Article

The International Journal of Robotics Research
2015, Vol. 34(4-5) 627–652

DOI: 10.1177/0278364914558493
ijr.sagepub.com
SAGE

A Gestaltist approach to contour-based object recognition: Combining bottom-up and top-down cues

Ching L. Teo, Cornelia Fermüller and Yiannis Aloimonos

Abstract

This paper proposes a method for detecting generic classes of objects from their representative contours that can be used by a robot with vision to find objects in cluttered environments. The approach uses a mid-level image operator to group edges into contours which likely correspond to object boundaries. This mid-level operator is used in two ways, bottom-up on simple edges and top-down incorporating object shape information, thus acting as the intermediary between low-level and high-level information. First, the mid-level operator, called the image torque, is applied to simple edges to extract likely fixation locations of objects. Using the operator's output, a novel contour-based descriptor is created that extends the shape context descriptor to include boundary ownership information and accounts for rotation. This descriptor is then used in a multi-scale matching approach to modulate the torque operator towards the target, so it indicates its location and size. Unlike other approaches that use edges directly to guide the independent edge grouping and matching processes for recognition, both of these steps are effectively combined using the proposed method. We evaluate the performance of our approach using four diverse datasets containing a variety of object categories in clutter, occlusion and viewpoint changes. Compared with current state-of-the-art approaches, our approach is able to detect the target with fewer false alarms in most object categories. The performance is further improved when we exploit depth information available from the Kinect RGB-Depth sensor by imposing depth consistency when applying the image torque.

Keywords

Object class detection, contour grouping, local contour features, shape matching, mid-level vision

1. Introduction

Humans have an uncanny ability to recognize objects of various shapes and sizes with relative speed and ease even in highly cluttered environments by exploiting a wide variety of visual cues. In this work we seek to use contours as the main cue for recognition.

The problem of object recognition in general, and recognition from contours specifically, still is considered a challenging problem. The problem is particularly difficult in clutter, when objects occlude each other, and only parts of an object's boundary are visible. How do we get from the simple edge responses detected by filters to characteristic contours at the boundaries of objects? What is the approach we should take in our computations? Is there inspiration we can get from human perception?

The Gestalt theorists proposed a very influential theory on how this can be resolved. They suggested that certain principles guide the processing in the vision system with the goal of extracting foreground regions from background ones. Here we focus on two of these principles: the *principle of closure*, which states that simple feature elements tend to be grouped together if they are parts of a closed figure, and the *principle of past experience*, implying that visual stimuli are categorized according to past experience. We propose a computational mechanism, a mid-level vision operator, to implement these principles. This mid-level operator groups edges within regions of different sizes to locate boundaries of objects, and it interacts with low-level and high-level processes. By using it first in a bottom-up fashion to group simple edge responses, it can be used to find in parallel potential object locations. Then by tuning it to object characteristic edges to group boundary edges of objects, even when only parts of the object are

University of Maryland, Department of Computer Science, College Park, MD, USA

Corresponding author:
Ching L. Teo, University of Maryland, Department of Computer Science, College Park, MD 20742, USA.
Email: cteo@cs.umd.edu

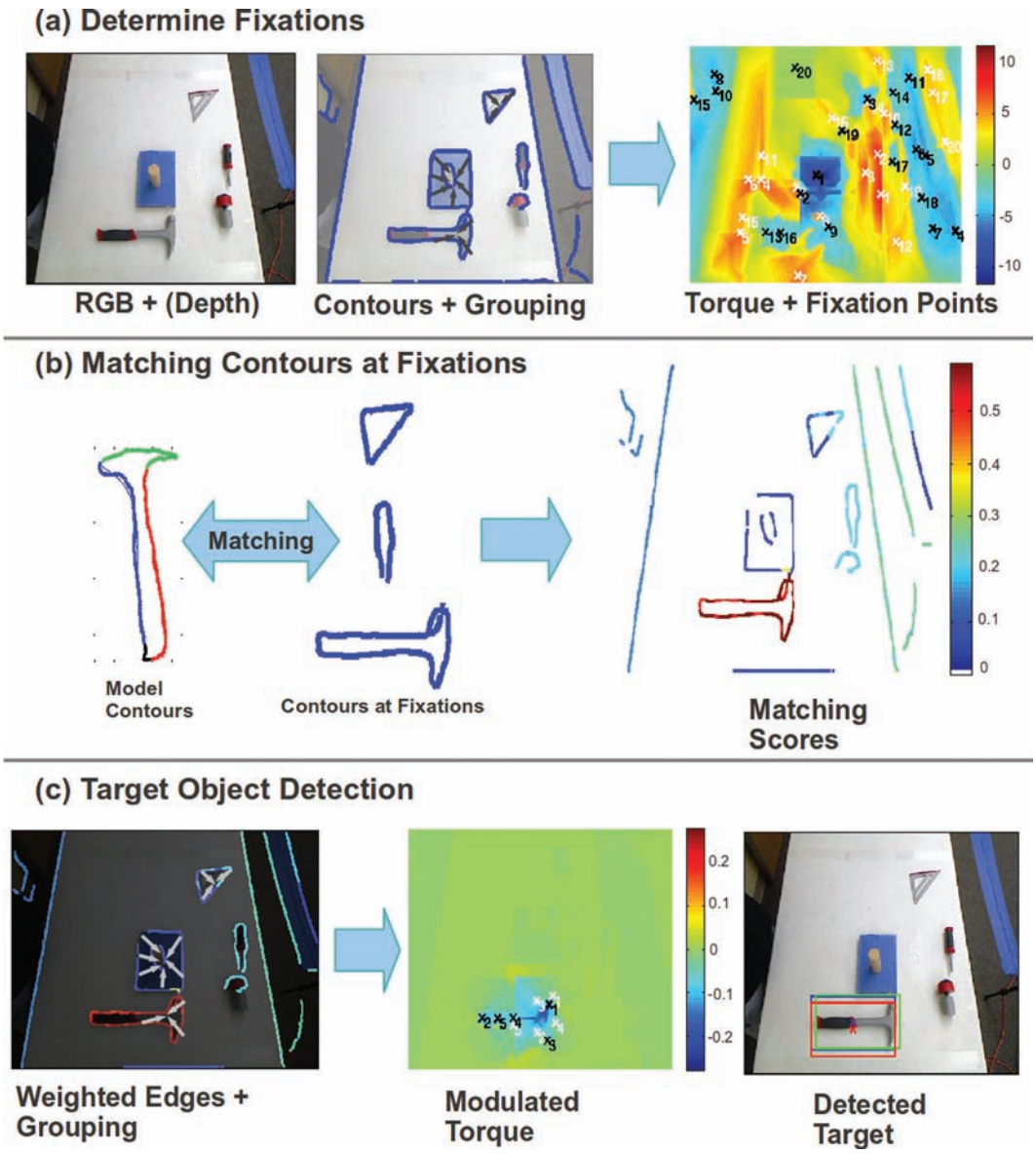


Fig. 1. From mid-level contour grouping to object recognition. (a) Attention-based contour grouping: by grouping contours that support the presence of an object, a set of initial fixation points are used for the recognition step. (b) Contour-based recognition at fixation points: using the supporting contours at each fixation point, we score the contour similarity in a hierarchical manner (increasing lengths) with a target contour model. (c) Target object detection: regrouping scored contours using the same mid-level grouping strategy reveals locations, scales and supporting contours of the target object.

visible, it can be used to locate and identify specific objects. We next describe the approach in a nutshell.

Motivated by the speed and ease of several biologically inspired robotic visual perception systems that use attention as a basis to reduce the visual search space for object detection and scene understanding (Navalpakkam and Itti, 2002; Frintrop, 2006; Yu et al., 2009), we have developed a mid-level contour grouping mechanism that first determines fixation points corresponding to potential object locations (Figure 1(a)). Each fixation point corresponds to a set of (almost) closed contours that are suggestive of object-like boundaries. We then perform recognition via scoring the contours in a hierarchical manner of increasing lengths with contours belonging to the target. In this step, a crucial issue is the choice of the representation for the contours. We extend the popular shape context descriptor of Belongie et al. (2002) by using the additional fixation information to create a descriptor that discriminates between ambiguous edge fragments by matching partial contour pieces. By applying the Fourier transform over the angular components of the descriptor at the fixation center, we are able to handle changes in scale and rotation (Figure 1(b)), in addition to changes in translation. The scores of the contours are then used as weights in the same mid-level grouping strategy to determine the locations, scales and supporting contours of the target, by grouping contours that are both

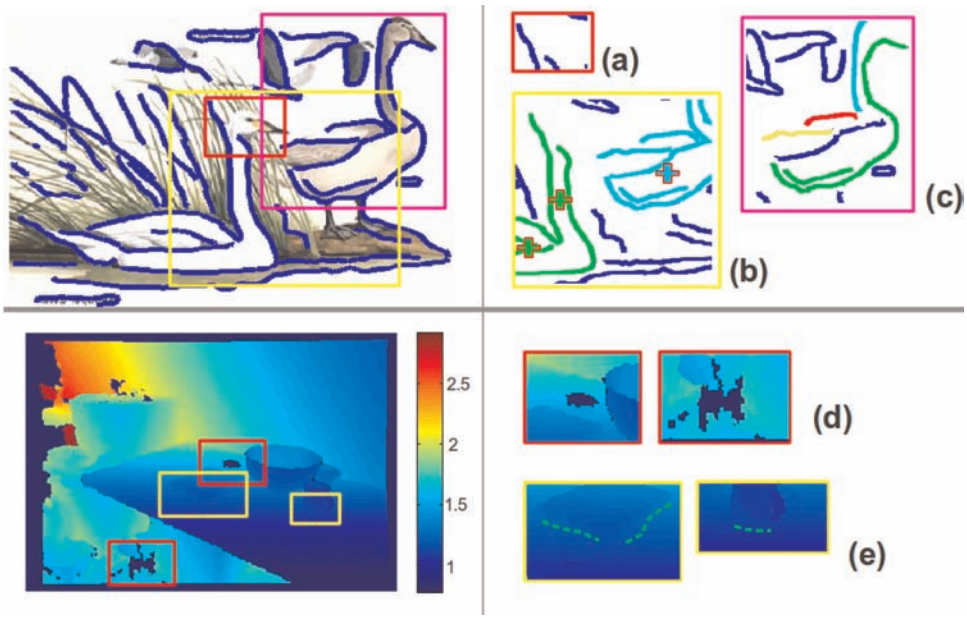


Fig. 2. Illustration of the challenges of contour-based categorical object recognition. (Top panel) 2D images. (a) Noisy edges: some edges on the head are missing. (b) Boundary ownership between two targets, with support marked as '+'. (c) Detecting partial contours in clutter. (Bottom panel) 2.5D images. (d) Noise and errors in depth/surface estimates, shown as dark blue gaps, make grouping edges at such regions difficult. (e) Edges at smoothly varying depth boundaries (dotted green lines) are hard to localize.

similar and share a common fate for object-like contours (Figure 1(c)).

The main advantages of using contour information for recognition in robotic vision are that they are: 1) extremely easy to obtain and process using recent state-of-the-art edge detectors (Dollár and Zitnick, 2013; Lim et al., 2013) and 2) robust against changes in lighting in comparison with other appearance- or pixel-based cues (e.g. color and texture) since one considers at least the first-order differences between low-level pixel signals in localizing the edge (Martin et al., 2004). In addition, since we are interested in recognizing *categories* of similarly shaped objects, by using contours we generalize better across object categories which share certain common shapes and functionality in different domains. This has important implications when the robot is tasked with searching for objects based on descriptions of shape (this work) or functionality, or when it is asked to suggest plausible alternatives when the actual target is not present. The main drawback of only using 2D-contour-based information is that it is affected by changes in viewpoints, which we address through our choice of a robust shape-based descriptor. The result is a simple and straightforward approach that enables robots to quickly recognize objects that share common 2D shape properties in cluttered environments.

The input is a 2D RGB image or a 2.5D image (RGB with depth information), and we are interested in the detection of the contours that correspond to the target object class: for example a `Hammer` class in the UMD Hand-Manipulation dataset or a `Bottle` class in the ETHZ-Shapes dataset (Section 4), which is defined by a specific outline (or shape) of the most representative contours of the object. The key challenge is to determine, from the edges derived from the input image, the set of contours that supports the presence of the target object. Although this task seems simple and straightforward, it poses several crucial challenges (Figure 2):

1. Inaccurate and noisy (broken) edges. Since edge detection in 2D or 2.5D images inherently depends on the local intensity gradients or surface normals, noise during the image formation process would inadvertently result in edges that are either inaccurate, incomplete or missing (Figure 2(a), (d)). Additionally, for 2.5D images, at the junctions of smoothly varying depth, boundaries cannot be accurately localized since their surface normals are ambiguous (Figure 2(e)). One common way of resolving this issue is to first attempt to group pieces of contours, using saliency measures and Gestalt principles of edge continuation, as in for example Kennedy et al. (2011) and Ming et al. (2012). Edge grouping techniques, however, will still fail when considerable clutter (see issue (3) below) occurs and when broken edges predominate.
2. Boundary detection and ownership. In order to distinguish between contours belonging to one object, a key challenge is to determine who 'owns' the edge. Once the ownership is determined, we can assign an orientation to the contour (Figure 2(b)), which makes it more discriminative. The problem is that to determine the ownership of a boundary we first need to detect the object (or at least the presence of an object): a chicken-and-egg problem.

3. Partial matching in clutter. Related to issue (1) above, occlusions from clutter and self-occlusions from the object's internal contours both produce contours that are broken and fragmented in the image (Figure 2(c)). Unfortunately, since in such situations we detect nearby contours that do not originate from the same physical entity, bottom-up edge grouping techniques will still fail. To overcome this, approaches such as those by Riemenschneider et al. (2010) and Ma and Latecki (2011) perform *partial* edge matching. The main limitation of such approaches is that even with good partial matches, a separate edge grouping and scoring step is still needed to determine the location of the object.

Indeed, the main reason for these challenges is that detecting and recognizing objects from edges alone is a very difficult task. This is because an edge, when it is used in *isolation*, does not convey a lot of discriminatory information. Compounded with the issues raised above, contour-based recognition of objects is therefore extremely challenging. In this work, we argue that by exploiting *mid-level* contour grouping mechanisms, we are able to effectively address all the above issues in a simple, holistic object-detection framework. The key insight is that by treating contours over larger scales, and combining them intelligently over longer contours while using depth information when it is available to enforce depth consistency, we are able to mitigate some of the limitations of using edges for object recognition, resulting in much better performance in difficult scenarios containing clutter and noise.

In particular, our contributions are threefold: 1) we demonstrate the usefulness of a recently introduced mid-level contour grouping operator, termed the *image torque* (Nishigaki et al., 2012; Xu et al., 2012), that computes a measure of contour completion. We describe an extension that exploits depth information from the robot's RGB-Depth camera to enforce depth consistency across the grouped contours (Section 3.1). 2) By integrating information derived from torque with the shape context descriptor, we introduce a novel descriptor that encodes boundary ownership information leading to better matching of partial edge fragments (Section 3.2.1). To make the descriptor invariant to rotation, the Fourier transform is applied over the angular bins of the descriptor so that rotation can be estimated prior to matching (Section 3.2.2). 3) Finally, using the matched contours (Section 3.2.3), we modulate the torque operator so that it becomes sensitive to the target object model, thereby introducing high-level object information into the grouping mechanism (Section 3.3).

2. Related work

The problem of contour-based object recognition has been studied extensively within the computer vision community. Existing approaches can be classified based on how the edges/contours are obtained, represented and scored, and on the basis of the algorithms used for classification.

Some approaches (Shotton et al., 2005; Opelt et al., 2006) learn a codebook of shape fragments. The learned class-specific shape fragments are then matched using oriented chamfer matching and voted via a star-shape model to detect objects in the image. More recently, Mairal et al. (2008) proposed a discriminative sparse coding that learns a class-specific dictionary that detects object-specific contours within clutter. Leibe et al. (2004) introduced the notion of an 'implicit shape model' where patches relative to an object center are used to create a codebook that encodes both spatial and appearance-based information for a particular class of objects.

Other methods transform the contour representations so that they become more amenable for classification. Ferrari et al. (2006) approximate them with straight adjacent fragments for part-based matching. Similarly, Ravishankar et al. (2008) use curves instead of straight lines, which are more discriminative, together with a novel scoring function. More recently, Wang et al. (2012) proposed a deformable 'fan-shape' object model that statistically encodes the expected deformation (scale and angle) of matched contour fragments with respect to an assigned center. A score for the object's location is determined via a Hough distance voting metric over several scales.

