus generally enables a significant reduction in the number of training examples, required to maintain high recognition accuracy. In this paper, we show that using contours as basic object features allows a sparse multi-view object representation in terms of a few shape templates, illustrated in Fig. 1. The templates are specified as 2D probabilistic maps of viewpoint-specific object shapes. They can be interpreted as “mental images” of an object category that are widely believed to play an important role in human vision [14]. While the templates are distinct, they are jointly analyzed in our inference. Given only a few of these shape templates, we show that it is possible to accurately identify boundaries and 3D pose of object occurrences amidst background clutter.

Instrumental to the proposed shape-based 3D object recognition is our new, mid-level feature, called bag of boundaries (BOB). A BOB located at a given point in the image is a histogram of boundaries, i.e., the right image contours that occur in the BOB’s neighborhood and belong to the foreground. If the object occurs, its boundaries will be “covered” by many BOBs in the image. Therefore, we represent the image and the shape templates of the object model by deformable 2D lattices of BOBs which can collectively provide a stronger support of the occurrence hypothesis than any individual contour. This allows conducting 3D object recognition by matching the image’s and template’s BOBs, instead of directly matching cluttered edges in the image and the shape templates. There are two main differ-

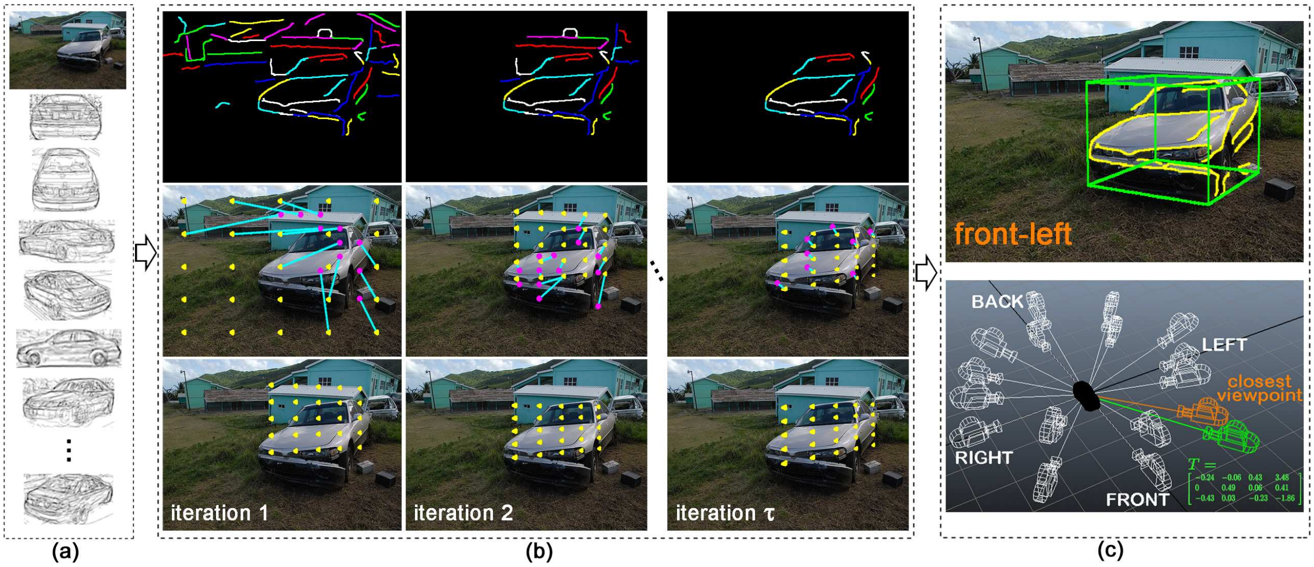

iteration 1
iteration 2
iteration τ
front-left
BACK
LEFT
closest viewpoint
RIGHT
FRONT
$T = \begin{bmatrix} -0.24 & -0.06 & 0.43 & 3.48 \\ 0 & 0.49 & 0.06 & 0.41 \\ -0.43 & 0.03 & -0.23 & -1.86 \end{bmatrix}$
(a)
(b)
(c)

Figure 1. Overview of our approach. We seek a subset of foreground image contours, referred to as boundaries, that jointly best match to the shape templates of the object model, under an arbitrary affine projection. Instead of directly matching individual contours, we match their summaries—our new, mid-level features, called bags of boundaries (BOBs). (a) A test image, and the shape templates of the category car. (b) Successive iterations of matching BOBs placed on deformable grids in the test image (magenta) and the shape templates (yellow). Top: current estimates of boundaries that best match to the shape templates. Middle: matches from the previous iteration define how to project the grids of BOBs of every shape template (yellow) onto the test image, and thus match them to the grid of BOBs in the image (magenta); the grids are deformable to accommodate invariance to non-rigid, locally affine shape deformations of object occurrences. Bottom: current estimates of the best matching shape template and its affine projection onto the test image. (c) Our results: boundary detection and 3D pose estimation of the car occurrence in the test image. The estimated viewpoint is depicted as the green camera, and the best matching shape template is shown as the orange camera. The label "front-left" is our discrete viewpoint estimate.

ences from other common mid-level features (e.g., Bag of Words, shape context). First, boundaries, which we use for computing the BOB histogram, are not observable, but hidden variables that must be inferred. The BOB histogram is computed from the right contours, not any edges (as in, e.g., BOW, shape context). Second, BOBs lie on a deformable 2D lattice, whose warping is iteratively guided top-down by the inference algorithm, such that the BOBs could better summarize boundaries for recognition.

