A Mid-Level Approach to Contour-based Categorical Object Recognition

Ching L. Teo  Cornelia Fermüller
Yiannis Aloimonos
University of Maryland, Department of Computer Science
College Park, MD 20742
{cteo, fer, yiannis}@umiacs.umd.edu

Abstract
We propose a method of detecting generic classes of objects from their representative contours that can be used by a robot with vision to find objects in cluttered environments. Since contours alone are insufficient to indicate the presence of an object, our approach focuses on the use of mid-level visual operators to extract such relevant information. We first apply a recently mid-level operator called the image torque to extract likely fixation locations of objects. We then use the operator’s output to create a novel contour-based descriptor that extends the shape-context descriptor to include boundary ownership information and is rotationally invariant. Next, the new descriptor is used in a multi-scale matching approach that modulates the torque operator towards the target, indicating its location and size. Compared to 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 our method. Experimental results over three challenging datasets with significant occlusions, clutter, viewpoint and orientation changes show comparable performance with current state of the art that uses richer image representations for contour grouping and matching, while handling such challenging situations effectively.

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 what is popularly known as mid-level cues following Marr’s description of visual perception Marr [1976]. Common mid-level cues include texture, corners, edges and object contours and we
Figure 1: From mid-level contour grouping to target object recognition. (I) 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. (II) 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. (III) Target object detection: regrouping scored contours using the same mid-level grouping strategy reveals locations, scales and supporting contours of the target object.
will focus on using edges and contours here. Motivated by the speed and ease of several biologically inspired robotic visual perception systems that uses attention as a basis to reduce the visual search space for object detection and scene understanding (e.g. Frintrop [2006]; Navalpakkam and Itti [2002]; Yu et al. [2009]), we have developed a mid-level contour grouping mechanism that first determines fixation points corresponding to potential object locations (Fig. 1(I)). Each fixation point corresponds to a set of (almost) closed contours that, by the Gestaltist notion of similar fate, 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. For this work, we extend the popular shape-context (SC) descriptor of Belongie et al. [2002] with additional fixation information of the objects to create a new descriptor that is discriminative to ambiguous edge fragments due to matching of partial contour pieces. By applying the Fourier Transform over the angular components of the descriptor, we are able to handle changes in translation, scale and rotation (Fig. 1(II)). 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 similar and share a common fate for object-like contours at the same time (Fig. 1(III)).

The main advantages of using contour information for recognition in robotic vision are that they are: 1) extremely easy to obtain and process (using standard edge detectors) and 2) robust against changes in lighting in comparison to other appearance based cues (e.g. color and texture). In addition, because we are interested in recognizing categories of similarly shaped objects – we generalize better across objects which share certain common shapes and functionality as well. This has important implications when the robot is tasked to search for objects based on descriptions of shape or functionality, or suggest plausible alternatives when the actual target is not present. The main drawback of using 2D contour based information alone 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 shapes properties in cluttered environments.

The input is a 2D image, and we are interested in the detection of the contours that correspond to the target object class – e.g. a Hammer class in the Hand-Manipulation dataset or a Bottle class in the ETHZ-Shapes dataset (sec. 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 (Fig. 2):
Figure 2: Illustration of the challenges of contour-based categorical object recognition. (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.