Many approaches have used local feature descriptors from interest points to match contours with the target. Leordeanu et al. (2007) use simple features based on orientations and pairwise interactions to create a local descriptor for matching. Srinivasan et al. (2010) view the problem as a many-to-one matching problem and used shape context to match long salient contours. Descriptors tuned for matching partial shape fragments were introduced by Riemenschneider et al. (2010) and used in a discriminative framework by Kontschieder et al. (2011). Toshev et al. (2010) proposed using a novel descriptor known as the 'chordiogram' to encode relative angles of boundaries obtained from an initial super-pixel segmentation step. In Lu et al. (2009), the authors used a triplet of edge points to create a histogram of angles over all triplets for representing and matching similar contours. Machine learning methods have also been employed to improve the matching function. Maji and Malik (2009) viewed the problem as a deformable shape matching problem where a max-margin learning approach was used to assign discriminative weights to potential contours while Ommer and Malik (2009) used a kernel-based support vector machine (SVM) (Cortes and Vapnik, 1995) with a Hough voting approach to detect object-specific contours. The work of Hinterstoisser et al. (2012) introduced a very fast 2D line matching technique by precomputing binarized gradient orientations of the model template known as LINE2D. By spreading the orientations of the model orientations over a small region, the approach is shown to be robust to changes in orientations within clutter. However, the method requires a large number of templates for precomputing the response

and is memory-intensive. As LINE2D does not explicitly handle occlusions, Hsiao and Hebert (2012) extended LINE2D's performance with the addition of occlusion priors learned from training data. The occlusion prior is obtained from the statistics of which object parts are likely to be occluded. The prior is estimated from the geometry of the object, occluder and camera. This yields a probabilistic occlusion prior that indicates which image points in LINE2D are consistent with the unoccluded object for matching with the model. Although the approach showed improved detection results under severe clutter, it requires significant amounts of annotated data to learn the occlusion prior, and it is unclear how the approach performs over different clutter interactions without retraining.

There are some works that focused on improving the robustness of shape-based descriptors against a variety of deformations. Since we are interested in detecting manipulated objects (for the UMD Hand-Manipulation dataset), rotational invariance is crucial. Jiang and Yu (2009) proposed searching over all possible rotations and selecting the one that yields the smallest matching score with the shape context descriptor. Extending the idea of searching over pose space, Lian and Zhang (2010) proposed using a fan-shaped triangulation technique with a novel optimization scheme to improve the rotational invariance of shape context. Instead of searching over rotations, Yang and Wang (2007) applied a 2D Fourier transform to contour points represented as Euclidean distances with respect to a manually selected center point, to create a descriptor that is invariant to translations, scaling and rotations.

The approach presented here is most related to our prior work in Teo et al. (2013) where we combined the torque mid-level operator with high-level information of a *specific* object (of known size and shape) represented by silhouettes obtained from 2.5D Kinect data from various poses. There are also many other works that have used the full 2.5D information for object recognition. See Han et al. (2013) for an extensive review. Approaches that use local 2.5D descriptors, such as fast point feature histograms (FPFH) (Rusu et al., 2009) or (histogram of oriented normal vectors (HONV) Tang et al., 2013), exploit local geometry and surface normals as their main features. To improve their discriminatory power, Bo et al. (2011) used hierarchical kernel descriptors to produce larger patch-based features and trained a linear SVM for 2.5D object recognition. A more recent extension (Bo et al., 2012) proposed a discriminatory dictionary learning method termed 'hierarchical matching pursuit' (HMP), to learn, in an unsupervised manner, hierarchical feature representations of image patches containing RGB-Depth data. Using a trained SVM, the approach achieves state-of-the-art recognition results over the RGB-D object dataset introduced in Bo et al. (2011).

These different approaches share several common characteristics. First, in order to overcome the noise and clutter that exist in real edge maps, some form of edge grouping is applied. Next, using specific local descriptors, edges that are grouped together are matched to see if they are similar enough to the target object model. However, in the approaches surveyed above, the two steps of grouping and matching are performed *independently* of each other, and their performance can depend on the effectiveness of either step. In addition, many of them do not address the issue of boundary ownership at all, which is a powerful cue for contour discrimination. Even among approaches that use object centers, either explicitly (Leibe et al., 2004) or implicitly (Toshev et al., 2010) to determine ownership, a key drawback is that the object centers are determined either by hand or from imprecise over-segmentation using superpixels. Our proposed approach, by way of contrast, uses the image torque operator in a holistic manner such that grouped edges are intrinsically endowed with boundary ownership information via their torque centers, to create a more robust descriptor for matching contour fragments with the target object model. As we will detail in the next section, because objects are represented in terms of *partial* contours with descriptors that enable us to estimate the amount of rotation with respect to the model, we are able to circumvent the need for an extensive pose search and allow for more complex shape representations compared to our prior work. Our proposed mid-level object recognition approach therefore provides robotic applications with a method that: 1) is effective under a wide variety of imaging conditions, 2) requires minimal training since only sample contours of the target shape are required and 3) generalizes well to similar-shaped objects (no retraining needed).

3. Approach

The proposed approach consists of several steps and is summarized in Figure 3. Prior to detection, contours of the target model are obtained from annotated ground truth of the training set (Figure 3(a)). As the ground truth consists of contours of varying sizes and scales, we first apply generalized Procrustes analysis (Gower, 1975) to align the contours of the objects of the same category. Next, motivated by the classical work on representing contours compactly using *codons* (Richards and Hoffman, 1985), we take a similar approach of breaking up the contours at locations of minimum and maximum curvature, where a codon is a set of ordered contiguous edge pixels in the image. Each codon from the training set is then represented as a set of B-splines and we apply expectation-maximization (EM) clustering over the spline coefficients to recover the set of u model codons, $\{b_1,\ldots,b_u\}$, which are arranged clockwise in the order they appear on the contour. For matching codons over multiple scales, we group these model codons, creating *longer* codons by combining neighboring codons in a cumulative way such that we create a set of l model codons of increasing length, $\mathcal{C}_{mo} = \{\mathcal{M}_1, \ldots, \mathcal{M}_l\}$, with $|\mathcal{M}_t| < |\mathcal{M}_{t+1}|$ until the entire contour is accounted for, per target object class.

At the detection step, we first obtain from the input image an edge map I_e of size $H \times W$ (height $\times$ width)

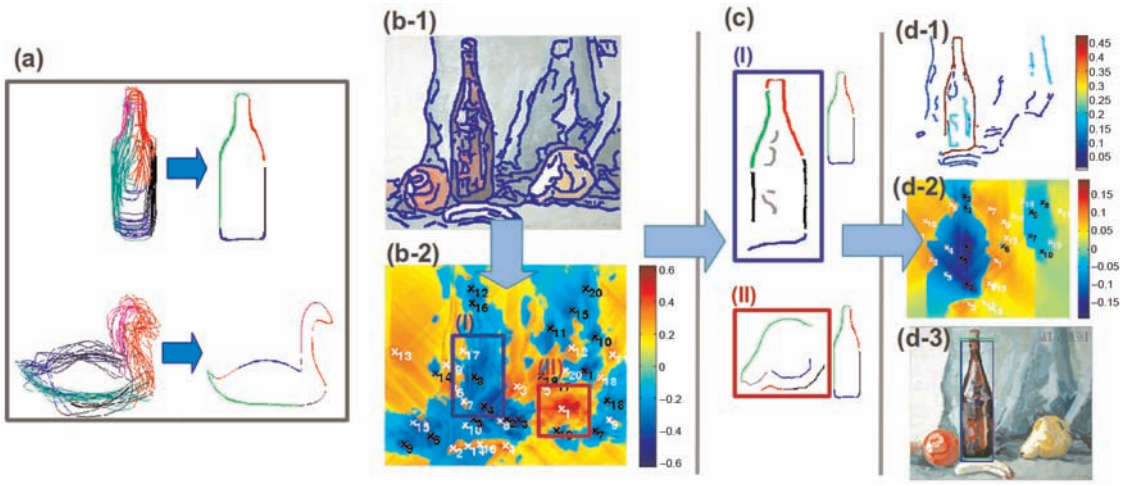


Fig. 3. Overview of proposed approach. (a) Example model contours fragments (codons) obtained via EM clustering of annotated training data: `Bottles` (top) and `Swans` (below). (b-1) Input image + edge map. (b-2) Original torque value map with detected proto-objects centers P_c: sorted by their torque values, black crosses (negative torque), white crosses (positive torque). (c) Multi-scale edge matching at two selected centers (I) and (II) compared to target `Bottles`. The codons selected have the strongest torque contribution $\tau_{p_c q_i}$. Matches to the model at one scale are shown in the same color; gray indicates no matches. (d-1) Weighted edge map (red means higher weights), (d-2) modulated torque value map and (d-3) predicted object location and scale at maximum torque. See text for details.

using any standard edge detection technique (Figure 3(b-1)). We detail the remaining steps in the sections that follow (Figure 3(c), (d)). First, we review the image torque and how it functions as an edge grouping mechanism to locate the contours of *proto-objects*: regions likely to contain objects in the image. Next, we show how information from the computed torque can be used to enhance the shape context descriptor with boundary ownership information and rotational invariance for robust matching. Finally, we describe how these matched contours are used in modulating the image torque operator in a multi-scale manner so that class-specific object contours can be extracted for recognition.

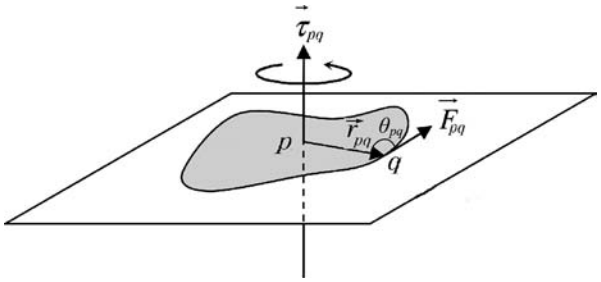


Fig. 4. Definition of the image torque: $\vec{r}_{pq}$ is the vector from the center pixel p to an edge pixel q. $\vec{F}_q$ is the tangent vector at q and θ_{pq} is the angle between $\vec{r}_{pq}$ and $\vec{F}_q$. Reproduced with kind permission from the IEEE (Nishigaki et al., 2012).

3.1 Contour completion using image torque

The image torque, introduced in Nishigaki et al. (2012), is a mid-level operator that is tuned to find closed boundaries, which are indicative of the presence of possible objects (proto-objects). Given an image edge map I_e, consider an image patch $P \in I_e$ with center point p. We denote the set of edge pixels (pixels corresponding to edges in P) as $E(P)$. This measure of edge completion is computed by summing the cross-product between the tangent vectors at an edge pixel $q \in E(P)$ and the corresponding displacement vectors between p and q as shown in Figure 4. Formally, the value of image torque[1] τ_{pq} of an edge pixel q within a discrete image patch with center p is defined as

$$\tau_{pq} = \vec{r}_{pq} \times \vec{F}_q \tag{1}$$

where $\vec{r}_{pq}$ is the displacement vector from p to q and $\vec{F}_q$ is the tangent vector[2] at q. In the original torque implementation $\vec{F}_q$ is a unit vector. $\vec{F}_q$ can be viewed as a 'force' unit vector in the image space that can be associated with the relative importance of a particular edge pixel (see (4)). The torque of an image *patch*, P, is defined as the sum of the torque values of all edge pixels, $E(P)$, within the patch as follows:

$$\tau_P = \frac{1}{2|P|} \sum_{q \in E(P)} \tau_{pq} \tag{2}$$

We compute (2) over multiple scales, $s \in \mathcal{S}$ for every image point, and we extract the largest τ_P over all scales to create a 2D torque *value map*, T_I (Figure 3(b-2)), with the same dimensions as I_e. The extrema in the value map indicate locations in the image that are likely *centers* of closed contours (crosses in Figure 3(b-2)), denoted as $\mathcal{P}_c$, and we consider the largest τ_P, which we evaluate as possible

proto-object centers. We use the top 20 largest τ_P in our current implementation. For each extremum center, $p_c \in \mathcal{P}_c$, we can also compute the torque *contribution* per edge pixel, $\tau_{p_c q_i}$, via (1). Setting a threshold t_c on the torque contribution, we obtain a set of n edge pixels (with $\tau_{p_c q_i} > t_c$) which we denote as $\mathcal{Q}_{p_c} = \{q_i\}, i \in \{1, \ldots, n\}$ (shown as selected contours in Figure 3(c)).

We highlight two important properties of the operator that make it ideal for grouping edges that support the presence of proto-objects. Firstly, the summation operation in (2) strongly biases the operator against edge pixels that have different orientations within the image patch P. This means that randomly oriented edges from noise or textures have a smaller torque contribution to τ_P compared to edges that have orientations that are more coherent towards forming a closed contour. Secondly, the cross-product between $\vec{r}$ and $\vec{F}$ will be large if an edge pixel is far away from the center p, implying that the patch size associated with an extremum point is a good estimate of the object's *scale*.

For 2.5D images, we modify the definition of the image torque above so that depth information is incorporated. The key idea is to add in an additional depth constraint so that contours with the same depth values as the torque centers are preferred over contours with different depth values. This way, we enforce some form of depth consistency within the torque contour grouping framework when depth information is available. Formally, from equation (1), we apply additional weights w_{pq} that measure the absolute difference in depth values between an edge point q and the center p:

$$\tau_{d_{pq}} = \vec{r}_{pq} \times (w_{pq} \vec{F}_q) \tag{3}$$

with $w_{pq} = \text{abs}(I_d(p) - I_d(q))$ where I_d is a $W \times H$ depth image that records the depth values per image pixel and abs $(\cdot)$ denotes the absolute value. The torque of an image patch with depth information is similarly derived via $\tau_{d_{pq}}$ using equation (2).

In practice, we use an efficient implementation[3] via the method of summed area tables (Crow, 1984) (integral images) to compute the image torque per patch in constant time. To achieve further efficiency, we use a discrete set of angles to represent the edge vectors (we used eight in our current implementation). We precompute and sum up the image torque per edge pixel into a summed table per angle. Summing up the responses over all discrete angles enables us to compute τ_{pq} efficiently. For $\tau_{d_{pq}}$ that includes depth information, we create at the same time a set of summed tables for w_{pq} per displacement angle which affords us the same constant time computation complexity as the non-depth torque τ_{pq}.

The original image torque (Nishigaki et al., 2012), however, is a purely bottom-up procedure: it detects potential proto-object locations, p_c, and supporting contours, $\mathcal{Q}_{p_c}$, with no preference for any particular object class. In the next two sections we show that by integrating this bottom-up information from torque with the shape context local descriptor, we extend the operator so that it becomes sensitive to a target object class.

3.2. Torque shape context descriptor

Let us return to (1), which defines the image torque, τ_{pq}, between an edge pixel q and the associated center pixel p. Since $\vec{r}_{pq}$ is fixed (edges are fixed in a 2D image), one way to modify τ_{pq} is to change the weight on $\vec{F}_q$ as follows:

$$\tau^{\omega}_{pq} = \vec{r}_{pq} \times \vec{f}(\vec{F}_q) \tag{4}$$

where $\vec{f}(\cdot)$ can be any vector-valued function that modifies the tangent unit vector $\vec{F}_q$ appropriately. In this work, we define $\vec{f}(\cdot)$ to be a normalized contour matching score function that is larger if edge pixel q is similar to the target object's contours and smaller otherwise. We detail in the sections that follow how the final form of τ^{ω}_{pq} in (12) is derived that tunes the torque mid-level operator towards the target object class for detection and recognition.

There are numerous methods for matching local edge pixels, among which the most popular is the shape context descriptor (Belongie et al., 2002). Given a set of edge pixels, $\mathcal{Q}_{p_c} = \{q_1, \ldots, q_n\}$, for each point q_i the shape context descriptor, h^{sc}_i, is defined as a coarse histogram of the relative coordinates of the remaining $n - 1$ points:

$$h^{sc}_i(k) = \#\left[q_j \neq q_i : (q_j - q_i) \in \text{bin}(k)\right], j \neq i \tag{5}$$

In the above equation, $(q_j - q_i)$ denotes the coordinate difference between q_j and q_i in log-polar space and bin (k) denotes the kth bin in the histogram in log-polar space centered over the ith edge point, q_i. This descriptor is tolerant to small localized deformations (due to the histogramming of the distances), and is scale- and translation-invariant.

However, when the descriptor is applied on contour *fragments*, $\mathcal{Q}'_{p_c} \subseteq \mathcal{Q}_{p_c}$, by breaking them up into codons there will be some ambiguous edge fragments that can be matched to object contour fragments of different target object classes. The reason is that the shape context in its original form does not encode any mid-level information on how the fragments are related to the object that it is supposed to support (Figure 5, middle-r1). In addition, shape context by construction is not rotationally invariant as the log-polar histograms are defined over a fixed coordinate system. Thus we need to account for target objects that present themselves in a variety of poses (Figure 5, middle-r2).