Overview: Our approach consists of two steps, illustrated in Fig. 1. In *Step 1*, we learn the viewer-centered shape templates of an object category. We assume that training images are labeled with bounding boxes around object instances. For each training image, we estimate its corresponding 3D camera location on the viewing sphere using a standard SfM method. For each camera viewpoint, the template is learned from boundaries detected within the bounding boxes around training instances, seen from that viewpoint. After normalizing the bounding boxes to have the same size as the template, their boundaries are copied to the template, and averaged. Every pixel in the template counts the average frequency it falls on a boundary, resulting in a probabilistic shape map (see Fig. 2). In *Step 2*, we conduct shape matching between all contours in a given image, and the shape templates learned in Step 1. The matching seeks a subset of foreground image contours, i.e., boundaries, that jointly best match to the shape templates under an arbitrary affine projection (3D rotation, translation, and scale). We lift from the clutter of image edges, and realize shape matching by establishing correspondences between 2D lattices of BOBs in the image and the templates. This is formulated as an efficient convex optimization that allows for *non-rigid*, locally affine, shape deformations. The best matching BOBs identify object boundaries, and the associated affine projection of the template onto the image. The parameters of this affine projection are taken as a real-valued, continuous estimate of 3D object pose, while the best matching template identifies a discrete pose estimate.

In the following, Sec. 2 points out our contributions; Sec. 3 describes the viewer-centered shape templates; Sec. 4 specifies BOBs; Sec. 5 and Sec. 6 formulate BOB matching; and Sec. 7 presents our empirical evaluation.

2. Our Contributions and Prior work

To our knowledge, this paper presents the first shape-based approach to view-invariant object detection and pose estimation from a single image. While most prior work detects only bounding boxes around objects [16, 22, 17, 1, 10, 15, 18, 21, 13], our approach is capable of detecting boundaries that delineate objects, and their characteristic parts, seen from arbitrary viewpoints. For delineating object parts, we do not require part labels in training. The approach of [7] also seeks to delineate detected objects. However, they employ computationally expensive inference of a generative model of Gabor-filter responses only to detect sparsely placed stick-like edgelets belonging to objects. By using contours instead of point-based features, we relax the stringent requirement of prior work that objects must have significant interior texture to carry out geometric registration.

We relax the restrictive assumption of some prior work (e.g., [17]) that objects are piece-wise planar, spatially related through a homography. We allow non-rigid object deformations, and estimate the affine projection matrix.

Our approach is fundamentally viewer-centered, because we use a set of distinct object representations corresponding to different camera viewpoints. However, we do not use the two-stage inference common in prior work [21, 13], where one first reasons about objects based on each viewpoint representation independently, and then fuses these hypotheses in the second stage. Instead, our inference *jointly* considers *all* distinct object representations within a unified convex optimization framework.

Shape-based single-view object detection has a long-track record in vision (e.g., [2, 5, 23]). The key research questions explored by this line of work concern the formulation of shape representation and similarity for shape matching. Similar to the work of [2, 23], we use a lattice of mid-level shape features, called BOBs, for shape matching. Unlike shape context, a BOB is aimed at filtering the clutter of image edges, and identifying and summarizing boundaries that occur in the BOB's neighborhood. For object detection, we find the best matching subset of BOBs to our shape templates, such that the matched BOBs maximally cover all image contours that are estimated to be boundaries. This is very different from most prior work on shape (e.g., [2]) that typically works with edge fragments, and seeks evidence for their matching in relatively short-range neighborhoods. Instead, we treat each contour as a whole unit, and require a joint support from multiple BOBs to either match it to our model, or declare it as the background. This makes our shape matching more robust to background clutter.

Our shape matching simultaneously estimates a 3D affine projection of the best matching shape templates to the image. Related to ours is prior work on matching two images under 2D locally affine transformation [9], or global 2D scale and rotation transformation [8]. However, they treat descriptors of image features as fixed vectors, and do not account that they change under local or global affine transformations. By contrast, we allow non-rigid deformations of contours by estimating the optimal placement of BOBs in the image. As the BOBs change positions, they cover different sets of contours, and thus change their associated histograms, as desired.

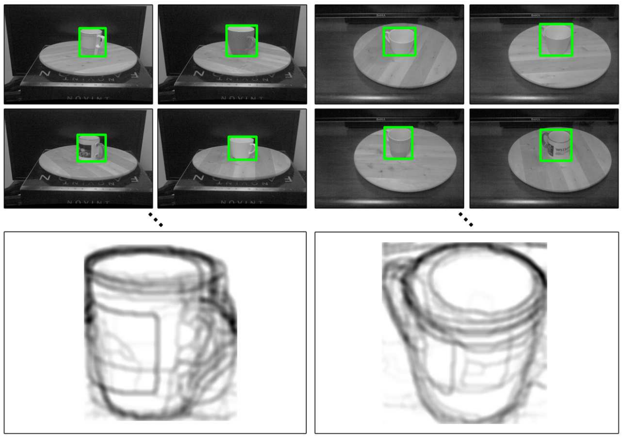


Figure 2. Example shape templates obtained for the mug category. Boundaries that fall in the bounding box of the object are averaged to form the probabilistic shape map.