1) Inaccurate and noisy (broken) edges. Since edge detection in 2D images depends inherently on the local intensity gradients, noise during the image formation process would inadvertently result in edges that are either inaccurate, incomplete or missing. One common way of resolving this issue is to first attempt to group pieces of contours together, using measures of saliency and Gestalt principles of edge continuation, for e.g. in Kennedy et al. [2011]; 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 (Fig. 2(b)), which makes it more discriminative. The key challenge is that ownership determination requires the objects (or at least the presence of an object) to be first detected – a paradoxical 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 similarly broken and fragmented in the image. 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 Riemenschneider et al. [2010]; 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 recogniz-
ing objects from edges alone is a very difficult task. 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. In particular, our contributions are threefold: 1) We demonstrate the usefulness of a recently introduced mid-level operator, termed the image torque Nishigaki et al. [2012], that was originally designed to be used in a bottom-up processing to detect the presence of regions likely to contain objects (we call them proto-objects) in the image by computing a measure of contour completion (sec. 3.1). 2) By integrating information derived from torque with the shape-context descriptor, we introduce mid-level information into a novel descriptor that encodes boundary ownership information leading to better matching of partial edge fragments. To make the descriptor invariant to rotation, the Fourier Transform is applied over the angular bins (sec. 3.2). 3) Finally, using the matched contours, we show how one can modulate the torque operator in a multi-scale manner so that it becomes sensitive to the target object model (sec. 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 it becomes more amenable for classification. Ferrari et al. [2006] approximates them with straight adjacent fragments for part-based matching. Similarly, Ravishankar et al. [2008] uses 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 encodes statistically 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] uses simple features based on orientations and pairwise interactions to create a local descriptor for matching. Srinivasan et al. [2010] views 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 in Riemenschneider et al. [2010] and used in a discriminative framework in 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 SVM with a hough voting approach to detect object specific contours.

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. Although the approach was able to localize objects effectively from 2.5D data, a major shortcoming is that one needs to search over a potentially large number of poses before one can modulate the mid-level operator appropriately. Also, the approach only works with closed, simple shape contours (e.g. no interior holes) which limits the kinds of objects that can be recognized.

These different approaches share several common characteristics. First, in order to overcome the noise and clutter that exists 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. These two steps, however, in the approaches surveyed above, are independent, and their performance can depend widely on the effectiveness of either steps. 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 are rotationally invariant, 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 is 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 Fig. 3. Prior to detection, a mid-level representation of the target model contours is obtained from annotated ground truth of the training set (Fig. 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 one 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 curvatures. Each codon from the training set is then represented as a set of B-splines and we apply EM clustering over the spline coefficients to recover the set of \( u \) model codons, \( \{w_1, \cdots, w_u\} \), which are arranged in the order they appear on the contour in a clockwise manner. 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 \( k \) model codons of increasing length, \( C_m = \{M_1 \cdots M_k\}, |M_p| < |M_{p+1}| \) until the entire contour is accounted for.
Figure 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 + Pb 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 $\tau_{p,q}$. 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.
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*width) using the Pb edge detector Martin et al. [2004] (Fig. 3(b-1)). We detail the remaining steps in the sections that follow (Fig. 3(c)-(d)). First, we review the image torque and how it functions as an edge grouping mechanism that discovers the contours and location of proto-objects. 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.

3.1 Contour completion using image torque

![Figure 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 $\theta_{pq}$ is the angle between $\vec{r}_{pq}$ and $\vec{F}_q$. Modified from Nishigaki et al. [2012].](image)

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). This measure of edge completion is computed via the cross-product between the edge pixel and a vector from the center point to the edge pixel as shown in Fig. 4. Formally, the value of torque $^1\tau_{pq}$ of an edge pixel $q$ within a discrete image patch with center $p$ is defined as:

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

where $\vec{r}_{pq}$ is the displacement vector from $p$ to $q$ and $\vec{F}_q$ is the tangent vector at $q$.

$^1$Since torque is a vector that is always perpendicular to the image plane, we drop the vector notation for simplicity.
\( \vec{F}_q \) can be viewed as a “force” vector\(^2\) in the image space that can be associated with the relative importance of a particular edge pixel (see (3)). 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:

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

We compute (2) over multiple scales for every image point, and we extract the largest \( \tau_P \) over all scales to create a two-dimensional torque value map. The extrema in the value map indicate locations in the image that are likely centers of closed contours (crosses in Fig. 3(b-2)), denoted as \( \mathcal{P}_c \), and we consider only the top \(|\mathcal{P}_c| = 20\) proto-objects with the largest \( \tau_P \). For each extrema center, \( p_c \in \mathcal{P}_c \), we can also compute the torque contribution per edge pixel, \( \tau_{p_c,q_i} \), via (1). By setting a threshold \( t_c \) on the torque contribution, we denote the remaining set of \( n \) edge pixels with \( \tau_{p_c,q_i} > t_c \) to be \( Q_{p_c} = \{ q_i \}, i \in [1, \cdots, n] \) (shown as selected contours in Fig. 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 \( \tau_P \) 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 extrema point is a good estimate of the object’s scale.