To overcome these two shortcomings, we introduce two enhancements to the shape context descriptor by 1) embedding *boundary ownership* information through image torque to create a more robust descriptor, termed the *torque* shape context (Figure 5, right), that can better match contour fragments (Section 3.2.1) and 2) as a pre-processing step, we estimate the amount of rotation between the test and model by computing the cross-correlation of the

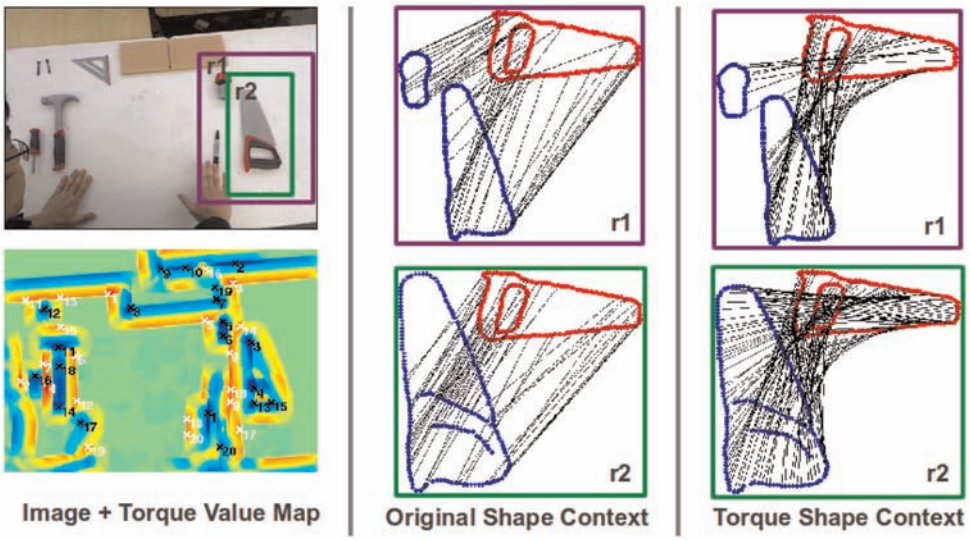


Fig. 5. Why shape context is insufficient for matching contour fragments in clutter. Left panel: input image with torque value map. The two torque fixations considered are boxed as r1 and r2 and the model is `Saw`. Model points are red and test points are blue. Black lines indicate correspondences. Middle panel: original shape context matchings. r1: wrong matches due to similar histograms; notice that the `Borer` object is matched to the handle of the `Saw` model; r2: wrong matches of test `Saw` points as shape context is not rotationally invariant. Right panel: robust matching using torque shape context. r1: fewer points from `Borer` object are matched due to boundary ownership embedding; r2: rotational invariance enables matching of rotated `Saw` to the model.

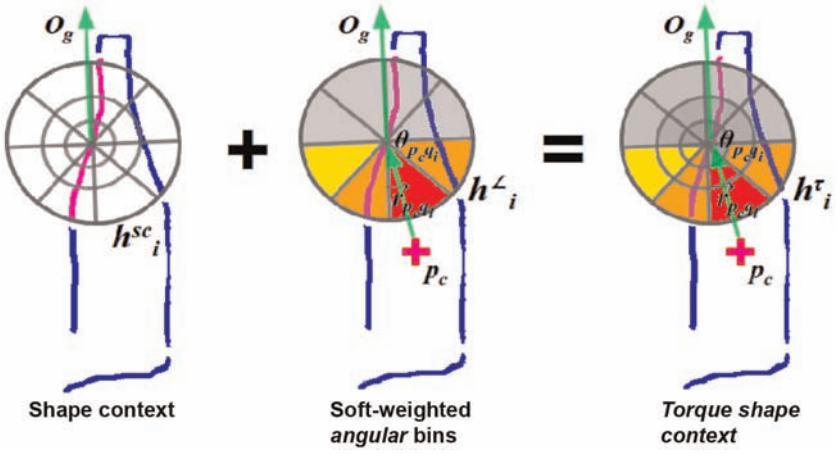


Fig. 6. Constructing the torque shape context: selected codon highlighted with respective p_c, $\vec{r}_{p_c q_i}$ and $\theta_{p_c q_i}$ from torque. h_i^{τ} is constructed by adding soft-weighted counts from angular bins in $h_i^{\angle}$ that are intersected with $\vec{r}_{p_c q_i}$ (oriented along O_g; see Figure 10). Red means more counts, gray means no counts. The sum of the original shape context h_i^{sc} bin counts with $h_i^{\angle}$ produces h_i^{τ}.

descriptor's angular bins via fast Fourier transform (FFT) (Section 3.2.2). Finally, we show how the torque shape context descriptor is matched efficiently via dynamic programming in Section 3.2.3.

3.2.1. Robust contour fragment matching from boundary ownership information. To improve the matching of contour fragments, we introduce in this work a new descriptor that extends shape context by embedding within the angular bins of the shape context histogram additional information that indicates the *location* of p_c, that is, the torque center that this fragment is supporting. Formally, consider a shape context histogram $h_i^{sc}(k)$ for edge point $q_i \in \mathcal{Q}_{p_c}$ with corresponding torque center p_c. We define the *torque* shape context histogram, $h_i^{\tau}(k)$, as the sum of the original shape context $h_i^{sc}(k)$ bins and 'soft-weighted' angular bins, $h_i^{\angle}(k)$, that are aligned towards p_c (Figure 6):

$$\begin{aligned} h_i^{\tau}(k) &= h_i^{sc}(k) + h_i^{\angle}(k) \\ &= h_i^{sc}(k) + \mathcal{K}(\angle\text{bin}(k) \equiv \theta_{p_c q_i}) \end{aligned} \tag{6}$$

where $\angle$bin (k) denotes the *angular* bins in the shape context histogram in $h_i^{sc}(k)$. $\theta_{p_c q_i}$ is the angle that vector $\vec{r}_{p_c q_i}$ makes with respect to O_g within the coordinate system of

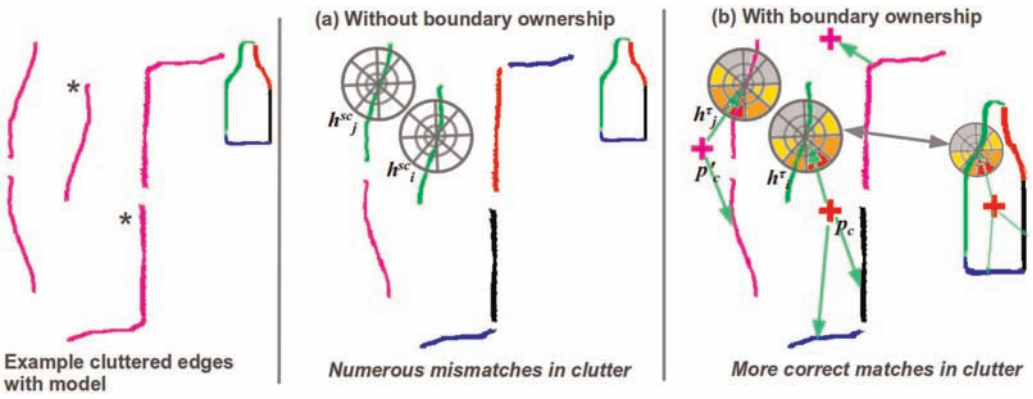


Fig. 7. Using boundary ownership information for robust matching in clutter. Left: codons in clutter to be matched with the model `Bottle`. The two codons marked with * are correct. (a) Using only shape context, many mismatches occur because of similar histograms in clutter, for example h_i^{sc} and h_j^{sc}. Codon colors code for corresponding matches with the model codons. (b) Using the boundary ownership information embedded in torque shape context, many mismatches are avoided since their histograms h_i^{τ}, h_j^{τ} and corresponding torque centers, p_c and p'_c, are more discriminative.

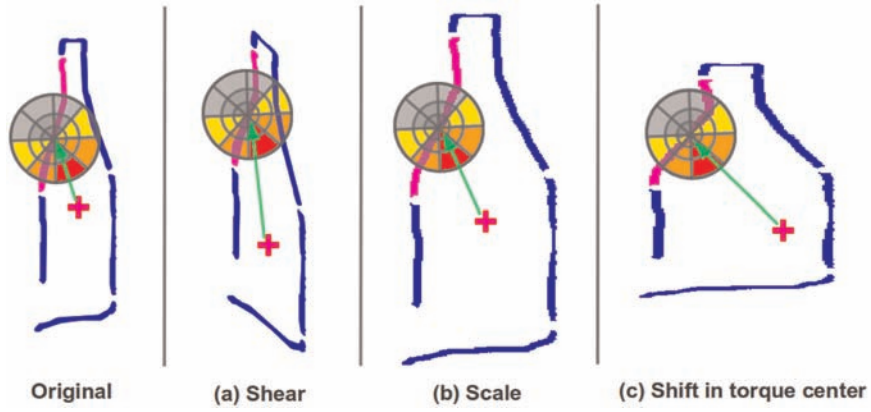


Fig. 8. Robustness against deformations. The selected torque shape context remains stable for deformations induced by (a) shearing, (b) scale changes and (c) shifts in the torque center.

the shape context, as shown in Figure 6. $\mathcal{K}(\cdot)$ is a normalized 'truncated' Gaussian $N(\theta_{p_c q_i}, \sigma_{\mathcal{K}}^2)\big|_{\theta_{p_c q_i}-\pi/2}^{\theta_{p_c q_i}+\pi/2}$ that reweighs bin counts only on the side of the shape context histogram pointing towards p_c and has zero influence on the other side.[4] What $\mathcal{K}(\cdot)$ does is to weigh angular bins in $h_i(k)$ nearest to $\theta_{p_c q_i}$ more than those angular bins that are not aligned towards $\theta_{p_c q_i}$. This truncated 'soft weighting' of angular bins in $h_i^{\tau}(k)$ entails two important properties that are key for matching contour fragments in clutter:

1. As $\mathcal{K}(\cdot)$ is active only on the side facing p_c, the set of torque shape contexts $\{h_i^{\tau} | q_i \in \mathcal{Q}_{p_c}\}$ effectively encodes the 'ownership' side of the set of edges in $\mathcal{Q}_{p_c}$ with respect to the torque center p_c. This makes matching contour fragments $\mathcal{Q}'_{p_c}$ with a target model much more discriminative in clutter since similarly shaped fragments (with similar $h_i^{sc}(k)$) must have the *same* p_c as support with the model for a strong match to occur. For example, Figure 7 illustrates the case where random fragments that have the same $h_i^{sc}(k)$ (due to noise in the histogram counts or nearby edges) can be differentiated using additional information from p_c.
2. By weighing the bin counts softly via $\mathcal{K}(\cdot)$, the matching of contour fragments is also *robust* to a certain amount of perturbation and deformation of the overall shape that the fragment belongs to. This is important, since the target model must match fragments from a variety of camera viewpoints. This also motivates why only *angular* bins are used, since (relative) angles are less likely to change under various image deformations (up to an affine transformation) (Figure 8).

We illustrate the advantages of using $h_i^{\tau}(k)$ using real cluttered data in Figure 9 where it enables us to 1) to distinguish between ambiguous contour fragments with similar shape contexts but different p_c and 2) perform partial contour matching under occlusion.

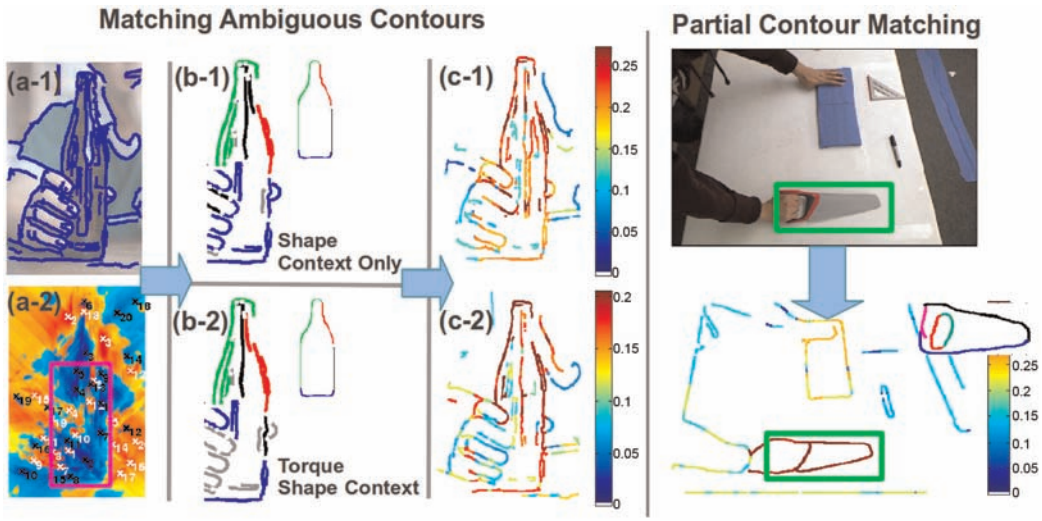


Fig. 9. Left: matching in clutter using torque shape context. (a-1) Input image + edge map. (a-2) Selected proto-object center boxed. (b) Comparing the matches to the model codons with (b-1) shape context and (b-2) torque shape context. Notice that fingers and noisy edges do not have the correct support, and are not matched in (b-2). (c) Final modulated edge weights: (c-2) with torque shape context identifies more of the correct edges than (c-1). Right: partial contour matching. The saw's handle (boxed) is occluded by the hand (top), but the blade is detected correctly (below).

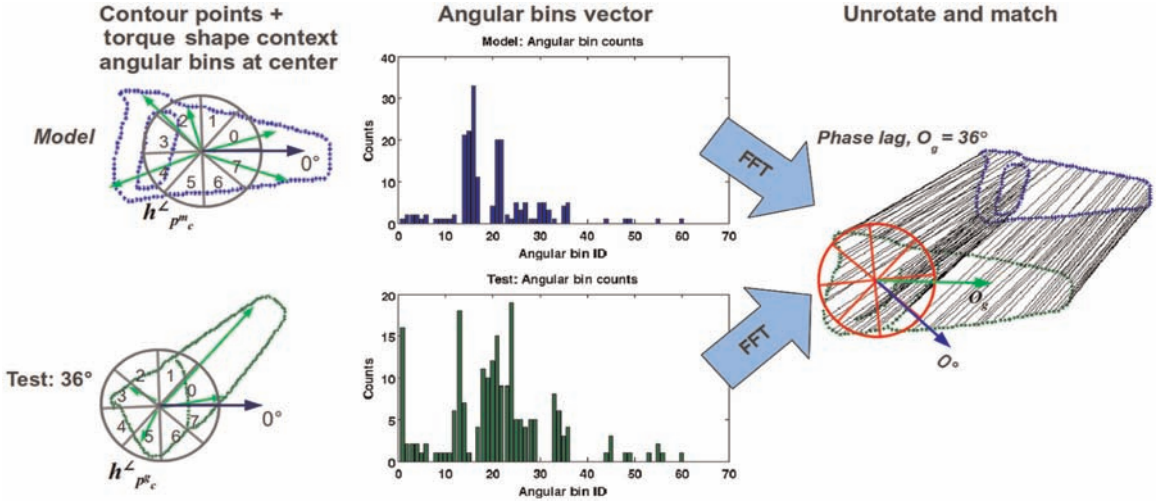


Fig. 10. Estimating the phase lag O_g from the angular bins of the torque shape context at the torque center. Left + middle: using the angular bins vectors (numbers indicate the bin ID) from the model and the test edges, we estimate the phase lag O_g from the FFT of the two signals. Right: using O_g, we 'unrotate' the test edges before matching the descriptors (black lines indicate correspondences).

3.2.2. Rotational invariance via the FFT. For rotational invariance, the most straightforward approach is to simply define the reference frame to be the tangent vector $\vec{F}_q$ at each edge point q. However, this approach in practice tends to significantly reduce the discriminatory power of the descriptor due to the fact that tangents are easily corrupted by noise and discretization effects. Instead, we propose to compute an additional torque shape context descriptor *centered* at the torque center of the test and model, p_c^g and p_c^m, that estimates the amount of rotation between them so that we can 'unrotate' the test contours before matching them with the model contours (Figure 10).