3. Building the Shape Templates

This section explains how to build the shape template from a given set of training images captured from a specific camera viewpoint. We will consider that the camera viewpoint is given together with bounding boxes around objects, for clarity. In Sec. 7, we relax this assumption, and describe how to estimate camera viewpoints of training images.

In each training image, we first extract long, salient contours, using the approach of [23]. This is illustrated in Fig. 2. Then, we set the size of our template to an average rectangle of all training bounding boxes (i.e., the length of each side of the template is equal to the average length of the corresponding sides of the bounding boxes). All training bounding boxes are scaled to have the same size as the template. This allows us to directly copy all boundaries from the bounding boxes to the template. Every pixel in the template counts the average number of times it falls on a boundary. This results in a probabilistic shape map, as shown in Fig. 2. As can be seen, due to the alignment and scaling of bounding boxes, the shape template is capable of capturing prominent object boundaries. Any contours that come from background clutter or belong to rare variations of the object category, by definition, will have low probability of occurrence in the shape template.

In this way, we learn a number of shape templates corresponding to distinct viewpoints present in the training set.

4. Shape Representation

Our shape representation is designed to facilitate matching between image contours, and all shape templates of the object category. We formulate this matching as many-to-many, because, in general, specific features of a category instance, and canonical features of the category model may not be in one-to-one correspondence. Thus, our goal is to identify a subset of image contours and a subset of template parts that match. Instead of directly matching contours, we match BOBs placed on deformable grids in the image and all the templates, as illustrated in Fig. 3. The BOBs serve to jointly collect evidence on candidate boundaries, and facilitate many-to-many shape matching.

A BOB that is placed in the shape template is the standard shape context [2], computed over a relatively large spatial neighborhood. The radius of every BOB in the template is data-driven and varies in inference, as described in Sec. 5.

A BOB that is placed in the image differs from the standard shape context in that its log-polar histogram includes only those image contours occurring in its neighborhood that are estimated as boundaries. Similar to the work of [23], we avoid enumerating exponentially many choices of figure/ground labeling of contours. Rather, for a BOB located at image point i, we compute the BOB's histogram, S_i, as the following linear function of an indicator vector, X, indexed by all contours in the image, and a matrix V_i, which serves to formalize the BOB's neighborhood:

$$S_i = V_i X. \tag{1}$$

An element of X is set to 1 if the corresponding contour is estimated as foreground, or 0, otherwise. An element of V_i, denoted as $(V_i)_{st}$, counts the number of pixels of tth contour that fall in sth bin of the log-polar neighborhood of i. Note that V_i is observable. However, computing S_i requires estimation of the hidden variables X in inference.

5. Shape Matching

This section presents our inference, under non-rigid shape deformations and arbitrary 3D affine projection. We place a number of BOBs in the image and the shape templates, and match them, as illustrated in Fig. 3. The result is a subset of best matching image BOBs which are closest to the expected affine projections of the corresponding template BOBs onto the image. Also, the corresponding pairs of BOBs have the smallest differences in their associated boundary histograms. To jointly minimize these two criteria, we estimate the optimal placement of image BOBs to maximally cover the identified object boundaries, and thus account for any non-rigid shape deformations. In the following, we gradually formalize the matching of BOBs from a simple standard linear assignment problem to the desired complex optimization which allows for non-rigid shape transformations.

More formally, let $\mathcal{M}$ be the set of template BOBs, $m = |\mathcal{M}|$, and $\mathcal{I}$ be the set of image BOBs, $n = |\mathcal{I}|$. The homogenous coordinates of image and template BOBs, p_i and q_j, are represented by 3×1 and 4×1 vectors, respectively. We want to estimate an $n \times m$ matrix $F = [f_{ij}]$, whose each element f_{ij} represents confidence that 2D point $i \in \mathcal{I}$ is the best match to 3D point $j \in \mathcal{M}$.

The criterion that best matching BOBs have maximally similar boundary histograms can be formalized as

$$\begin{array}{ll} \min\limits_{F} & \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{M}} f_{ij} c_{ij} \\ \text{s.t} & \forall i,j,\ f_{ij} \geq 0,\ \ \sum_i f_{ij} = 1,\ \ \sum_j f_{ij} \leq 1, \end{array} \tag{2}$$

where the constraints on the f_{ij}'s enforce one-to-many matching, such that every BOB in the template finds its corresponding image BOB. c_{ij} is the histogram dissimilarity of BOBs i and j defined as

$$c_{ij;X} = (V_i X - S_j)^T \Sigma_j^{-1} (V_i X - S_j) \tag{3}$$

where the indicator vector of boundaries in the image $X \in \{0,1\}^n$, the BOB's neighborhood matrix V_i, and the boundary histogram S_j are defined in Sec. 4. The covariance matrix Σ_j is learned in training for each template point $j \in \mathcal{M}$. We organize dissimilarities $c_{ij;X}$ in an $n \times m$ matrix C_X. This allows expressing the objective of (2) in a more convenient matrix form: $\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{M}} f_{ij} c_{ij;X} = tr\{C_X^T F\}$.