The image torque, however, is a purely bottom-up procedure: it detects potential proto-object locations, \( p_c \) and supporting contours, \( Q_{p_c} \), with no preference towards any particular object class. We show in the next two sections that by integrating these 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, \( \tau_{pq} \), between an edge pixel \( q \) and the associated center pixel \( p \). Since \( \vec{r}_{pq} \) is fixed (edges are fixed in a 2D image),

\(^2\)The sign of \( \tau_{pq} \) depends on the direction of the tangent vector and for this work, we compute the direction based on the sign of the image gradient.
one way to modify $\tau_{pq}$ is to change the weight on $\vec{F}_q$ as follows:

$$\tau_{pq}^m = \vec{r}_{pq} \times f(\vec{F}_q)$$

(3)

where $f(\cdot)$ can be any function that modifies $\vec{F}_q$ appropriately. In this work, we define $f(\cdot)$ 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 this and the next section how the final form of $\tau_{pq}^m$ in (10) 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, $Q_{pc} = \{q_1, \cdots, q_n\}$, the shape context descriptor, $h_i$ is defined as a coarse histogram for each $q_i$ with respect to the remaining $n-1$ points:

$$h_i(k) = \#\{q_j \neq q_i : (q_j - q_i) \in \text{bin}(k)\}, j \neq i$$

(4)

where bin$(k)$ denotes a bin in the histogram in log-polar space centered over the $i^{th}$ edge point. This descriptor is tolerant against small localized deformations (due to the histogramming of contour counts), scale and translation invariant.

Figure 5: Construction of the Torque Shape Context descriptor. (Left): Integrating torque with shape-context: Selected codon highlighted with respective $p_{pc}$, $\vec{r}_{pq}$ and $\theta_{pq}$ from torque. $h_i^{\tau}$ is thus defined by additional counts of the angular bins where $\vec{r}_{pq}$ intersects the angular bins (oriented along $O_g$, see Fig. 7). Red means more counts, gray means no counts. (Right): Contour fragments with different $p_{pc}$ are considered different, improving discrimination in ambiguous situations.

However, when the descriptor is applied on contour fragments, $Q'_{pc} \subseteq Q_{pc}$ by breaking them up into codons there will be some ambiguous edge fragments that
can be matched to object contour fragments of different 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. However, object support information is available from torque via the associated centers \( p_c \) and \( \vec{r}_{pq} \). We therefore augment the angular bins of the shape-context histogram that matches with \( \theta_{pq} \), the angle between \( \vec{F}_q \) and \( \vec{r}_{pq} \) (Fig. 5(left)) to form the torque shape-context defined by:

\[
h^\tau_i(k) = K(\angle \text{bin}(k) \equiv \theta_{pq})
\]

(5)

where \( K \) is a discrete truncated Gaussian kernel that adds in more counts to angular bins that correspond to \( \theta_{pq} \), with no counts for angular bins that are not pointing towards \( p_c \). This creates a histogram that encodes a rough direction of where the object center should be. This is equivalent to encoding which side of the edge pixels belongs to or is “owned” by the object. By encoding this boundary ownership information, ambiguous looking boundaries that have different \( p_c \)s would be dissimilar (Fig. 5(right)), which improves discrimination in cluttered situations. We illustrate the advantages of using boundary ownership information in Fig. 6 where it enables us to 1) distinguish between ambiguous contour fragments with similar shape contexts but different \( p_c \) and 2) perform partial contour matching under occlusion and clutter.