The key idea is to apply a 1D FFT over a 1D vector, $\vec{a}_g = \langle h^{\angle}_{p_c^g}(1), \ldots, h^{\angle}_{p_c^g}(\kappa)\rangle$, derived from the angular bin counts of a torque shape context located at the torque center, $h^{\angle}_{p_c^g}(k)$, with bin 0 equivalent to the first component of the vector used and so on until all bins are accounted for (we used $\kappa = 60$ bins for this part to get more resolution). This vector succinctly captures the structure of the edge points while it is robust against changes in scale and translation. We obtain in a similar fashion $\vec{a}_m$, from $h^{\angle}_{p_c^m}(k)$ of the model. The cross-correlation between the two discrete signals, $(\vec{a}_g \bigstar \vec{a}_m)[\nu]$, is then obtained via FFT to determine the most

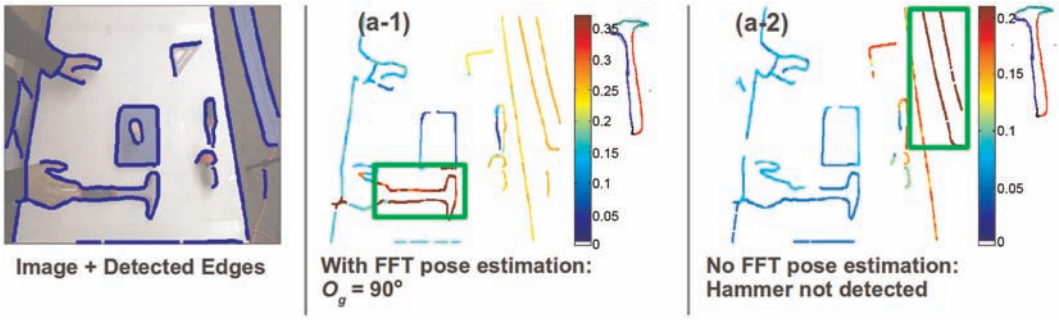


Fig. 11. Effects of using FFT to estimate O_g on matching accuracy: (a-1) with pose estimation; (a-2) without pose estimation. The hammer is much better localized and scored using the torque shape context when O_g is applied.

significant phase lag $O_g = \text{argmax}_\nu(\vec{a}_g \bigstar \vec{a}_m)[\nu]$ which is an estimate of the rotation that exists between the test and model edges. We then use O_g to 'unrotate' the test contours before matching them with the model.

Since the two signals $\vec{a}_g, \vec{a}_m$ are (potentially) circularly shifted versions of each other, there are four possible orientations (at each quadrant) that relate the test contours to the model, and we consider all four orientations when we perform multiscale contour matching (Section 3.3). Compared to matching over a large number of orientations, this approach drastically reduces the number of orientation poses to search to just four. We demonstrate the effects of imposing rotational invariance in Figure 11. One can see that without imposing O_g, the object `Hammer` is not as well detected compared to the case where O_g is used to define the reference frame. We demonstrate in Section 4.1 quantitative results that highlight the importance of this procedure over a challenging hand manipulation dataset in improving the recognition of tools that are often occluded and placed at random orientations.

3.2.3. Matching of torque shape context descriptors. Following Belongie et al. (2002), we compare the torque shape context defined in equation (6) using the χ^2 statistic. We use the dynamic programming method of Thayananthan et al. (2003) to compute correspondences ϕ by minimizing the overall cost of matching, C_ϕ, between two edge fragments $\mathcal{G}'_{p_c^g}$ (test) and $\mathcal{M}_{p_c^m}$ (model):

$$C_\phi(\mathcal{G}', \mathcal{M}) = \gamma_{sc} C_{sc}(\mathcal{G}', \mathcal{M}) + \gamma_\angle C_\angle(\mathcal{G}', \mathcal{M}) \qquad (7)$$

where we drop the subscripts p_c^g and p_c^m for simplicity in notation. $C_{sc}(\cdot)$ and $C_\angle(\cdot)$ are the shape context matching costs for the original shape context (first term in equation (6)) and the angular bin histograms (second term in equation (6)) respectively. We impose $\gamma_{sc} + \gamma_\angle = 1$ so that we control the relative importance of these two histograms in influencing the local matching score within the torque shape context. We denote for simplicity the shape context and angular components of the torque shape context histogram for the ith test point and corresponding $\phi(i)$ model point as g_i^t and $m_{\phi(i)}^t$, with $t \in \{sc, \angle\}$ respectively. The

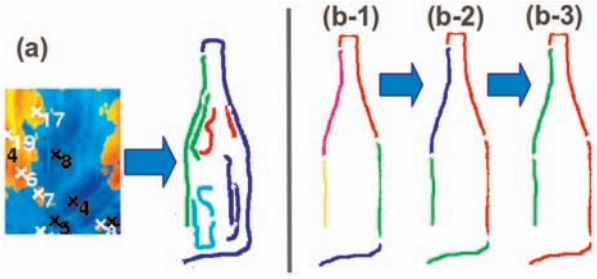


Fig. 12. Multi-scale edge matching: (a) detail of p_8 from Figure 3(b-2). Neighboring torques p_c with their supporting edges (in similar colors) are combined. (b-1) to (b-3) show increasing scales of combining neighboring codons together for matching.

matching costs $C_t(\cdot)$, $t \in \{sc, \angle\}$ for these two components are similarly defined as

$$C_t(\mathcal{G}', \mathcal{M}) = \sum_{i=1}^{n'} \chi^2(g_i^t, m_{\phi(i)}^t) \qquad (8)$$

where we sum up the χ^2 distances computed between the t components of the test points' torque shape contexts g_i^t and their n' corresponding shape contexts $m_{\phi(i)}^t$ in the model. χ^2 is defined for two sets of shape context histograms centered at $(g_i^t, m_{\phi(i)}^t)$ as

$$\chi^2(g_i^t, m_{\phi(i)}^t) = \frac{1}{2}\sum_{k=1}^{K} \frac{[h_i^t(k) - h_{\phi(i)}^t(k)]^2}{h_i^t(k) + h_{\phi(i)}^t(k)} \qquad (9)$$

Using the correspondences, we define the torque shape context matching *distance*, D_{sc}^τ, as the weighted mean of shape context matching costs over the n' matched points in $\mathcal{G}'$:

$$D_{sc}^\tau(\mathcal{G}', \mathcal{M}) = \frac{1}{n'}\sum_{i=1}^{n'} C_{\phi(i)}(\mathcal{G}', \mathcal{M}) \qquad (10)$$

Since D_{sc}^τ is a local measure of similarity of partial edge fragments, we show in Section 3.3 how we use it in a multi-scale approach to develop a mid-level contour matching

score function $\vec{f}(\cdot)$ that is sensitive to the target object class.

3.2.4. Object-sensitive torque via multi-scale matching of supporting contours. Although the matching of edge fragments enables us to detect possible partial contours that indicate the presence of the target object, it is only a weak indicator, and one needs to check if there also is sufficient support from neighboring fragments to strengthen the hypothesis. Motivated by this observation, we pursue the following multi-scale approach of progressively combining and matching neighboring edge fragments aided by torque as shown in Figure 12. From the torque grouped edges $\mathcal{Q}_{p_c}$, we first combine neighboring $\mathcal{Q}_{r_c}$ belonging to nearby centers that fall within the detected bounding box of p_c to form a larger set of grouped edges $\mathcal{R}_{N_c}$, where N_c is a new object center estimated from the center of gravity of all the contributing neighbors' proto-object centers. This combination of neighboring torques is crucial for target object classes (e.g. `Giraffes`) that have long and thin structures, and can only be represented via multiple torque centers.

Next, we group edge pixels r_i in $\mathcal{R}_{N_c} = \{r_1, \ldots, r_f\}$ into codon fragments so as to obtain a more compact representation of a set of d codons, $\mathcal{C}_g = \{\mathcal{R}'_1, \ldots, \mathcal{R}'_d\}$. Starting at codon $\mathcal{R}'_1$, we progressively select and combine the next $\mathcal{J}$ neighboring codons $\{\mathcal{R}'_{\{1\}}, \ldots, \mathcal{R}'_{\{1+\mathcal{J}\}}\}$ for comparison[5] with each of the l codons from the model contours $\mathcal{C}_{mo} = \{\mathcal{M}_1, \ldots, \mathcal{M}_l\}$ by computing $D^{\tau}_{sc}(\mathcal{R}'_{\{1\},\ldots,\{1+\mathcal{J}\}}, \mathcal{M}_{\{1\},\ldots,\{l\}})$ from (10) with a slight abuse of notation. This results in a $W \times H \times \mathcal{J} \times l$ matrix of distance scores corresponding to each combination. This process is repeated for each of the d codons which gives us a final $W \times H \times (d \times \mathcal{J} \times l)$ matrix that records the value of D^{τ}_{sc} at every edge pixel location in $\mathcal{R}_{N_c}$. We then select the smallest D^{τ}_{sc} across all $d \times \mathcal{J} \times l$ levels to yield the final distance score for each r_i denoted as a 2D torque shape context distance map, $E_{D^{\tau}_{sc}}$. This is repeated over all four possible global orientations O_g described in Section 3.2.2 and we select the orientation that yields the smallest $E_{D^{\tau}_{sc}}$. A note on the computational complexity of this step. Since d and l are small (typically, 15 and 6) and we set $\mathcal{J}$ to a small number as well (3 to 5 depending on the object class), we are able to reasonably compare all combinations of codons over several scales by a direct brute-force approach. This is an important advantage of using the compact codon representation (a mid-level representation by itself). In comparison, other methods performing partial edge matching (Riemenschneider et al., 2010; Ma and Latecki, 2011) use all edge pixels at once.

In order to convert the distance scores for each r_i in $E_{D^{\tau}_{sc}}$ to a normalized weight we use an exponential function

$$W_{D^{\tau}_{sc}}(r_i) = \beta_c + \beta_f(\exp(-E_{D^{\tau}_{sc}}(r_i)/(2\sigma))) \tag{11}$$

where β_c, β_f, σ are parameters that determine how much we penalize for distances that are large versus distances that are smaller. For any edge point q, by applying (11) to the scale at which $\vec{F}_q$ was detected, we obtain the *modulated* image torque that is sensitive to the target object class:

$$\tau^{\omega}_{pq} = \vec{r}_{pq} \times (W_{D^{\tau}_{sc}}(q)\vec{F}_q) \tag{12}$$

where $\vec{f}(\vec{F}_q) = W_{D^{\tau}_{sc}}(q)\vec{F}_q$ as in (4). Finally we can compute the modulated torque per patch P by replacing τ_{pq} in (2) with τ^{ω}_{pq}:

$$\tau^{\omega}_P = \frac{1}{2|P|}\sum_{q \in E(P)} \tau^{\omega}_{pq} \tag{13}$$

For 2.5D images, we add in the depth constraint $w_{pq} = \text{abs}(I_d(p) - I_d(q))$ similarly as described in Section 3.1 to redefine the modulated torque with depth information

$$\tau^{\omega}_{d_{pq}} = \vec{r}_{pq} \times (w_{pq} W_{D^{\tau}_{sc}}(q)\vec{F}_q) \tag{14}$$

and we define the modulated torque per patch in the same way as in equation (13) by replacing τ^{ω}_{pq} with $\tau^{\omega}_{d_{pq}}$.

A crucial point to note is that even though our approach does not consider *all* possible lengths and combinations of the test edges with the model, by embedding $W_{D^{\tau}_{sc}}$ with the mid-level torque operation, we retain all the advantages of the image torque. As long as we have sufficiently strong support for an edge to belong to the target arranged in a coherent manner with other edges of similar weights, it is a strong indication of the presence of the target object. τ^{ω}_P thus transforms the original image torque so that it is now tuned towards the object model: $W_{D^{\tau}_{sc}}(q)$ is large for edge points q from test codons $\mathcal{C}_g$ that are similar to the model codons in $\mathcal{C}_{mo}$ while it is small for codons that are dissimilar to the model. In addition, because $W_{D^{\tau}_{sc}}(q)$ is derived from codon comparisons with the model, codons from other (unknown) object categories that are similar to the model will be detected as well: for example apples and oranges (round), sticks and rods (elongated), and so on. Viewed in this way, our approach can generalize to be sensitive to common object parts if we use a generic model of such parts. On the other hand, if the model is extremely specific and it has codons that are unique to the particular object class, then our approach becomes very selective. The choice between how selective or general we want the model to be is task-dependent and selecting the appropriate model automatically is part of our future work.

We summarize the entire approach in Algorithm 1. The inputs are an image edge map I_e of size $W \times H$, and $\mathcal{C}_{mo}$, the set of model codons for the target model. The output is the final modulated torque value map T^m_I that contains the modulated torque per patch, $\tau^{\omega}_P, P \in I_e$ at every image point. When the depth image I_d is available, the algorithm remains the same except we replace equations (2) and (13) with equations (3) and (14) respectively. The run-time complexity of the complete approach is $O(|\mathcal{P}_c| \times \mathcal{J} \times d \times l)$ as it is dominated by Step 3.

Algorithm 1. Pseudocode for the proposed approach.

Input: Image edge map I_e, model codon set $\mathcal{C}_{mo} = \{\mathcal{M}_1, \ldots, \mathcal{M}_l\}$ and torque shape contexts per model codon: $(m_i^{\angle}, m_i^{sc}), i \in \mathcal{M}_l$ for the ith edge point in $\mathcal{M}_l$
Output: Modulated torque map, T_I^m
Step 1: Compute original torque map T_I
for $(r, c) \leftarrow (1, 1)$ **to** (H, W) **do**
  Compute equation (2) for every patch $P_s(r, c)$ centered at (r, c) over I_e over all scales $s \in \mathcal{S}$;
  $T_I(r, c) \leftarrow \max_{s\in\mathcal{S}} \tau_{P_s(r,c)}$;
end
Step 2: Extract top $|\mathcal{P}_c|$ torque centers, $p_c \in \mathcal{P}_c$ from T_I;
Step 3: Compute modulated torque map T_I^m
for $p_c \in \mathcal{P}_c$ **do**
  Select contour fragments with torque contribution $> t_c$, $\mathcal{Q}_{p_c}$;
  Group neighboring torque centers, $\mathcal{Q}_{r_c}$, to form a larger set of d test codons $\mathcal{C}_g = \{\mathcal{R}'_1, \ldots, \mathcal{R}'_d\}$;
  Compute torque shape context per test codon: $(g_i^{\angle}, g_i^{sc}), i \in \mathcal{R}'_d$, for the ith edge point in $\mathcal{R}'_d$;
  Get O_g by computing the cross-correlation between angular bins of torque shape contexts at the torque center (Section 3.2.2);
  $\mathcal{O}_g \leftarrow \{O_g, O_g + 90, O_g + 180, O_g + 270\}$;
  for $o_g \in \mathcal{O}_g$ **do**
    for $\mathcal{R}'_e \in \mathcal{C}_g$ **do**
      Group neighboring $\mathcal{J}$ test codons: $\mathcal{R}'_{\{e\},\ldots,\{e+\mathcal{J}\}}$;
      Unrotate all test codons using o_g;
      for $(a, b) \leftarrow (1, 1)$ **to** $(\mathcal{J}, l)$ **do**
        Compute $V(a, b) = D^{\tau}_{sc}(\mathcal{R}'_{\{e\},\ldots,\{e+a\}}, \mathcal{M}_b)$ via equation (10);
      end
      $E_{D^{\tau}_{sc}}(e, o_g) \leftarrow \min_{\mathcal{J},l} V$;
    end
    $E_{D^{\tau}_{sc}}(o_g) \leftarrow \min_e E_{D^{\tau}_{sc}}(e, o_g)$;
  end
  $E_{D^{\tau}_{sc}} \leftarrow \min_{o_g} E_{D^{\tau}_{sc}}(o_g)$;
  for $e \leftarrow 1$ **to** d **do**
    for $r_i \in \mathcal{R}'_e$ **do**
      Convert $E_{D^{\tau}_{sc}}$ to weights $W_{D^{\tau}_{sc}}(r_i)$ via equation (11);
      Compute $\tau^{\omega}_{p_c r_i}$ via equation (12);
    end
  end
  for $(r, c) \leftarrow (1, 1)$ **to** (H, W) **do**
    Compute $\tau^{\omega}_{P_s(r,c)}$ via equation (13) for every patch $P_s(r, c)$ centered at (r, c) over I_e over all scales $s \in \mathcal{S}$;
    $T_I^m(r, c) \leftarrow \max_{s\in\mathcal{S}} \tau^{\omega}_{P_s(r,c)}$;
  end
end