To identify boundaries, i.e., estimate X, we extend (2) as

$$\begin{array}{ll} \min\limits_{F,X} & \text{tr}\{C_X^T F\} \\ \text{s.t} & F \geq 0,\ F^T \mathbf{1}_n = \mathbf{1}_m,\ F\mathbf{1}_m \leq \mathbf{1}_n,\ X \in [0,1]^n \end{array} \tag{4}$$

where $\mathbf{1}_n$ is n-dimensional vector with all 1's, and X is relaxed to take continuous real values in $[0,1]^n$.

The formulation in (4) has two major limitations. First, the resulting matches may contain template BOBs from all viewpoints, which would mean that the image shows all object views at once. Therefore, it is necessary to additionally constrain (4), such that a majority of correspondences are established between the image and a cluster of templates corresponding to neighboring camera locations on the viewing sphere (or, in a special case, one particular template). The best matching subset of templates can be jointly used to robustly estimate the object viewpoint, which may have not been seen previously in training. Second, (4) does not provide invariance to non-rigid shape deformations and 3D affine transformations. Both limitations could be addressed by allowing image BOBs to iteratively move to the expected affine projections of their corresponding template BOBs. This is similar to the EM algorithm. For computing the expected locations of BOBs (E-step), we maximize their matches (M-step). The iterative displacements of image BOBs are constrained to be locally similar. In this way, we enforce that image BOBs match to the shape templates with similar (neighboring) viewpoints on the viewing sphere. Below, we specify these additional constraints.

Let $\mathcal{T}$ be a set of all projection matrices, so $T \in \mathcal{T}$ has the form $T = K\,[R|t]$, where R is a 3×3 rotation matrix,

t is a 3×1 translation vector, and K captures scale and camera parameters. Given $T \in \mathcal{T}$ that projects the template onto the image, the expected location of image point p_i can be estimated as $\hat{p}_{i;T} = \sum_{j \in \mathcal{M}} f_{ij} T q_j$. After finding best correspondences $F = [f_{ij}]$, we move each p_i to its expected location $\hat{p}_{i;T}$, and then repeat matching. The criterion that neighboring BOBs should have the same displacements can be formalized as

$$\min_{T \in \mathcal{T}} \sum_{i \in \mathcal{I}} \|(\hat{p}_{i;T} - p_i) - \sum_{k \in \mathcal{I}} w_{ik} (\hat{p}_{k;T} - p_k)\|, \quad (5)$$

where $\|\cdot\|$ is ℓ_2 norm, and the w_{ik}'s are elements of the $n \times n$ adjacency matrix of image BOBs, $W = [w_{ik}]$, representing the neighbor strength between all BOB pairs, $(i,k) \in \mathcal{I} \times \mathcal{I}$. We specify w_{ik} as inversely proportional to the distance between p_i and p_k.

The objective in (5) minimizes only the magnitude of relative displacements of BOBs in the image. We also want to bound their absolute displacements as

$$\min_{T \in \mathcal{T}} \sum_{i \in \mathcal{I}} \|\hat{p}_{i;T} - p_i\|. \quad (6)$$

By introducing a $3 \times n$ matrix of image coordinates P, and a $4 \times m$ matrix of template coordinate Q, we combine the objectives of (4), (5), and (6) into our final formulation:

$$\begin{array}{ll} \min\limits_{X,F,T} & \text{tr}\left\{C_X^T F\right\} + \alpha \|TQF^T - P\| \\ & +\beta \|(TQF^T - P) - (TQF^T - P)W^T\| \\ \text{s.t} & X \in [0,1]^N;\ T \in \mathcal{T} \\ & F \geq 0;\ F^T \mathbf{1}_N = \mathbf{1}_M;\ F\mathbf{1}_M \leq \mathbf{1}_N \end{array} \quad (7)$$

Note that when $\alpha = \beta = 0$ and Σ_j is the identity matrix, (7) is equivalent to the 2D shape packing of [23]. Also, (7) is similar to recent matching formulations, presented in [9, 8]; however, they do not account that image features change under affine transformation.

6. Algorithm

This section describes our algorithm for solving (7). Input to our algorithm are the BOB coordinates Q and P, and their adjacency matrix W. We experimentally find optimal $\alpha = 2$ and $\beta = 1$. We use an iterative approach to find F, X and T in (7), and use the software CVX http://cvxr.com/cvx/ to compute the optimization. Each iteration consists of the following steps.

(1) We fix X and T and compute F. Initially, all image contours are considered, so X is set to $\mathbf{1}_n$. T is initially set to the orthogonal projection matrix $[1\,0\,0\,0; 0\,1\,0\,0; 0\,0\,1\,0]$.