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 reduce significantly the discriminatory power of the descriptor due to the fact that tangents are easily corrupted by noise and discretization effects. Instead, we again use the mid-level information obtained from torque to obtain a shape-context descriptor located at the torque centers. Using only the angular bins, we apply the Fast Fourier Transform (FFT) over the normalized bin counts so as to perform a frequency analysis of the angular signals of the target \( G(p) = \mathcal{F}[g(n)] \) and the model \( M(p) = \mathcal{F}[m(n)] \) (Fig. 7(Middle row)). The key idea is that, under ideal conditions, for a target that is rotated \( \delta^\circ \) with respect to the model, we can detect the equivalent lag of \( \delta^\circ \) in \( G(p) \) when we compute cross-correlation (Fig. 7(Bottom row)) with the FFT of the model’s angular bins \( M(p) \):

\[
\Delta(p) = G(p)M(p)^*
\]

(6)

where \( * \) represents the complex conjugate of \( M(p) \). Since both \( g(n) \) and \( m(n) \) are real, \( \Delta(p) \), which contains the lags at different frequency components will

\[ \text{that is, greater than } \pi \text{ rad} \]
Figure 6: (Top) Illustration of how torque shape-context reduces errors in matching with ambiguous partial edge fragments. (a-1) Input image + Pb edge map. (a-2) Selected proto-object center boxed. (b) Comparing the matches to the model codons with (b-1) only 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 edge weights over multiples-scales for the respective two cases. (c-2) with torque shape-context identifies more of the correct edges than (c-1). (Bottom) Illustrating how torque shape-context helps us in detecting partial matches. The saw’s handle at the lower right (boxed) is occluded by the hand (left), but the blade is still correctly detected (right).
Figure 7: Analyzing the Fast Fourier Transform (FFT) over the shape-context’s angular bins located at the torque center in order to obtain rotation invariance via a global orientation estimate, $O_g$. (Columns) Left to Right: (L) Model of Saw, (M and R) Rotated targets at $36^\circ$ and $187^\circ$ with respect to the model. (Top row): Contour points with torque shape-context located at torque center. (Middle row): Torque shape-context angular bin signal (1D). (Bottom row): FFT of all three signals and lags computed after applying cross-correlation. The lags correspond directly to the four possible orientations that the targets may exhibit.
have zeros for its imaginary parts. We then use the component $p$ that has the largest cross-correlation between both signals to obtain the most significant lag component $O_g$ (in degrees) which represents the global orientation transform that we can apply to make the torque shape-context rotationally invariant. Since the two signals are circularly shifted versions of each other, there are four possible orientations (at each quadrant) for the target, and we consider all four orientations when we perform the contour matching (sec. 3.3). Compared to matching over a large number of orientations, this approach dramatically reduces the number of orientation poses to search. We demonstrate the effects of imposing rotational invariance in Fig. 8. One can see that without imposing $O_g$, the target object is not as well detected compared to the case where $O_g$ is used to define the reference frame.

Following Belongie et al. [2002], we use a combination of both shape-context histograms (original shape-context with additional angular bin counts towards $p_c$) and compare using the $\chi^2$ statistic. We use the dynamic programming method of Thayananthan et al. [2003] to compute correspondences $\phi$ by minimizing the overall cost of matching, $C_\phi$ between two edge fragments $Q_{\phi}$ ($p_c$ for