4. Experiments

We perform experiments over four datasets. The first one, termed the UMD Hand-Manipulation dataset, is collected by a mobile robot observing humans performing manipulation activities using various tools and objects. This dataset is challenging because the hands, tools and objects induce occlusions, clutter and deformations (translation, scale and rotation), which are typical of manipulation activities. The goal is to show that our approach can handle such situations reliably. The second dataset is the CMU Kitchen Occlusion dataset (Hsiao and Hebert, 2012) that consists of eight common kitchen objects collected under severe occlusions and clutter. We demonstrate our approach's ability to detect the presence of the target from a single viewpoint and compare its performance with state-of-the-art template-based object detectors embedded with a learned occlusion model. To show that our approach compares well with other state-of-the-art contour-based object recognition approaches, we use the ETHZ-Shapes dataset for evaluating object detection and localization performance when there are significant variations in environmental conditions: background, lighting and camera viewpoints. Finally, we demonstrate the feasibility of our approach on a mobile robot platform where the task is to search for a specific object in clutter as the robot moves around the table, inducing occlusions and viewpoint changes. All data and code used in the experiments are available online at http://www.umiacs.umd.edu/research/POETICON/contour_based_recognition/

For all four experiments, we use the following meta-parameters: $\gamma_{sc} = \gamma_{\tau} = 0.5$, $\beta_c = 0.05, \beta_f = 0.95, \sigma = 0.05$, $\sigma_{\mathcal{K}} = 0.5$. These parameters were determined by optimizing the mean precision rate with groundtruth from a separate subset of 100 training images derived from the four datasets used in the experiments. The threshold to select the strongest edges, t_c, is set to the 50$^{\text{th}}$ percentile of the ranked

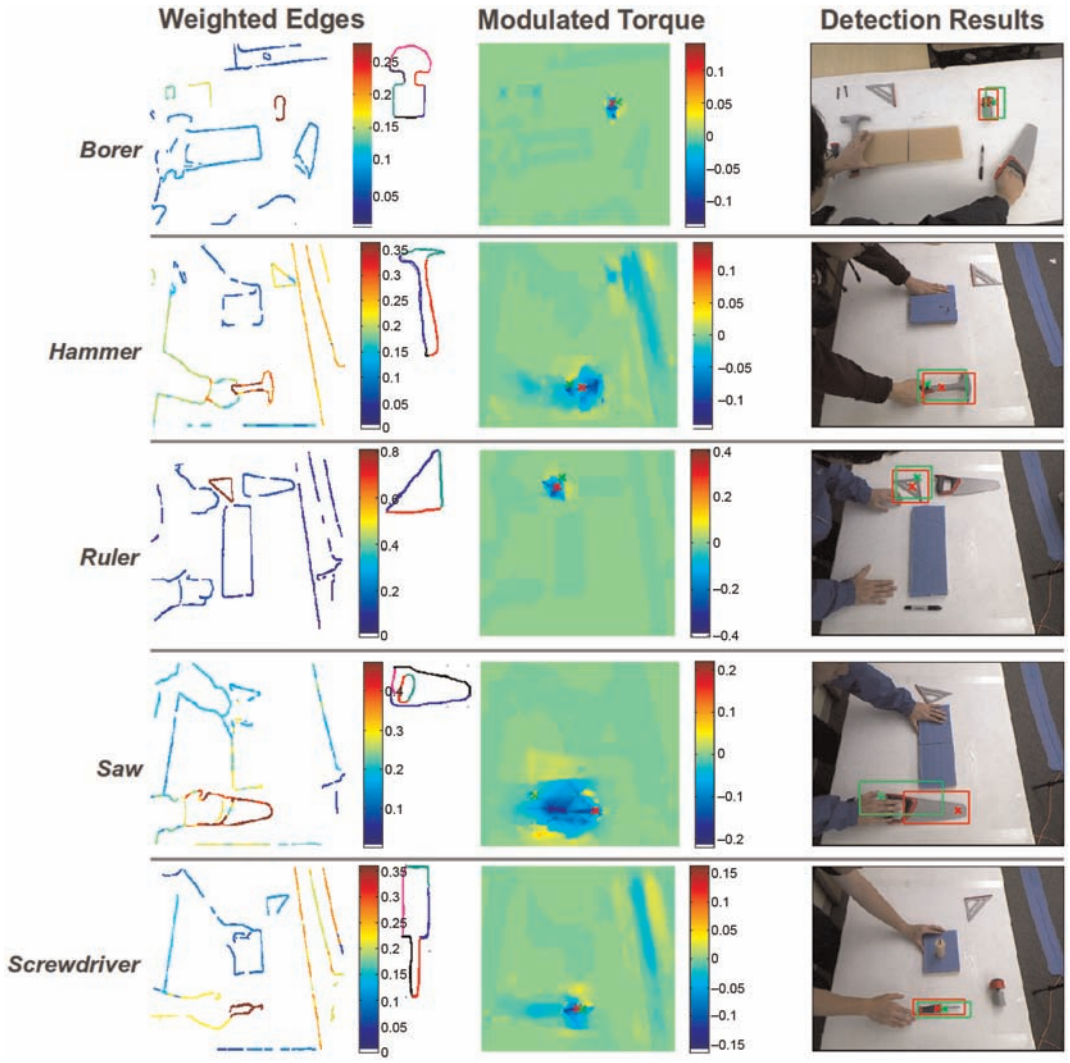
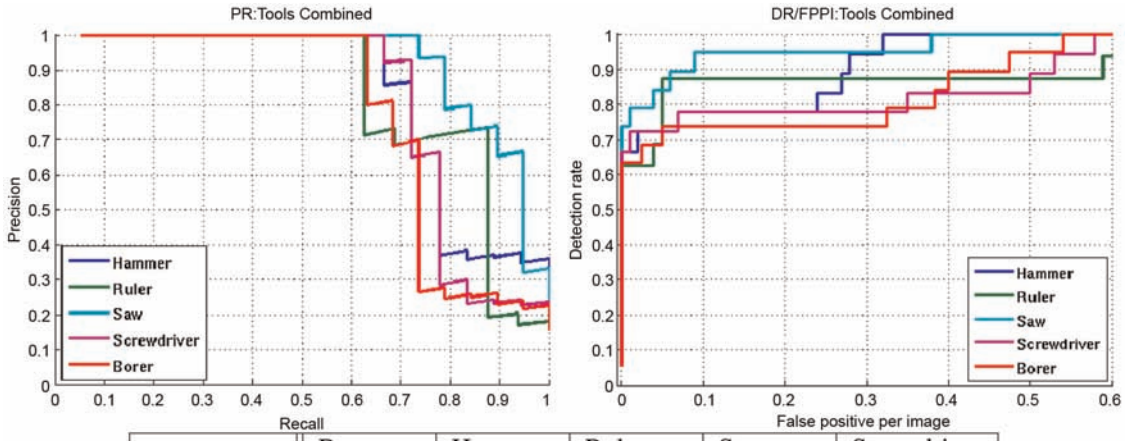


Fig. 13. Detection results from five sample frames of the UMD Hand-Manipulation dataset. (Rows) Target object class: (top to bottom) `Borer`, `Hammer`, `Ruler`, `Saw`, `Screwdriver`. (Columns) Left: $W_{D_{sc}^{\tau}}$ where red means higher values and target model contours at top-right; middle: modulated torque showing the top two object detections (red and green crosses); right: RGB frames overlaid with detection results. Note that for `Hammer` and `Saw`, the objects are partially occluded by the hands.

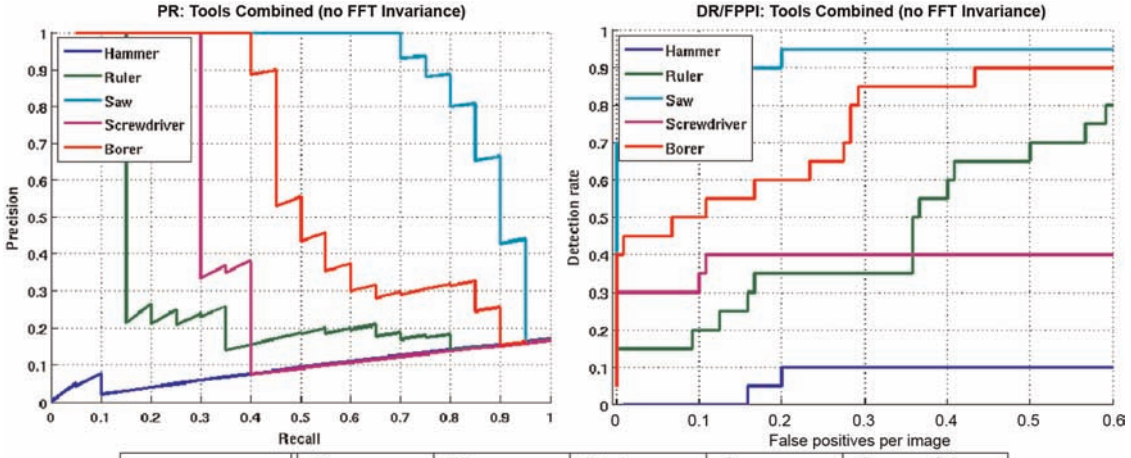
torque contribution scores from the grouped edges. The number of codon neighbors to combine, $\mathcal{J}$, is set to three for all object categories except for `Giraffes` and `Swans` (from the ETHZ-Shapes dataset), which have $\mathcal{J} = 5$ so as to fully account for long thin structures (neck and legs) that are common in these two categories. It is possible to set $\mathcal{J} = 5$ for *all* categories, but at the cost of longer processing time. The recognition accuracy would not be affected since we are simply doing a more extensive search over larger scales of combined codons. We use the Pb edge detector of Martin et al. (2004) to derive I_e. For computing torque, we search over image patches with sizes ranging from three pixels to a quarter of the input image height and width. In practice, we found that the recognition accuracy (mean precision) of the approach is not very sensitive to the parameters used, but setting $\mathcal{J}$ and $|\mathcal{P}_c|$ to large values will slow down the recognition times significantly. Using the current parameters, typical running times for a 320 $\times$ 240 image are around ~15 s using a Matlab implementation running on a Core i7 2.4 GHz machine.

We predict the target's location and scale from the modulated torque map, T_I^m, by selecting the largest modulated torque response over the same image patch scales as noted above. For evaluating object-detection performance, we admit a true positive using the PASCAL criterion: when the overlap between the predicted object's bounding box and the ground truth bounding box exceeds 50% of the union of the two boxes. For multiple detections near ground truth, we select the one with the largest absolute torque value. For scoring the detections, we normalize the modulated torque at the predicted object center, τ_P^m (replacing τ_{pq} in (2) with τ_{pq}^m from (4)), with τ_P



	Borer	Hammer	Ruler	Saw	Screwdriver
AP	0.77	0.82	0.80	0.89	0.79
0.3/0.4 FPPI	0.74/0.90	0.94/1.00	0.88/0.88	0.95/1.00	0.78/0.83

Fig. 14. (Top-left) Precision/recall (PR) curves over the six videos in the UMD Hand-Manipulation dataset. (Top-right) Corresponding DR/FPPI curves. (Below) Interpolated average precision (AP) and detection rates at 0.3/0.4 FPPI over the five tool categories.



	Borer	Hammer	Ruler	Saw	Screwdriver
AP	0.61	0.10	0.34	0.85	0.41
0.3/0.4 FPPI	0.85/0.85	0.10/0.10	0.35/0.60	0.95/0.95	0.40/0.40

Fig. 15. Performance without incorporating rotational invariance via FFT. (Top-left) Precision/recall (PR) curves over the six videos in the UMD Hand-Manipulation dataset. (Top-right) Corresponding DR/FPPI curves. (Below) Interpolated average precision (AP) and detection rates at 0.3/0.4 FPPI over the five tool categories.

4.1. Evaluation over UMD Hand-Manipulation dataset

We demonstrate our approach on a dataset collected by a mobile robot that is actively observing a table full of tools/objects in clutter manipulated by humans. This dataset, termed the UMD Hand-Manipulation dataset, consists of six video sequences (around 1500 frames each) of three different human subjects constructing a partial wooden frame using five tool classes: {`Borer, Hammer, Ruler, Saw, Screwdriver`}. This dataset is challenging because it has significant occlusions and orientation changes due to the hands and active nature of the frame-making process. The goal is to show that our approach is able to handle partial occlusions under various viewpoints/orientations. In addition, we demonstrate the contribution of estimating O_g using FFT (Section 3.2.2) in improving the recognition accuracy.

We used the meta-parameters and evaluation procedure as indicated above. For obtaining the target model codons, we used the initial first 10 frames and hand-annotated the target tool's contours to obtain the model codons. We then

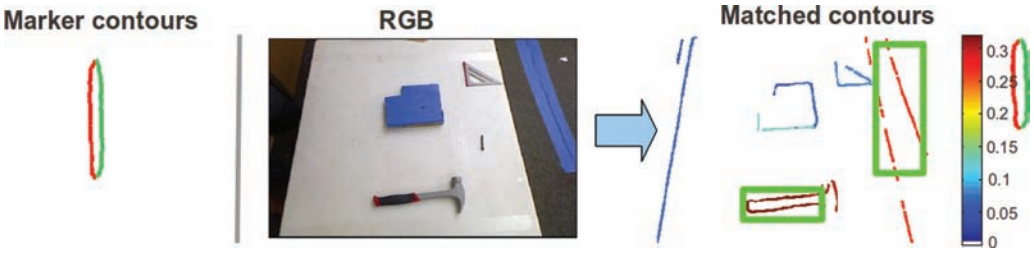


Fig. 16. When contour information alone is not enough. (Left) Model contours of the `Marker` class. (Right) Contours that have long parallel lines are highlighted (in green boxes): for example the handle of the hammer or the sides of a tape.

evaluated the rest of the sequence at sample intervals of 10 frames each, which yielded a total of around 800 evaluated frames in the entire dataset. We show some results from sample frames of the dataset in Figure 13: final edge weights $W_{D_{sc}^{\tau}}$ and the predicted target objects with centers marked as crosses.

For evaluation, we report the precision/recall rates and corresponding detection rate/false positives per image (DR/FPPI) curves. The results are summarized in Figure 14 for all five tools considered and compared to Figure 15 where no rotational invariance is applied to the procedure. From the results of the full approach, we are able to localize the target objects in clutter with average precision ranging from 0.77 to 0.89, with detection rates at the standard 0.3/0.4 FPPI that range from 0.74 to 1.00. These results are on par with current object recognition approaches. The best detection using the full approach comes from `Saw` and `Hammer`, and it is probably due to the fact that the contours belonging to these two classes are very distinctive (and hence easy for discrimination) compared with other tools. The worst results (in terms of average precision) are from `Borer`, which is most confused with `Screwdriver`. This is not surprising since both of these tools share many common parts (with similar functions).

The contribution of estimating O_g via FFT is also clearly shown in Figure 15 when we note the improvements in average precision that range from 0.04 (`Saw`) to 0.72 (`Hammer`). The improvement is modest for `Saw` as for most of the frames the target tool was well aligned with the model's original orientation. This makes the estimation of O_g unnecessary for most of the frames considered. However, for other tools, the improvements are much more significant since they were placed and manipulated in very different orientations (such as `Hammer`) compared to the model (see Figure 13 first column where the model codons are shown on the top right).

The decrease in performance of `Borer` (and to a large extent `Screwdriver` as well) compared with other tools as noted in Figure 14 highlights one of the key shortcomings of the approach: the mid-level groupings over multiple scales do not capture *enough* global information about parts and their relationships to accurately separate out objects that consist of a subset of contours from other targets. An extreme example is that of the `Marker` class, which we have not considered here, but consists only of two parallel contours as shown in Figure 16, left. Due to the small number of contours in the model, such a configuration is highly ambiguous (Figure 16, right). This result points to future work that should incorporate additional *global* mid-level information on the spatial configuration of object parts. For example, the `Hammer` class consists of two distinctive (functional) parts: 1) the handle and 2) the hammer head. Modifying the torque operator to enforce the grouping at the level of these subparts would enable us to distinguish hammer handles from markers since a marker consists solely of a single part.

4.2. Evaluation over CMU Kitchen Occlusion dataset

We investigate the performance of our approach in severe clutter and occlusion using the single viewpoint subset of the CMU Kitchen Occlusion dataset introduced by Hsiao and Hebert (2012) and compare it with the state-of-the-art LINE2D algorithm of Hinterstoisser et al. (2012) as baseline and the robust version, rLINE2D, which compares edge points with the model's gradient orientation to decide if an edge point is consistent with a learned occlusion model. We did not compare the full approach of Hsiao and Hebert (2012) that includes the probabilistic occlusion prior, since our approach does not explicitly model it. The dataset consists of eight textureless objects: {`baking-pan`, `colander`, `cup`, `pitcher`, `saucepan`, `scissors`, `shaker`, `thermos`} placed among other common kitchen objects with a severe amount of occlusion. There are 100 testing frames per object class, with a single positive target per test image. For training, we are provided with a single viewpoint of the model as a mask and an image. We used the training image mask to extract the model codons and used the same meta-parameters and evaluation procedure as described above over all eight object categories. Since Hsiao and Hebert (2012) used the same PASCAL criterion to generate DR/FPPI curves, we are able to directly compare our results with LINE2D and rLINE2D as shown in Figure 17. The detection rates at 0.3/0.4/1.0 FPPI are summarized in Table 1.