(2) We fix X and F and compute T. We linearize the quadratic constraint on the rotation matrix R by relaxing the orthogonality constraint $RR^T = I$ to the norm constraint $\|R\|_\infty \leq 1$. This can be done without affecting the original optimization problem (see [12] for details). $\|R\|_\infty$ is the spectral norm of R. After T is determined, we transform the image BOBs p_i to their expected locations $\hat{p}_{i;T} = \sum_{j \in \mathcal{M}} f_{ij} T q_j$.

(3) We fix T and F, and compute X, and C_X.

(4) Steps (1)–(3) are iterated. After convergence, i.e., when F, T and X no longer change, we remove the boundaries indicated by X from the initial set of image contours.

(5) To detect multiple object occurrences in the image, steps (1)–(4) are repeated until the set of image contours reduces to the 10% of its initial size.

Implementation. On average, we extract around 80 contours in each image. Our Matlab CVX implementation of the above steps (1)–(5) takes about 3min on a 2.66GHz, 3.49GB RAM PC.

7. Results

Datasets. We evaluate our approach on the 3D object dataset [16] and the Table Top dataset of [18]. The 3D object dataset is used for evaluating on classes cars and bikes; whereas the Table Top dataset is used for evaluating on classes staplers, mugs and computer mice. In both datasets, each class contains 10 object instances. The first 5 are selected for training, and the remaining 5 for testing, as in [16, 7, 18]. In the 3D object dataset, each instance is observed under 8 angles ($A_1..A_8$), 2 heights (H_1, H_2), and 3 scales ($S_1..S_3$), i.e. 48 images. For training, we use only the images from scale S_1. For testing, we use all $5 \times 48 = 240$ images per category. In the Table Top dataset, each instance is observed under 8 angles ($A_1..A_8$), 2 heights (H_1, H_2), and one scale (S_1), i.e. 16 images. For cars, we also evaluate our method on the PASCAL VOC 2006 dataset, and on the car show dataset [13]. The PASCAL dataset contains 544 test images. The car show dataset contains 20 sequences of cars as they rotate by 360 degrees. Similar to [13], we use the last 10 sequences for testing, a total of 1120 images. Additionally, we evaluate our method on the mug category of the ETHZ Shape dataset [5]. It contains 48 positive images with mugs, and 207 negative images with a mixture of apple logos, bottles, giraffes and swans.

Training. Each training image is labeled with the object's bounding box. We use two approaches to identify the camera viewpoint of each training image. For the two object categories cars and bikes, we use publicly available, AUTO CAD, synthetic models, as in [11, 10, 7]. For the other object categories studied in this paper synthetic models are not available, and, therefore, we estimate camera viewpoints via standard SfM methods, as in [1]. Then, for each training image, we extract long, salient contours using [23], and build 16 shape templates (8 angles and 2 heights). For each template, we sample 25 BOBs on a uniform 5x5 grid, so we

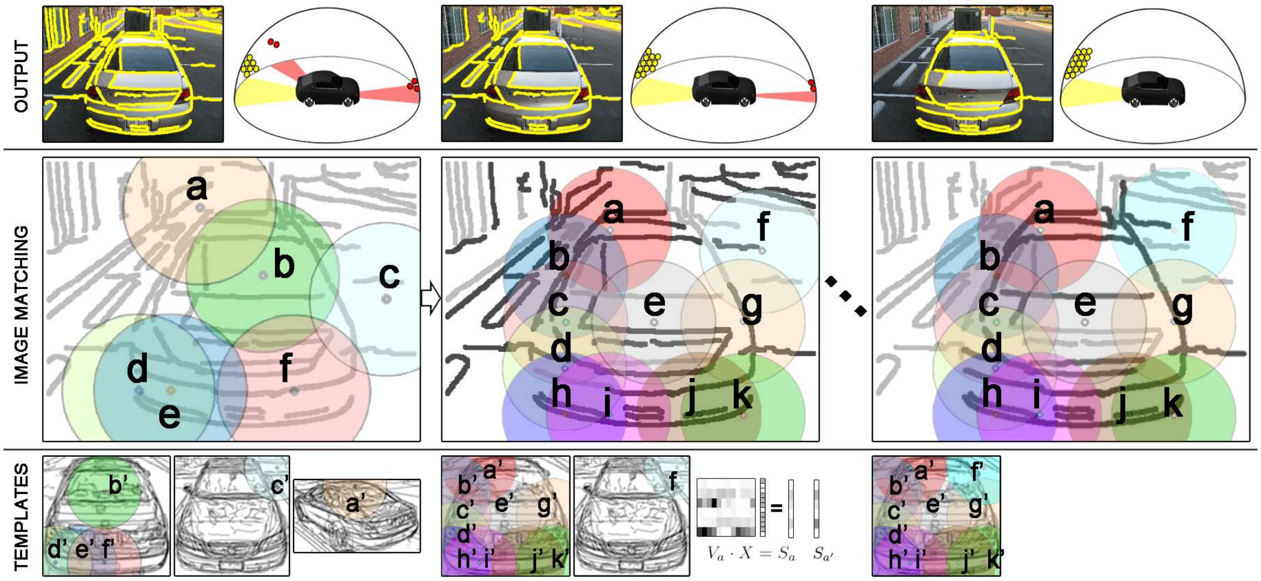


Figure 3. Iterative matching of BOBs in the image and the shape templates. Top: Estimated boundaries, and 3D BOBs that are matched to the image BOBs. Middle and Bottom: Initially, the matches are established with multiple templates. After a few iterations, the matching identifies the correct shape template, and the image contours selected as foreground indeed fall on the object. Corresponding pairs of image and template BOBs are marked with the same color.