Figure 8: 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.
simplicity in notation) and $M_{p_{c}}$ (a target model codon fragment):

$$C_{\phi}(Q', M) = \gamma_{sc} C_{sc}(Q', M) + \gamma_{\tau} C_{\tau}(Q', M)$$  \hspace{1cm} (7)

where $C_{sc}(\cdot)$ and $C_{\tau}(\cdot)$ are the shape-context matching costs for the shape-context and torque shape-context components (angular bin counts) respectively. We impose $\gamma_{sc} + \gamma_{\tau} = 1$ so that we control the relative importance of these two histograms in influencing the local matching score. Finally from the correspondences, we define the torque shape-context matching distance, $D_{sc}^{\tau}$, as the weighted mean of shape-context matching costs over the $n'$ matched points in $Q'$:

$$D_{sc}^{\tau}(Q', M) = \frac{1}{n'} \sum_{q_i' \in Q', m_{\phi(i)} \in M} (\gamma_{sc} C_{sc}(q_i', m_{\phi(i)}) + \gamma_{\tau} C_{\tau}(q_i', m_{\phi(i)})) .$$  \hspace{1cm} (8)

Since $D_{sc}^{\tau}$ is a local measure of similarity of partial edge fragments, we show in the next section how we use it in a multi-scale approach to develop a mid-level contour matching score function $f(\cdot)$ that is sensitive to the target object class.

### 3.3 Multi-scale matching of supporting contours

![Multi-scale edge matching](image)

Figure 9: Multi-scale edge matching: (a) Detail of $p_{8}$ from Fig. 3(b-2). Neighboring torques $p_{c}$ with their supporting edges (in similar colors) are combined. (b-1) to (b-3) Increasing scales of combining neighboring codons together for matching.

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 Fig. 9. From the torque grouped edges $Q_{pc}$, we first combine neighboring $Q_{rc}$ belonging to nearby centers that fall within the detected bounding box of $p_c$ to form a larger set of grouped edges $R_{Nc}$ 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 $R_{Nc} = \{r_1, \ldots, r_s\}$ into codon fragments so as to obtain a more compact representation of a set of $d$ codons, $C_e = \{R'_1, \ldots, R'_d\}$. Starting at codon $R'_1$, we progressively select and combine the next $j$ neighboring codons: $\{R'_{1}, \ldots, R'_{1+j}\}$ for comparison with each of the $k$ codons from the model contours: $C_m = \{M_1, \ldots, M_k\}$ by computing $D_{sc}(R'_{1}, \ldots, R'_{1+j}, M_{1}, \ldots, M_{k})$ (8) with a slight abuse of notation. This results in a $W \times H \times j \times k$ matrix of distance scores corresponding to each combination. This process is repeated for each of the $d$ codons which gives us a final $W \times H \times (d \times j \times k)$ matrix that records the value of $D_{sc}$ for every edge pixel location in $R_{Nc}$. We then select the smallest $D_{sc}$ across all $d \times j \times k$ levels to yield the final distance score for each $r_i$ denoted as a 2D torque shape-context distance map, $E_{D_{sc}}$. This is repeated over all four possible global orientations $O_g$ described in the preceding section and we select the orientation that yields the smallest $E_{D_{sc}}$. A note on the computational complexity of this step. Since $d$ and $k$ are small (typically, 15 and 6) and we set $j$ 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 $E_{D_{sc}}$ to a normalized weight we use an exponential function:

$$W_{D_{sc}}(r_i) = \beta_c + \beta_f(\exp(-r_i/(2\sigma)))$$

(9)

where $\beta_c, \beta_f, \sigma$ 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 (9) to scale $\vec{F}_q$, we obtain the modulated image torque that is sensitive to the target object class:

$$\tau_{pq}^m = \vec{r}_{pq} \times (W_{D_{sc}}(q) \ast \vec{F}_q)$$

(10)

$^4$the codons are indexed in a clockwise direction.
where \( f(\vec{F}_q) = W_{\tau sc}^D(q) \ast \vec{F}_q \) as in (3). 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 \( W_{\tau sc}^D \) 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.