From the DR/FPPI curves, we first note that for all the objects our method significantly outperforms LINE2D and

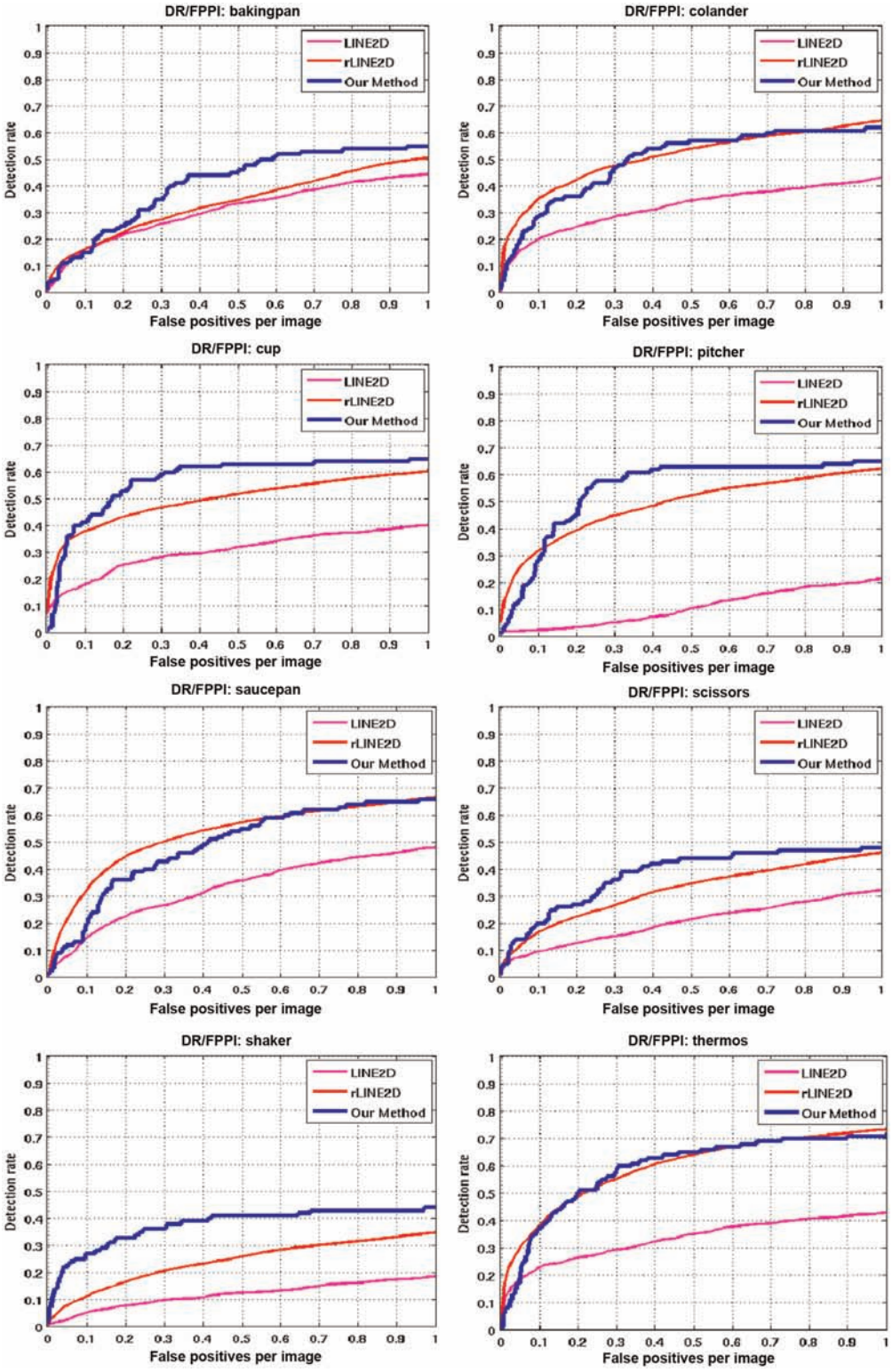


Fig. 17. DR/FPPI curves comparing our approach with LINE2D (Hinterstoisser et al., 2012) and rLINE2D (Hsiao and Hebert, 2012) over the eight object categories in the CMU Kitchen Occlusion dataset.

has a performance that is at least on par with or better than rLINE2D which includes an occlusion model of the target (which our method does not have). Second, from Table 1, our approach is able to obtain much better detection rates with a lower number of false positives (lower FPPI) compared to both methods. This shows that our approach is discriminative even when severe clutter is present. This improvement is due to the partial hierarchical matching via

Table 1. Comparing detection rates with our method, LINE2D (Hinterstoisser et al., 2012) and rLINE2D (Hsiao and Hebert, 2012) at 0.3/0.4/1.0 FPPI over the CMU Kitchen Occlusion dataset.

	bakingpan	colander	cup	pitcher
Our method	**0.35/0.44/0.55**	0.47/**0.54**/0.62	**0.59/0.62/0.65**	**0.58/0.62/0.65**
LINE2D	0.26/0.29/0.44	0.28/0.31/0.43	0.28/0.29/0.40	0.05/0.07/0.21
rLINE2D	0.27/0.32/0.51	**0.48**/0.51/**0.65**	0.47/0.49/0.60	0.45/0.48/0.62
(cont.)	saucepan	scissors	shaker	thermos
Our Method	0.43/0.49/0.66	**0.36/0.42/0.48**	**0.36/0.39/0.44**	**0.58/0.63/0.84**
LINE2D	0.27/0.31/0.48	0.15/0.18/0.32	0.10/0.11/0.18	0.29/0.32/0.43
rLINE2D	**0.50/0.54/0.67**	0.27/0.31/0.46	0.20/0.23/0.35	0.55/0.60/0.73

codons and the torque shape context that match edge points with better accuracy while rejecting false positives with different torque centers more effectively compared to LINE2D or rLINE2D, which use gradient orientations only. We show some example detection results with the modulated torque in Figure 18 that illustrate how the approach performs over this dataset.

4.3. Evaluation over ETHZ-Shapes dataset

We further evaluate our approach using the ETHZ-Shapes dataset which is often used in the computer vision community as a standard baseline for evaluating 2D contour-based object recognition approaches. This dataset is divided into five object categories: {`Applelogos`, `Bottles`, `Giraffes`, `Mugs`, `Swans`}, and consists of 255 images containing instances of the objects with varying backgrounds, clutter, scales and viewpoints. We follow the same test/train split procedure as suggested by Srinivasan et al. (2010) for evaluation: the first half of each category is used to obtain the model codons from the ground truth contours and the remaining half, together with the rest of the images, are used for testing. Because this dataset is widely used, it enables us to compare the performance of our approach with other state-of-the-art contour-based object recognition approaches.

We focus on comparisons with recent state-of-the-art contour-based object-detection methods (Maji and Malik, 2009; Srinivasan et al., 2010; Ma and Latecki, 2011; Wang et al., 2012). The precision/recall curves of these methods and their interpolated average precision are compared with the proposed method in Figure 19 and Table 2 respectively. Across all five categories, the proposed approach is comparable with state-of-the-art procedures: its most dominant performance is for `Applelogos`. Averaged over all five categories, our approach is able to achieve the overall best mean average precision among the compared methods, with a small improvement over Ma and Latecki (2011).

In addition, we plot the DR/FPPI curves in Figure 20. The detection rates at 0.3 and 0.4 FPPI are compared with several reported results in the literature in Table 3. The detection performance at these two levels is consistently on par with the state of the art, with the largest improvements in `Applelogos` and `Giraffes`. We show some example results in Figure 21: the modulated torque with the final detections, and some failure cases. Similar to the discussion in the preceding sections, these cases occur due to the fact that some model codons between classes may be very similar (such as between `Swans` and `Giraffes`). A more discriminative learning approach that incorporates more global-level part-based information should yield even better results.

4.4. Object recognition in clutter by a mobile robot

We demonstrate the feasibility of our approach for practical robotic applications on our mobile robot platform (Figure 22, left). The robot consists of the Adept Pioneer P3-DX base together with a custom-made frame on which a Kinect RGB-Depth sensor[6] is attached via a directed perception PTU-D46 pan-tilt unit (PTU). The robot's software runs over the robot operating system (ROS) (Quigley et al., 2009) with appropriate interfaces implemented to send the Kinect RGB-Depth data to Matlab for processing by the proposed method. The robot is tasked with performing random movements using either the base or the PTU while observing a cluttered scene of objects on a table. The goal is to detect objects in clutter while inducing changes in viewpoint and occlusion from the movements. We used the same 'UMD-clutter' dataset reported in our previous work (Teo et al., 2013): we performed three different collections of the Kinect RGB-Depth data with a differing amount of clutter per dataset, with around 1000 frames per sequence. We focus on detecting four object categories (Figure 22, right): {`Book`, `Bowl`, `Mug`, `Spoon`} which are located at random positions on a table, under various degrees of occlusion. We used the same meta-parameters and evaluation described above. For this dataset, we evaluated frames at intervals of 10 frames, yielding around 300 frames that were considered for evaluation. As a baseline, we compared it with our previous work (Teo et al., 2013) termed 'Shape-Torque' that uses multiple shape templates to define a multi-view model to modulate the torque response towards the desired target object. As a comparison, we used the recent HMP method of Bo et al. (2012) that learns a dictionary of RGB-Depth features for object recognition. A linear SVM classifier is then trained over the features for

Table 2. Comparing interpolated average precision with the proposed method over the ETHZ-Shapes dataset.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	**0.917**	0.931	0.796	**0.888**	0.891	**0.885**
Maji and Malik (2009)	0.869	0.724	0.742	0.806	0.716	0.771
Srinivasan et al. (2010)	0.845	0.916	0.787	**0.888**	0.922	0.872
Ma and Latecki (2011)	0.881	0.920	0.756	0.868	**0.959**	0.877
Wang et al. (2012)	0.866	**0.975**	**0.832**	0.843	0.828	0.869

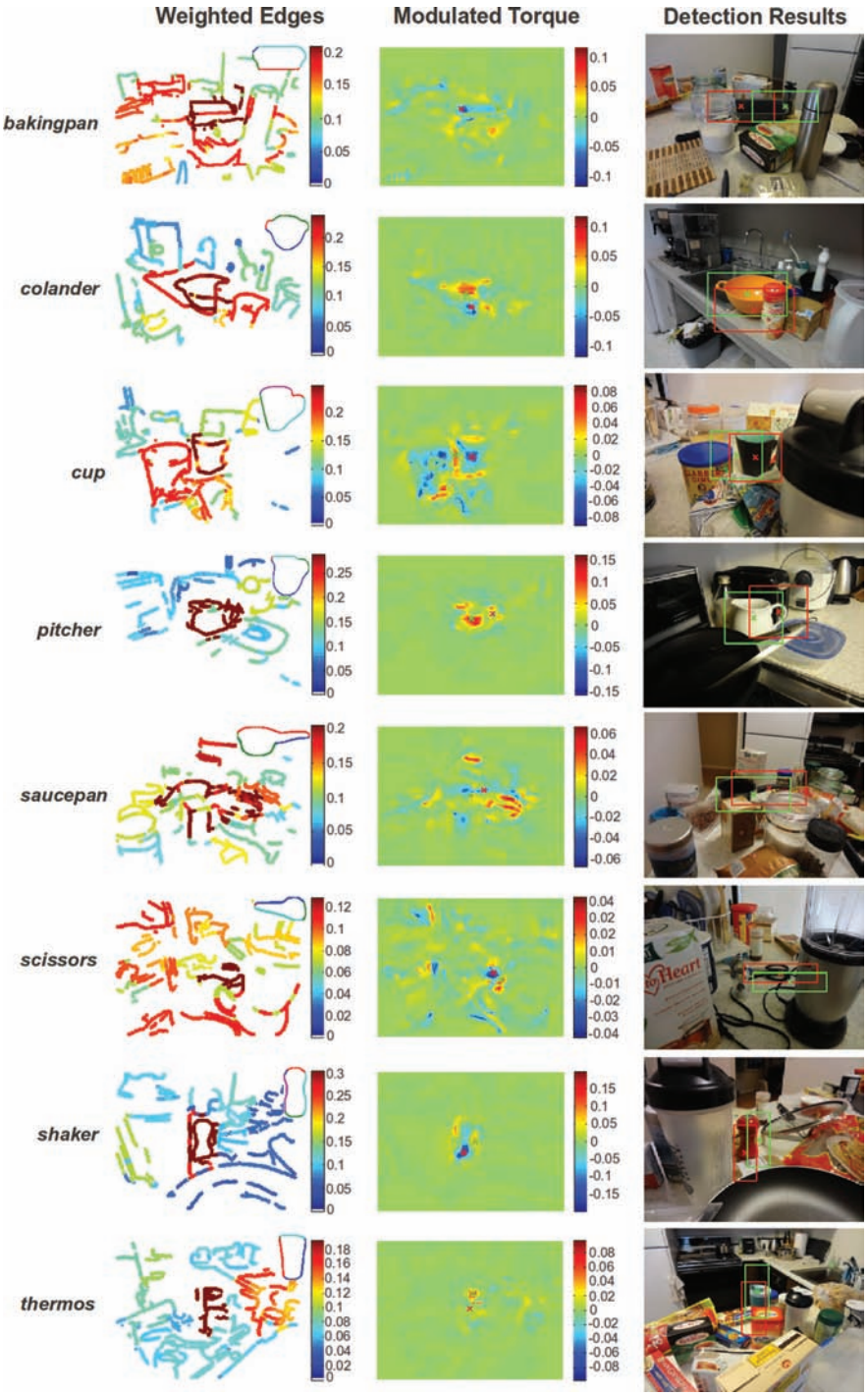


Fig. 18. Detection results for the eight objects in the CMU Kitchen Occlusion dataset. (Rows) Target object class: (top to bottom) `bakingpan`, `colander`, `cup`, `pitcher`, `saucepan`, `scissors`, `shaker`, `thermos`. (Columns) Left: $W_{D_{sc}^T}$ where red means higher values and target model contours at top-right; middle: modulated torque showing the top two object detections (red and green crosses); right: RGB frames overlaid with detection results.