sample a total of 400 BOBs $q_j \in \mathcal{M}$. A shape context descriptor S_j is associated with each q_j, with a radius equal to $\frac{3}{10}$ of the object size. This way, each descriptor represents a significant part of the object, and there is a large overlap between adjacent descriptors, see Fig. 3. Using the camera pose of each viewpoint, we can compute the 3D location of each BOB q_j.

Testing. For each test image, we extract contours by the approach of [23]. We sample 121 BOBs p_i on a uniform 11x11 grid (empirically found optimal), and compute a shape context descriptor for every point p_i. Initially, the right scale for the descriptor is unknown. We try multiple BOB radii, proportional to the image size, i.e. $radius = \gamma \frac{w+h}{2}$, with $\gamma \in \{0.05, 0.1, 0.15, 0.2\}$. We run one iteration and keep the solution (F, T, X) that returns the best score for the objective function in (7). In further iterations, the projection matrix T gives us an estimate of the scale of the object. The radius of the descriptor is then set to $\frac{3}{10}$ the size of the estimated object, to match the BOB defined in the templates. This constitutes our default setup.

Evaluation criteria. To evaluate the 2D detection, we use the standard PASCAL VOC detection quality criterion. For a correct localization, the overlap a_o between predicted bounding box B_p and ground truth bounding box B_{gt} must exceed 50%, as defined by $a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$. Our 2D localization is created by fitting a bounding box to the contours that are selected by our algorithm. Since our method outputs contours, and not just a bounding box, we can compute precision and recall of our detector in terms of contour pixels, which is more precise. We count as true positives the detected contour pixels that intersect with the object's mask. The contours extracted originally form the total set of true positives and true negatives.

In addition to the 2D localization, the proposed approach yields an estimate of the object's 3D pose. For viewpoint classification, we take the viewpoint-label of the best matching template, whose camera centroid is the closest (Euclidean distance) to the estimated camera.

Evaluating our training setup. To determine how many viewpoints are necessary to represent an object category, we train (a) 4, (b) 8, (c) 16, and (d) 32 shape templates for the car category of the 3D object dataset [16]. The selected viewpoints are (a) front, left, back and right, height H_1, scale S_1, (b) all 8 viewpoints, height H_1, scale S_1, (c) all 8 viewpoints, heights H_1, H_2, scale S_1, and (d) all 8 viewpoints, heights H_1, H_2, scales S_1, S_2. We test each setup on the task of 8-viewpoint classification, and report the average classification performance in Tab. 1.

Number of templates	4	8	16	32
Average performance	64.5% ± 1.5%	78.9% ± 0.7%	85.4% ± 0.6%	86.1% ± 0.5%

Table 1. 3D object car dataset. Influence of the number of templates on the pose estimation performance.

As expected, the performance improves as we add more templates. We choose to use 16 templates and not 32, because the small performance gain does not justify the large increase in computation time. Our formulation is linear but the solver takes much longer to handle large matrices, which makes it impractical to use 32 templates on large datasets.

Precision of pose estimation. To evaluate the precision of our camera pose estimate, we use synthetic data, for which it is easy to obtain ground truth. We collect 6 synthetic car models from free CAD databases, e.g. turbosquid.com. Cameras are positioned at azimuth $a = 0..360^\circ$ in 10° steps, elevation $e = 0..40^\circ$ in 20° steps, and distance $d = 3, 5, 7$ (generic units). 324 images are rendered for each 3D model, for a total of 1944 test images (see the supplemental material). For each image, we run our car detector and record the 3D location of the estimated camera. The position error is defined as the Euclidean distance between the centroids of ground truth camera and estimated camera. We measure an average error of 3.1 ± 1.2 units. There is a large variation because when we incorrectly estimate the camera, it is oftentimes because we have mis-interpreted a viewpoint for its symmetric, e.g. front for back. The position error is also due to the underestimation of the distance between object and camera, which is probably caused by our choice of resolving the scale via the camera pose.

Independent viewpoints. We test a setup where one considers each viewpoint independently. We solve 16 independent optimizations, as defined by (7), for each of the 16 shape templates. The image receives the viewpoint-label of the template that yields the best score. We here get a drop in classification performance by $5.3\% \pm 0.4\%$ compared to our default setup.

Qualitative results. Fig. 4 shows examples of successful detections and 3D object pose estimation. We successfully detect the object boundaries, and we correctly estimate the 3D poses. We are also able to identify intermediate poses that are not available in our discrete set of shape templates, e.g. the car in the lower-right image.