4 Experiments

We perform experiments over three datasets. The first one, termed the UMD Hand-Manipulation dataset, is collected by a mobile robot observing different sets of humans performing manipulation activities using various tools and objects. This dataset is challenging because the hands, tools and objects induce occlusions, clutter and projective deformations (translation, scale and rotation) which are typical of manipulation activities. The goal is to show that our approach can handle such situations reliably. To further highlight that our approach compares well with other state of the art contour-based object recognition approaches, we use the standard 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 our approach directly 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.

For all three experiments, we use the following meta-parameters in our proposed method: \( \gamma_{sc} = \gamma_{e} = 0.5, \beta_{e} = 0.05, \beta_{f} = 0.95, \sigma = 0.05, t_{c} \), the threshold to select the strongest edges is set to the top 50 percentile of all grouped edges. The number of codons neighbors to combine, \( j \), is set to 3 for all object categories except Giraffes and Swans (from the ETHZ-Shapes dataset), which has \( j = 5 \) so as to fully account for long thin structures (neck and legs) that are common in these two categories. For computing torque, we search over image patches with sizes ranging from 3 pixels to a quarter of the input image height and width.

For evaluating object detection performance, we admit a true positive using the PASCAL criteria: 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 the 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, \( \tau_P^m \) (replacing \( \tau_{pq} \) in (2) with \( \tau_{pq}^m \) from (3)) with \( \tau_P \).
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 6 video sequences (around 1500 frames each) of 3 different human subjects constructing a partial wooden frame using 5 tool classes: \{Borer, Hammer, Ruler, Saw, Screwdriver\}. This dataset is challenging because it contains more clutter and occlusion and orientation changes than ETHZ-Shapes (next section), due to the hands and active nature of the frame making process. The goals are to show: 1) that our approach is able to generalize from learned models to other scenarios – a key advantage of contour-based approaches and 2) that our approach is able to handle clutter and partial occlusions under various viewpoints/orientations.

We used the meta-parameters and evaluation procedure as indicated above. For obtaining the target model codons, we used the initial first ten frames and hand annotated the target tool’s contours to obtain the model codons. We then evaluate the rest of the sequence at sample intervals of 10 frames each, which yields a total of around 800 evaluated frames in the entire dataset. We show some results from sample frames of the dataset in Fig. 10: final edge weights $W_{D_{\tau}}$ and the predicted target objects with centers marked as crosses.

For evaluation, we report the Precision/Recall rates and corresponding False Positives Per Image (FPPI) vs. Detection rates (DR). The results are summarized in Fig. 11 for all 5 tools considered. As one can see from the results, our approach is able to localize the target objects in clutter with Average Precision (AP) ranging from 0.77 to 0.89, which is consistent with the results reported for the ETHZ-Shapes dataset (see next section). The detection rates at the standard 0.3/0.4 FPPI range from 0.74 to 1.00 which is on par with current object recognition approaches. The best detection comes from Saw and Hammer, and it is probably due to the fact that the contours belonging to these two classes are very unique (and hence easy for discrimination) compared to other tools. The worst results (in terms of AP) 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 decrease in performance of Borer (and to a large extent Screwdriver as well) highlights one of the key shortcomings of the approach: the mid-level groupings over multiple scales does not capture enough global information of parts and their relationships to accurately separate out objects that consists of a subset of contours from other targets. An extreme example is that of the Marker class, which we have not considered but consists only of two parallel contours as shown in Fig. 12 (left). Due to the small number of contours in the model, such a configuration is easily confused in clutter (Fig. 12 (right)). This result points to future work that
Figure 10: 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: $W_{\text{Dr}_c}$ where red means higher values and target model contours at top-right, Middle: Modulated torque showing the top 2 object detections (red and green crosses), Right: RGB frames overlaid with detections results. Note that for Hammer and Saw, the objects are partially occluded by the hands.
Figure 11: (Top-Left) Precision/Recall curves over the 6 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 5 tool categories.