Table 3. Comparing detection rates at 0.3/0.4 FPPI over the ETHZ-Shapes dataset.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	**1/1**	**1/1**	**0.930/0.930**	**0.958**/0.958	0.938/0.938	**0.965/0.965**
Maji and Malik (2009)	0.95/0.95	0.929/0.964	0.896/0.896	0.936/**0.967**	0.882/0.882	0.919/0.932
Srinivasan et al. (2010)	0.95/0.95	**1/1**	0.872/0.896	0.936/0.936	**1/1**	0.952/0.956
Ma and Latecki (2011)	0.92/0.92	0.979/0.979	0.854/0.854	0.875/0.875	**1/1**	0.926/0.926
Wang et al. (2012)	0.90/0.90	**1/1**	0.92/0.92	0.94/0.94	0.94/0.94	0.940/0.940
Riemenschneider et al. (2010)	0.933/0.933	0.970/0.970	0.792/0.819	0.846/0.863	0.926/0.926	0.893/0.905
Ferrari et al. (2010)	0.777/0.832	0.798/0.816	0.399/0.445	0.751/0.8	0.632/0.705	0.671/0.72

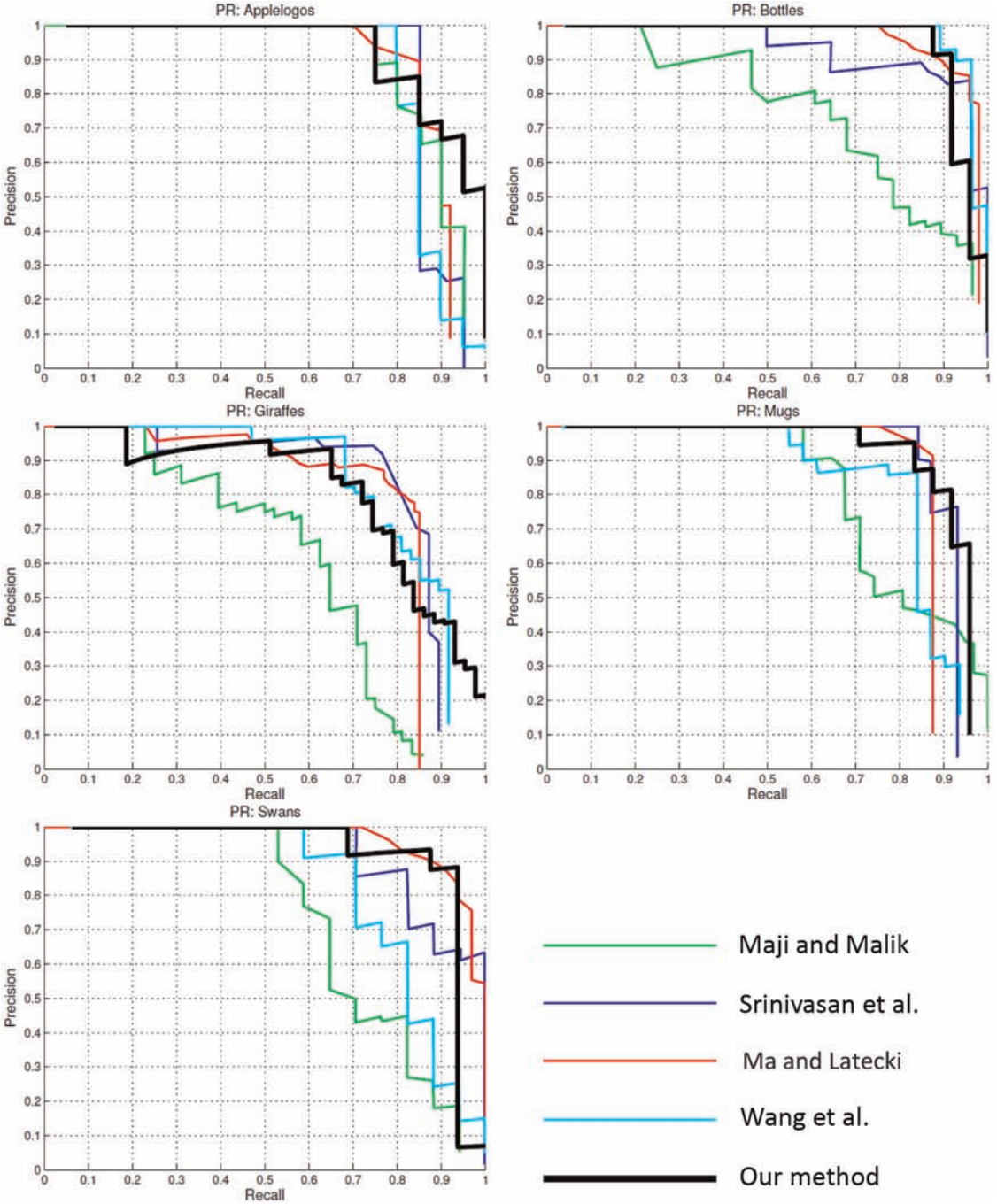


Fig. 19. Precision/recall (PR) curves comparing Maji and Malik (2009), Srinivasan et al. (2010), Ma and Latecki (2011) and Wang et al. (2012) to the proposed method over the ETHZ-Shapes dataset.

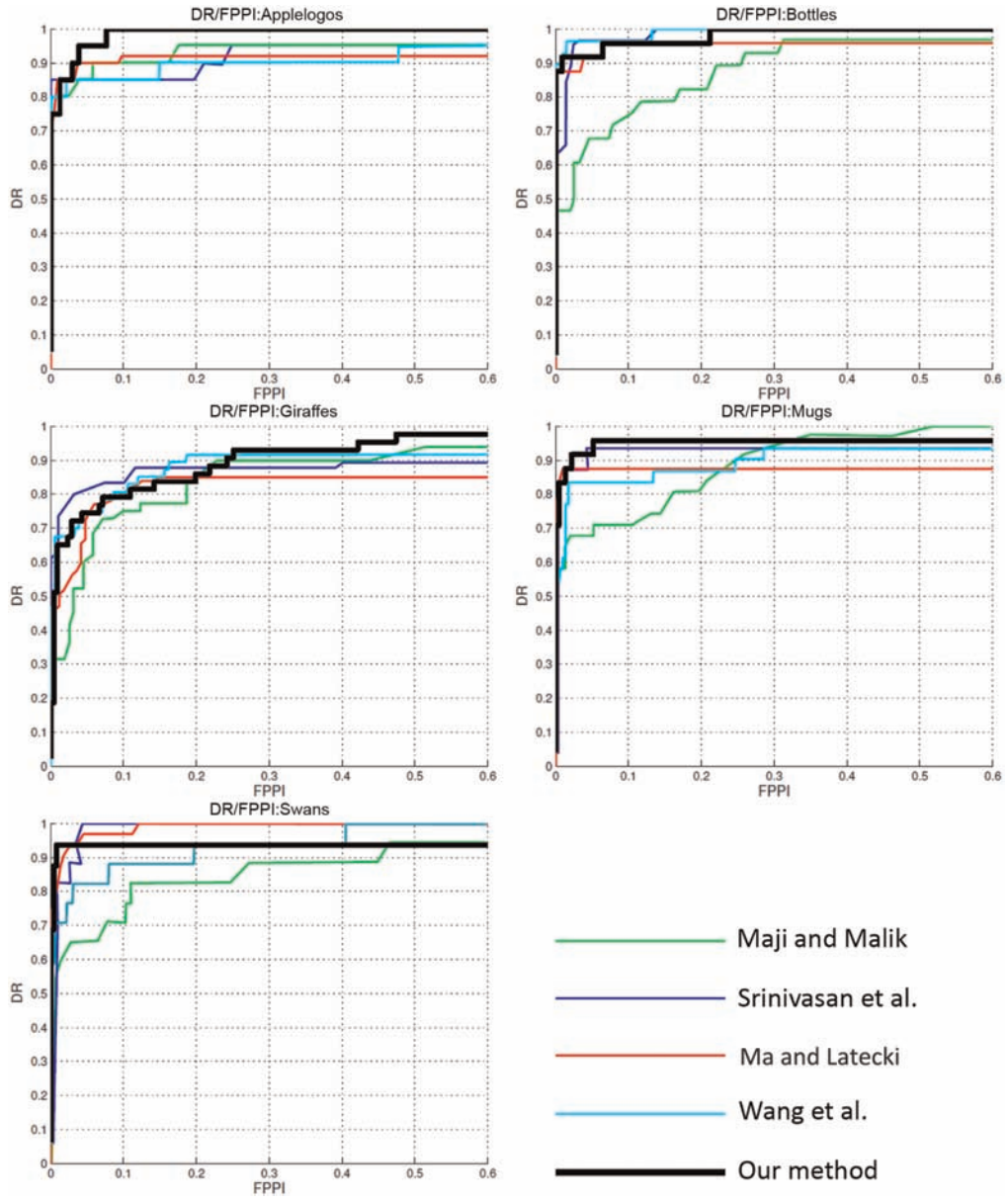


Fig. 20. Comparison of DR/FPPI curves over the ETHZ-Shapes dataset.

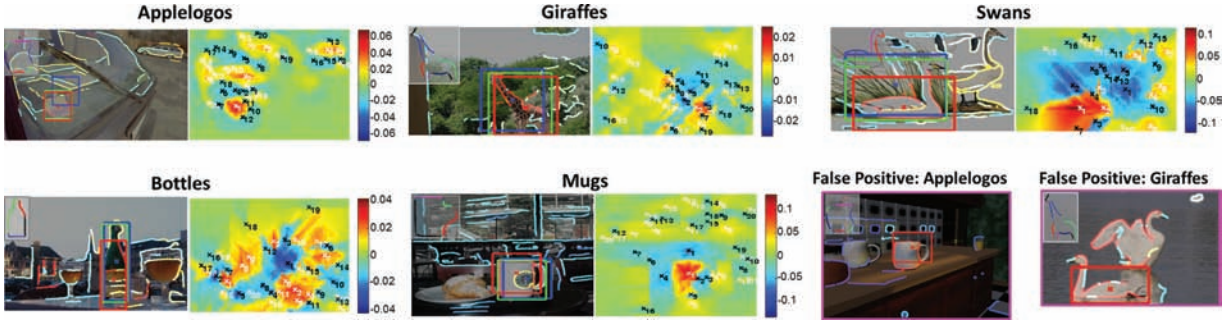


Fig. 21. Some example detection results with their modulated torque. Edges show values of $W_{D_{sc}^{\tau}}$, green boxes are ground truth, red and blue boxes are the top min/max modulated torque values. Top row (left to right): `Applelogos`, `Giraffes`, `Swans`. Bottom row (left to right): `Bottles`, `Mugs`. False detections of `Applelogos` and `Giraffes`. Best viewed in color.

Table 4. Comparing detection rates of our approach that uses depth information and one that does not use depth information with the baseline Shape-Torque (Teo et al., 2013) and HMP (Bo et al., 2012) at 0.3/0.4 FPPI over the UMD-clutter dataset.

	Book	Bowl	Mug	Spoon	*Wood Spoon*
Our Method + Depth	0.27/0.37	**0.61/0.68**	**0.57/0.60**	0.32/0.37	**0.42/0.42**
Our Method (no Depth)	0.29/**0.39**	0.56/0.62	0.54/0.59	**0.43/0.46**	0.36/0.36
Shape-Torque (Teo et al., 2013)	**0.33**/0.33	0.14/0.14	0.43/0.43	0.17/0.17	0.09/0.09
HMP-RGBDepth (Bo et al., 2012)	0.00/0.01	0.13/0.38	0.00/0.00	0.06/0.06	0.25/0.33
HMP-RGB	0.00/0.00	0.00/0.01	0.00/0.00	0.00/0.00	0.02/0.05

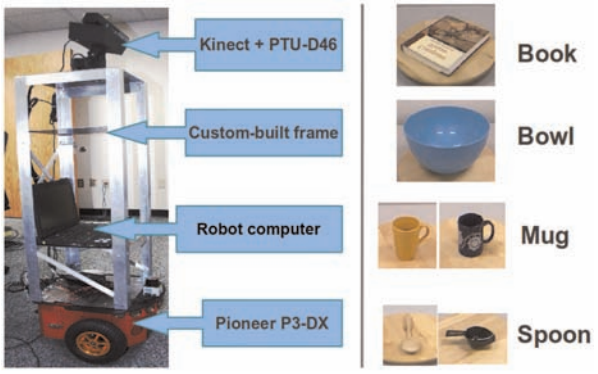


Fig. 22. (Left) The mobile robot with relevant hardware components highlighted used in the experiment. (Right) The four object categories used. For `Mug` and `Spoon`, two different instances exist in the dataset.

the four target object categories. For evaluation, we select the top 20 initial torque fixations per test frame which are processed by the SVM classifier.

In order to evaluate the contribution of the depth information in influencing the detection rates of the approach, we compared the standard (no depth) approach for computing modulated image torque (equation (12)) with the approach that uses depth information (equation (14)). For HMP, we trained two SVM classifiers: one using RGB features only (HMP-RGB) and another using RGB-Depth features (HMP-RGBDepth). Since HMP does not provide bounding boxes, we cannot use the PASCAL criterion for evaluation. Instead, we admit all positive predictions, which results in a much higher detection rate at high recalls compared to the other approaches that localize the prediction with a bounding box. Figure 23 shows the DR/FPPI curves for the entire dataset over the four object categories considered. The detection rates at 0.3/0.4 FPPI are summarized in Table 4.

From the results, we see that with the exception of `Spoon`, the proposed method with depth information is on par with or better than the baseline and standard non-depth approach at both FPPI levels. This shows that, given a cluttered environment, using the depth information enables us to reduce the influence of false contour groupings that have very different depth values and hence are unlikely to come from the same object. Figure 24 shows an example of how using depth information reduces false groupings to improve the detection of the target. Obviously this assumption has its limitations, especially for objects with large depth disparities, for example `Book`, and this is shown by a slightly worse performance compared with not using depth information at 0.4 FPPI. The method also significantly outperforms both variants of HMP, which use RGB information in addition to depth. This is indicative of the robustness of using contour information for recognition under such challenging scenarios. Finally, it is interesting to note that the improvements of both variants of the approach against the baseline Shape-Torque approach occur when we use only a single viewpoint in the model, versus the multiple (6 to 10) viewpoints used in Shape-Torque. This highlights the contribution of using our codon-based torque shape context for robust matching under occlusion and clutter.

For `Spoon`, the reason for the consistent poorer performance using depth is because the depth estimates from the *Black Spoon* instances are usually wrong. This is due to the dark surface coloration that tends to absorb the Kinect's IR radiation. The *Wood Spoon* instance, however, does not suffer from this issue. This is shown in the DR/FPPI curves of the *Wood Spoon* instances only (Figure 23, boxed), where depth information improves the result.

We show some results from sample frames of the dataset in Figure 25. Specifically the figure shows the final edge weights $W_{D_{sc}^{\bar{\tau}}}$, the modulated torque value map with depth constraint and the predicted objects with centers marked as crosses. The supplementary video, included as Multimedia Extension 1, contains output from the full dataset that shows the results of `Mug` detection.

5. Summary and future work

We have presented an approach to contour-based categorical object recognition, which makes use of a new mid-level image operator, the image torque, for the selection and grouping of target-specific object contours in clutter, occlusions and viewpoint changes. Our approach proceeds in two stages. In a first stage, we use the torque as an attention mechanism to find initial proto-object locations by applying the torque on simple edge responses possibly

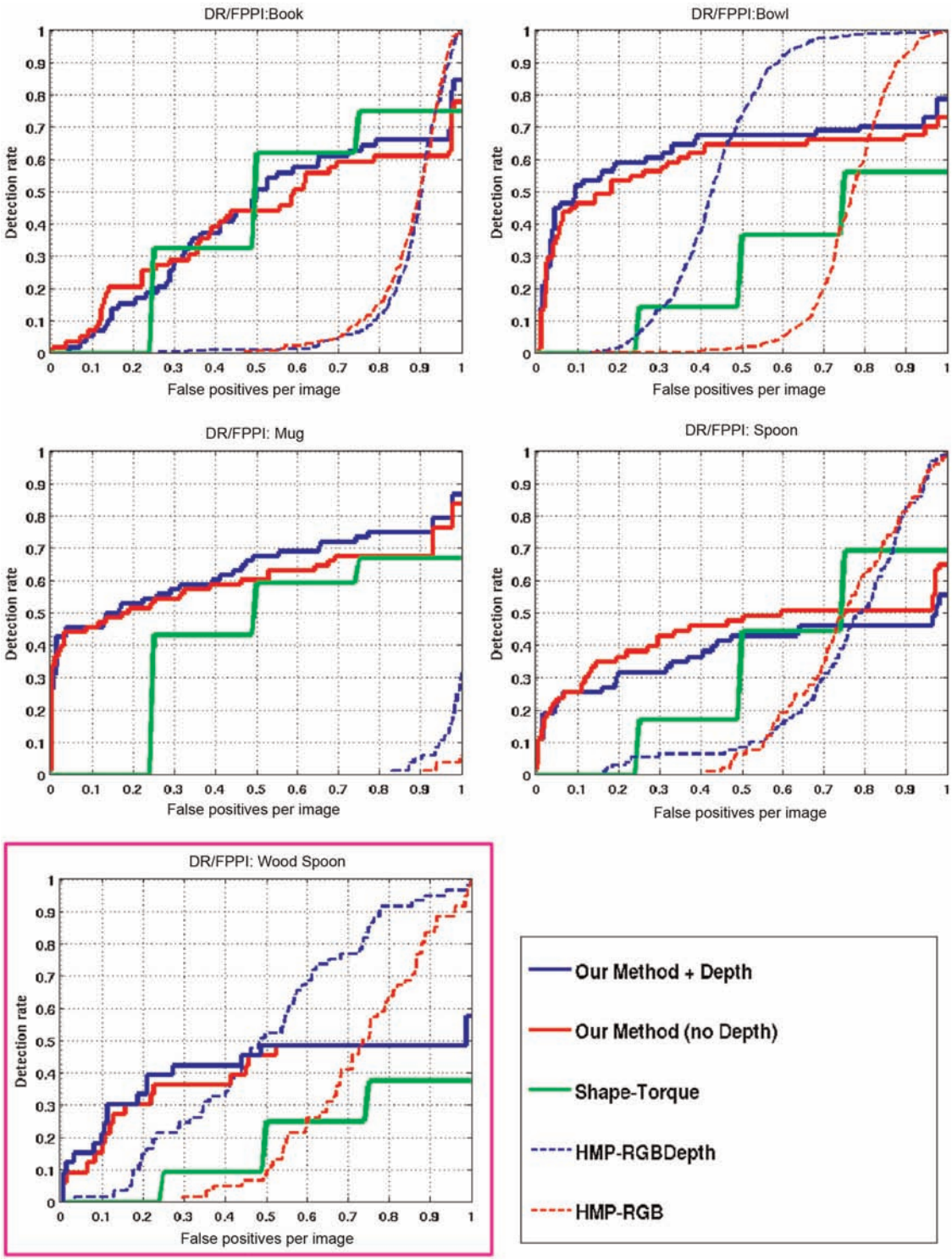


Fig. 23. DR/FPPI curves of the UMD-clutter data evaluated over four object classes. The DR/FPPI curves for the *Wood Spoon* (boxed) instance is presented to contrast with the results shown for the `Spoon` category, which is affected due to bad depth estimates, see text for details.