Quantitative results. We first evaluate our performance on object detection. Fig. 5 shows the precision/recall of our detector on the PASCAL cars and the car show dataset. We outperform the existing methods of [17, 11, 7, 13]. Our approach allows for non-rigid deformations and estimates a full affine projection matrix, which explain our superior results. Our method can also detect object parts. We count 425 wheels in the car images of the 3D object dataset, and record precision/recall at equal error rate (EER) of $63.7\% \pm 0.5\%$ for the wheel parts. Also, for the contour pixels detection, we measure precision/recall at EER of $68.3\% \pm 0.2\%$ on the 3D object car dataset. After the deadline of camera-ready submissions, we became aware of competitive detection results, presented in [6] – specifically, they reported an ROC curve that saturates at about 60% recall.

On the ETHZ Shape dataset, we use 24 positive mug images and 24 negative images from the other classes to estimate the equal error rate detection threshold t_{eer}. We run our mug detector on the remaining 207 images. Each candidate detection with an objective score below t_{eer} is classified as mug. The precision and recall at equal error rate is measured at $84.3\% \pm 0.5\%$, which is better than the 59% reported in [23]. This also suggests that our shape templates generalize well to other datasets.

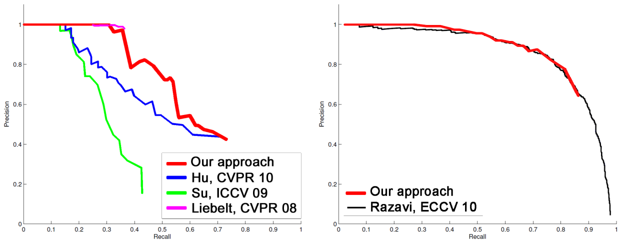


Figure 5. Our detection results on the PASCAL VOC 2006 car dataset (left) and the car show dataset (right).

Fig. 6 shows our confusion matrices for viewpoint classification, and compares our performance to that of [1, 19, 10, 18, 7] on different datasets. Our approach shows superior performance for nearly all viewpoints and categories relative to these approaches. After the deadline of camera-ready submissions, we became aware of the state-of-the-art results of viewpoint classification, presented in [6] – specifically, they reported the viewpoint classification accuracy of 92.8% for cars, and 96.8% for bicycles. For the mice and staplers of the 3D object dataset, we achieve a viewpoint classification of $78.2\% \pm 1.1\%$, resp. $77.6\% \pm 0.9\%$, and improve by 3.2%, resp. 4.1% the results of [18].

8. Conclusion

We have presented a novel, shape-based approach to 3D pose estimation and view-invariant object detection. Shape, being one of the most categorical object features, has allowed us to formulate a new, sparse, view-centered object representation in terms of a few, distinct, probabilistic, shape templates. The templates are analogues to the well-known "mental images", believed to play an important role in human vision. We have formulated 3D object recognition as matching image contours to the set of shape templates. To address the background clutter, we have lifted shape matching from considering individual contours to matching of new, mid-level features, called bags of boundaries (BOBs). BOBs are histograms of the right contours estimated to belong to the foreground. In inference, BOBs in the image are iteratively re-located to jointly best summarize object boundaries and match them to the shape templates, while accounting for likely non-rigid shape deformations. Our experiments have demonstrated that BOBs are rich contextual features that facilitate view-invariant inference, yielding favorable performance relative to the state of the art on benchmark datasets.

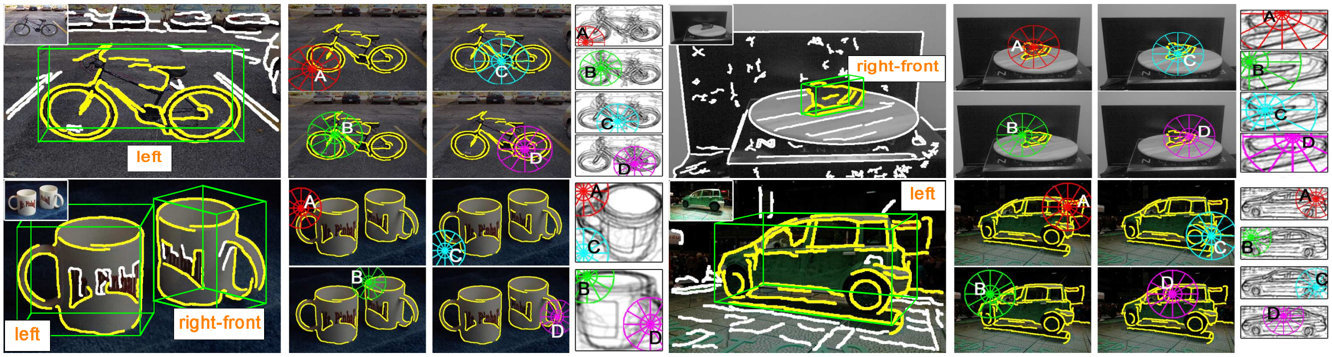
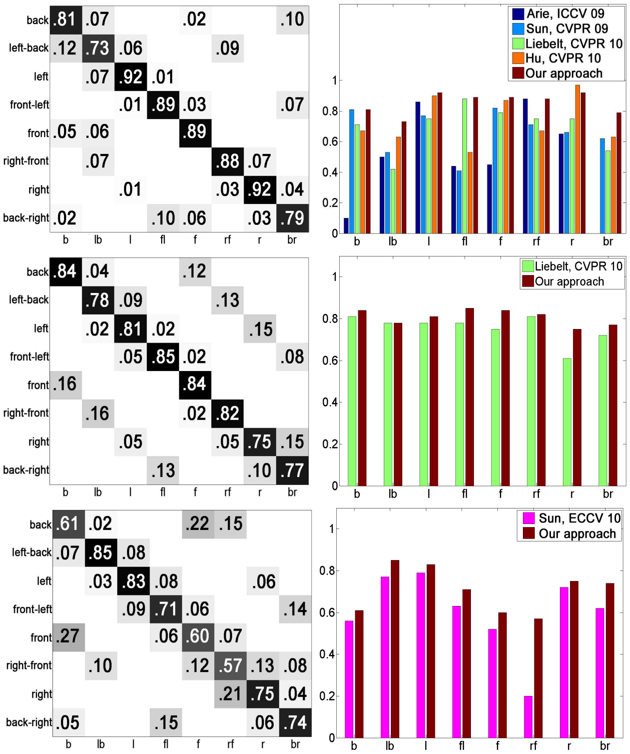