augmented with depth information. With the help of these proto-object locations, we then match in a multi-scale approach edges using a new shape context descriptor that takes into account boundary ownership information and object rotation. In a second stage, we then use the torque to group the matched edge responses by modulating their weights within the operator. We evaluated the approach over four datasets: 1) the UMD Hand-Manipulation dataset,

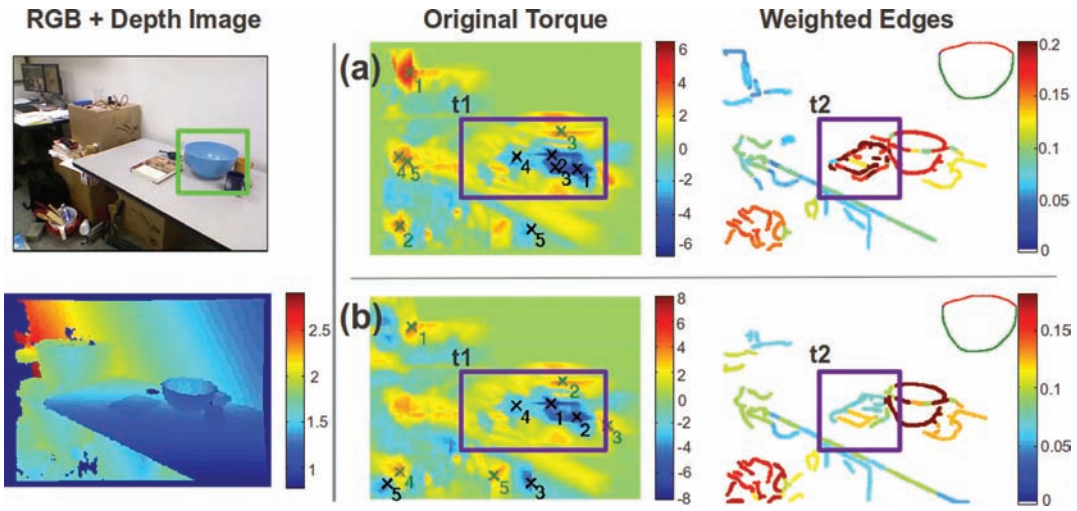


Fig. 24. How depth information helps in improving the image torque. (Left) Input Kinect RGB-D image with target `Bowl` in green box. (Right) Comparing effects of (a) not using depth information and (b) using depth information. Region t1: using depth information produces more depth-consistent grouping in (b) compared to (a): notice there are three fixations corresponding to three objects on the table in (b) compared to four in (a). Region t2: as a result of this grouping, we are able to combine and compare groups of codons more accurately with the model. In (a), codon groupings near `Book` are erroneously weighted more due to wrong groupings which are weighed down in (b) as their depth values are inconsistent. This enables the target `Bowl` to be correctly detected in (b).

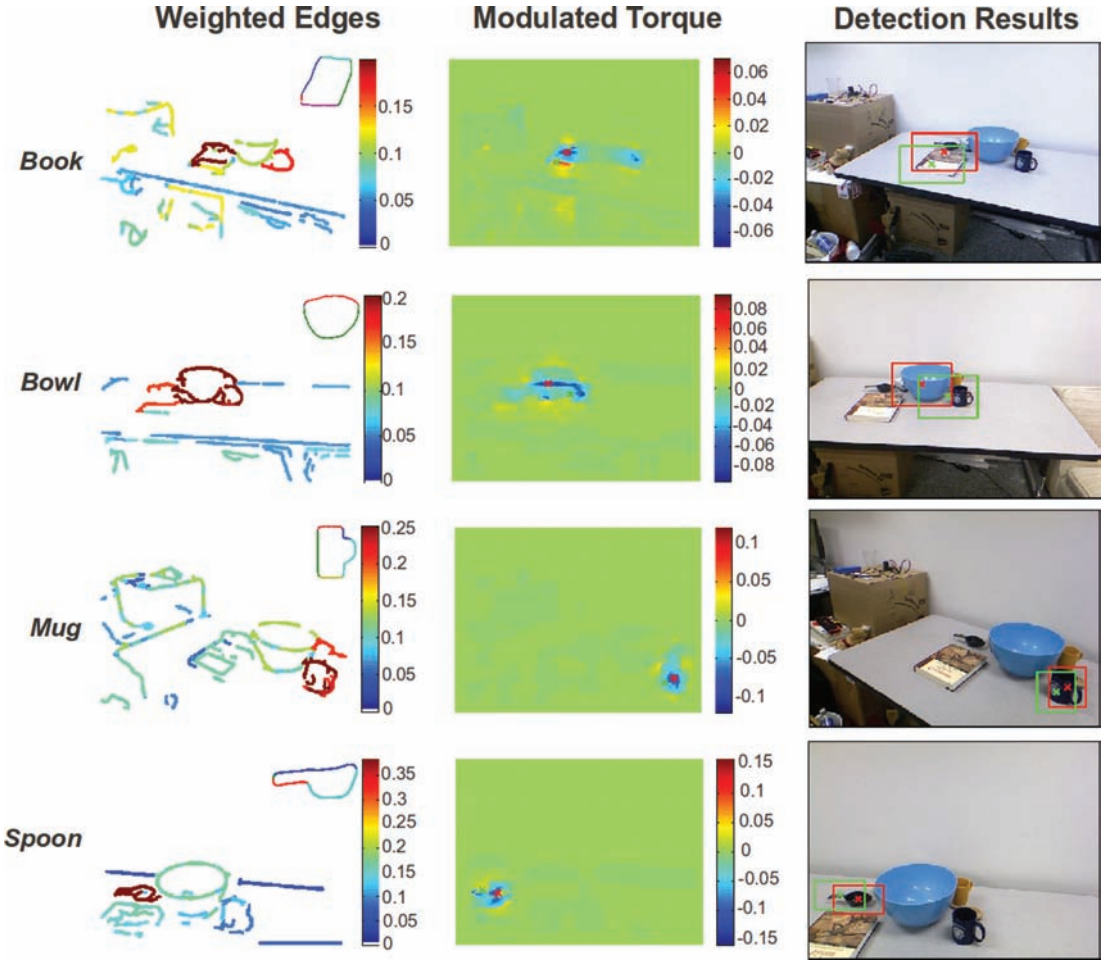


Fig. 25. Detection results using depth information for the four objects in the UMD-clutter dataset. (Rows) Target object class: (top to bottom) `Book, Bowl, Mug, Spoon`. (Columns) Left: $W_{D_{sc}^{\tau}}$ where red means higher values and target model contours at top-right; middle: modulated torque with depth constraint showing the top two object detections (red and green crosses); right: RGB frames overlaid with detection results.

2) the CMU Kitchen Occlusion dataset, 3) the ETHZ-Shapes dataset and 4) the UMD-clutter dataset collected by a moving robot observing a table with clutter. The results highlight the ability of the approach to handle occlusions, partial matches and orientation changes over large variations in environmental conditions which makes it suitable for practical robotic applications.

We have introduced here the concept of a grouping mechanism that is implemented as an operator that acts on semi-global image regions, and this operator interacts with both low-level and high-level information. In this work, we have demonstrated a very feasible implementation, but there may be many other ways the concept can be realized. First, we plan to investigate how to apply the torque to other edge or discontinuity information, such as discontinuities in image motion, or on depth edges in combination with the shapes of surfaces that surround them. Second, we will look into developing similar grouping mechanisms that implement other Gestalt principles, such as symmetry or responses to common global shapes, for example spirals, ellipses, or star-like shapes. Finally we will investigate how to incorporate higher-level grouping information of object parts to improve the performance of the approach for targets that share a significant number of contours.

Funding

This research was funded in part by the support of the European Union under the Cognitive Systems program (project POETICON++), the National Science Foundation under INSPIRE grant SMA 1248056, and support from the US Army, grant W911NF-14-1-0384 under the project: Shared Perception, Cognition and Reasoning for Autonomy.

Notes

1. Here with a slight abuse of notation, we describe $\vec{r}_{pq}$ and $\vec{F}_q$ as 2D vectors, and denote the cross-product of these 2D vectors as the signed scalar magnitude of the resulting vector obtained by cross-multiplying these vectors. Writing $\vec{r}_{pq}$ and $\overrightarrow{F_q}$ as 3D dimensional vectors (with 0 in the third component), their cross-product, $\vec{\tau}_{pq}$, either points 'out' (upwards) in which case τ_{pq} is positive, or downwards in which case τ_{pq} is negative.
2. The sign of τ_{pq} depends on the direction of the tangent vector. In this work, we define the direction based on the image contrast and we compute it based on the sign of the image gradient.
3. Available online at http://www.umiacs.umd.edu/research/SRVC/NSF-project/.
4. We define bins that are orientated towards the torque center as those bins that are captured within the half-circle centered along $\vec{r}_{p_cq_i}$, the vector with direction $\theta_{p_cq_i}$. Since the distribution $N(\theta_{p_cq_i}, \sigma_K^2)$ is positive everywhere, truncating the distribution so that it is active only between $\theta_{p_cq_i} + \pi/2$ and $\theta_{p_cq_i} - \pi/2$ achieves the desired effect.
5. The codons are indexed in a clockwise direction.
6 Information on the Kinect camera is available at http://en.wikipedia.org/wiki/Kinect.

References

Belongie S, Malik J and Puzicha J (2002) Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(4): 509–522.

Bo L, Lai K, Ren X, et al. (2011) Object recognition with hierarchical kernel descriptors. In: *2011 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1729–1736.

Bo L, Ren X and Fox D (2012) Unsupervised feature learning for RGB-D based object recognition. In: *ISER*.

Cortes C and Vapnik V (1995) Support-vector networks. *Machine Learning* 20(3): 273–297.

Crow FC (1984) Summed-area tables for texture mapping. *ACM SIGGRAPH Computer Graphics* 18: 207–212.

Dollár P and Zitnick CL (2013) Structured forests for fast edge detection. In: *2013 IEEE international conference on computer vision (ICCV)*, pp. 1841–1848.

Ferrari V, Jurie F and Schmid C (2010) From images to shape models for object detection. *International Journal of Computer Vision* 87(3): 284–303.

Ferrari V, Tuytelaars T and Van Gool L (2006) Object detection by contour segment networks. In: *Computer vision–ECCV* 2006, pp. 14–28.

Frintrop S (2006) *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search* (*Lecture Notes in Computer Science*, vol. 3899). New York, NY: Springer.

Gower JC (1975) Generalized Procrustes analysis. *Psychometrika* 40(1): 33–51.

Han J, Shao L, Xu D, et al. (2013) Enhanced computer vision with Microsoft Kinect sensor: A review. *IEEE Transactions on Cybernetics* 43(5): 1318–1334.

Hinterstoisser S, Cagniart C, Ilic S, et al. (2012) Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(5): 876–888.

Hsiao E and Hebert M (2012) Occlusion reasoning for object detection under arbitrary viewpoint. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Jiang H and Yu S (2009) Linear solution to scale and rotation invariant object matching. In: *IEEE conference on computer vision and pattern recognition*, pp. 2474–2481.

Kennedy R, Gallier J and Shi J (2011) Contour cut: Identifying salient contours in images by solving a Hermitian eigenvalue problem. In: *2011 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2065–2072.

Kontschieder P, Riemenschneider H, Donoser M, et al. (2011) Discriminative learning of contour fragments for object detection. In: *Proceedings of the British machine vision conference*.

Leibe B, Leonardis A and Schiele B (2004) Combined object categorization and segmentation with an implicit shape model. In: *Workshop on statistical learning in computer vision, ECCV.*

Leordeanu M, Hebert M and Sukthankar R (2007) Beyond local appearance: Category recognition from pairwise interactions of simple features. In: *2007 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1–8.

Lian W and Zhang L (2010) Rotation invariant non-rigid shape matching in cluttered scenes. In: *Computer vision–ECCV* 2010, pp. 506–518.

Lim JJ, Zitnick CL and Dollár P (2013) Sketch tokens: A learned mid-level representation for contour and object detection. In: *2013 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3158–3165.

Lu C, Latecki LJ, Adluru N, et al. (2009) Shape guided contour grouping with particle filters. In: *2009 IEEE conference on computer vision–ICCV*, pp. 2288–2295.

Mairal J, Bach F, Ponce J, et al. (2008) Discriminative learned dictionaries for local image analysis. In: *2008 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1–8.

Maji S and Malik J (2009) Object detection using a max-margin Hough transform. In: *2009 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1038–1045.

Martin DR, Fowlkes CC and Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(5): 530–549.

Ma T and Latecki LJ (2011) From partial shape matching through local deformation to robust global shape similarity for object detection. In: *2011 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1441–1448.

Ming Y, Li H and He X (2012) Connected contours: A new contour completion model that respects the closure effect. In: *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 829–836.

Navalpakkam V and Itti L (2002) A goal oriented attention guidance model. In: *Proceedings of the 2nd workshop on biologically motivated computer vision (BMCV'02)*, pp. 453–461.

Nishigaki M, Fermuller C and DeMenthon D (2012) The image torque operator: A new tool for mid-level vision. In: *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 502–509.

Ommer B and Malik J (2009) Multi-scale object detection by clustering lines. In: *2009 IEEE conference on computer vision–ICCV*, pp. 484–491.

Opelt A, Pinz A and Zisserman A (2006) A boundary-fragment-model for object detection. In: *Computer vision–ECCV* 2006, pp. 575–588.

Quigley M, Conley K, Gerkey B, et al. (2009) ROS: An open-source robot operating system. In: *ICRA workshop on open source software*.

Ravishankar S, Jain A and Mittal A (2008) Multi-stage contour based detection of deformable objects. In: *Computer vision–ECCV* 2008, pp. 483–496.

Richards W and Hoffman DD (1985) Codon constraints on closed 2D shapes. *Computer Vision, Graphics, and Image Processing* 31(3): 265–281.

Riemenschneider H, Donoser M and Bischof H (2010) Using partial edge contour matches for efficient object category localization. In: *Computer vision–ECCV* 2010, pp. 29–42.

Rusu RB, Blodow N and Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: *IEEE international conference on robotics and automation*, 2009, pp. 3212–3217.

Shotton J, Blake A and Cipolla R (2005) Contour-based learning for object detection. In: *Computer vision–ICCV* 2005, pp. 503–510.

Srinivasan P, Zhu Q and Shi J (2010) Many-to-one contour matching for describing and discriminating object shape. In: *2010 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1673–1680.

Tang S, Wang X, Lv X, et al. (2013) Histogram of oriented normal vectors for object recognition with a depth sensor. In: *Computer vision–ACCV* 2012, pp. 525–538.

Teo CL, Myers A, Fermüller C, et al. (2013) Embedding high-level information into low level vision: Efficient object search in clutter. In: *Proceedings of the 2013 IEEE international conference on robotics and automation, ICRA*, pp. 1–7.

Thayananthan A, Stenger B, Torr PH, et al. (2003) Shape context and chamfer matching in cluttered scenes. In: *2003 IEEE conference on computer vision and pattern recognition (CVPR)*, volume 1 pp. I-127–I-123.

Toshev A, Taskar B and Daniilidis K (2010) Object detection via boundary structure segmentation. In: *2010 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 950–957.

Wang X, Bai X, Ma T, et al. (2012) Fan shape model for object detection. In: *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 151–158.

Xu Y, Quan Y, Zhang Z, et al. (2012) Contour-based recognition. In: *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3402–3409.

Yang S and Wang Y (2007) Rotation invariant shape contexts based on feature-space Fourier transformation. In: *Fourth international conference on image and graphics*, 2007, pp. 575–579.

Yu Y, Mann GKI and Gosine RG (2009) Modeling of top-down object-based attention using probabilistic neural network. In: *CCECE'09*, pp. 533–536.

Appendix: Index to Multimedia Extension

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia

Table of Multimedia Extension

Extension	Media type	Description
1	video	Results of `Mug` detection from the UMD-clutter dataset