Figure 4. Examples of our contour detection and 3D object pose estimation. Upper left: 3D object car dataset. Upper right: Table Top stapler dataset. Lower left: ETHZ dataset. Lower right: car show dataset. We are successfully detecting the contours of the objects, and we correctly estimate their 3D pose. The viewpoint label of the best matching template is taken as a discrete estimate of object pose, e.g. right-front for the stapler. Example matches between image and shape templates BOBs are also shown. (Best viewed in color.)



	b	lb	l	fl	f	rf	r	br
back	.81	.07			.02			.10
left-back	.12	.73	.06			.09		
left		.07	.92	.01				
front-left			.01	.89	.03			.07
front	.05	.06			.89			
right-front		.07				.88	.07	
right			.01			.03	.92	.04
back-right	.02			.10	.06		.03	.79

	b	lb	l	fl	f	rf	r	br
back	.84	.04			.12			
left-back		.78	.09			.13		
left		.02	.81	.02			.15	
front-left			.05	.85	.02			.08
front	.16				.84			
right-front		.16			.02	.82		
right			.05			.05	.75	.15
back-right				.13			.10	.77

	b	lb	l	fl	f	rf	r	br
back	.61	.02			.22	.15		
left-back	.07	.85	.08					
left		.03	.83	.08			.06	
front-left			.09	.71	.06			.14
front	.27			.06	.60	.07		
right-front		.10			.12	.57	.13	.08
right						.21	.75	.04
back-right	.05			.15			.06	.74

Figure 6. Viewpoint classification results. Top: cars in the 3D object dataset. Middle: bikes in the 3D object dataset. Bottom: mice-staplers-mugs in the Table Top dataset. Left: confusion matrices. Right: diagonal elements of our confusion matrices are compared with the state of the art.

References

[1] M. Arie-Nachimson and R. Basri. Constructing implicit 3D shape models for pose estimation. In *ICCV*, 2009.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24(4):509–522, 2002.

[3] I. Biederman. Surface versus edge-based determinants of visual recognition. *Cog. Psych.*, 20(1):38–64, 1988.

[4] S. J. Dickinson, R. Bergevin, I. Biederman, J.-O. Eklundh, R. Munck-Fairwood, A. K. Jain, and A. Pentland. Panel report: the potential of geons for generic 3-D object recognition. *Image and Vision Computing*, 15(4):277–292, 1997.

[5] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *ECCV*, 2006.

[6] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010.

[7] W. Hu and S.-C. Zhu. Learning a probabilistic model mixing 3D and 2D primitives for view invariant object recognition. In *CVPR*, 2010.

[8] H. Jiang and S. X. Yu. Linear solution to scale and rotation invariant object matching. In *CVPR*, 2009.

[9] H. Li, E. Kim, X. Huang, and L. He. Object matching with a locally affine-invariant constraint. In *CVPR*, 2010.

[10] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR*, 2010.

[11] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *CVPR*, 2008.

[12] A. Nemirovski. Sums of random symmetric matrices and quadratic optimization under orthogonality constraints. In *Mathematical Programming*, 2007.

[13] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[14] Z. W. Pylyshyn. Mental imagery: In search of a theory. *Behavioral and Brain Sciences*, 25(2):157–182, 2002.

[15] N. Razavi, J. Gall, and L. V. Gool. Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections. In *ECCV*, 2010.

[16] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[17] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.

[18] M. Sun, G. Bradski, B. Xu, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, 2010.

[19] M. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view probabilistic model for 3d object classes. In *CVPR*, 2009.

[20] M. J. Tarr and I. Gauthier. Do viewpoint-dependent mechanisms generalize across members of a class? *Cognition*, 67(1-2):73–110, 1998.

[21] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[22] P. Yan, S. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007.

[23] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, 2008.