<table>
<thead>
<tr>
<th></th>
<th>Borer</th>
<th>Hammer</th>
<th>Ruler</th>
<th>Saw</th>
<th>Screwdriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.77</td>
<td>0.82</td>
<td>0.80</td>
<td>0.89</td>
<td>0.79</td>
</tr>
<tr>
<td>0.3/0.4 FPPI</td>
<td>0.74/0.90</td>
<td>0.94/1.00</td>
<td>0.88/0.88</td>
<td>0.95/1.00</td>
<td>0.78/0.83</td>
</tr>
</tbody>
</table>

Figure 12: 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): e.g. the handle of the hammer or the sides of a tape.
should incorporate more *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 ETHZ-Shapes dataset

We further evaluate our proposed mid-level approach of detecting class specific object contours using torque over the ETHZ-Shapes dataset. This dataset is divided into five object categories: \{Applelogos, Bottles, Giraffes, Mugs, Swans\}, with 255 images containing instances of the objects in various background, clutter, scale 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.

<table>
<thead>
<tr>
<th></th>
<th>Applelogos</th>
<th>Bottles</th>
<th>Giraffes</th>
<th>Mugs</th>
<th>Swans</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our method</strong></td>
<td>0.917</td>
<td>0.931</td>
<td>0.796</td>
<td>0.888</td>
<td>0.891</td>
<td>0.885</td>
</tr>
<tr>
<td>Maji and Malik [2009]</td>
<td>0.869</td>
<td>0.724</td>
<td>0.742</td>
<td>0.806</td>
<td>0.716</td>
<td>0.771</td>
</tr>
<tr>
<td>Srinivasan <em>et al.</em> [2010]</td>
<td>0.845</td>
<td>0.916</td>
<td>0.787</td>
<td>0.888</td>
<td>0.922</td>
<td>0.872</td>
</tr>
<tr>
<td>Ma and Latecki [2011]</td>
<td>0.881</td>
<td>0.920</td>
<td>0.756</td>
<td>0.868</td>
<td>0.959</td>
<td>0.877</td>
</tr>
<tr>
<td>Wang <em>et al.</em> [2012]</td>
<td>0.866</td>
<td><strong>0.975</strong></td>
<td><strong>0.832</strong></td>
<td>0.843</td>
<td>0.828</td>
<td>0.869</td>
</tr>
</tbody>
</table>

Table 1: Comparing interpolated average precision (AP) with the proposed method over the ETHZ-Shapes dataset.

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

Table 2: Comparing detection rates at 0.3/0.4 FPPI over the ETHZ-Shapes dataset.
Figure 13: Precision/Recall curves comparing Maji and Malik [2009]; Srinivasan et al. [2010]; Ma and Latecki [2011]; Wang et al. [2012] to the proposed method over the ETHZ-Shapes dataset.
Figure 14: Comparison of DR/FPPI curves over the ETHZ-Shapes dataset.
We focus our 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 (PR) curves of these methods and their interpolated average precision (AP) are compared with the proposed method in Fig. 13 and Table. 1 respectively. Within all five categories, the proposed approach is able to produce very comparable results with other state of the art approaches – with the largest improvement in Applelogos. Averaged over all 5 categories, our approach is able to achieve the overall best mean AP among the compared methods, with a small improvement over Ma and Latecki [2011].

In addition, we plot the false positives per image (FPPI) vs. the Detection Rate (DR) in Fig. 14. The detection rates at 0.3 and 0.4 FPPI are compared with several reported results in the literature in Table. 2. The detection performance at these two levels are consistently on par with the state of the art, with the largest improvements in Applelogos and Giraffes. We show in Fig. 15 some example results: the modulated torque with the final detections, and some failure cases. Similar to the discussion in the preceding section, these cases occur due to the fact 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.

Figure 15: Some example detection results with their modulated torque. Edges show values of $W_{D^{sc}}$, 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 are shown in colored boxes. Best viewed in color.
4.3 Object recognition in clutter on a mobile robot

We demonstrate the feasibility of our approach for practical robotic applications on our mobile robot platform (Fig. 16). The robot consists of the Adept Pioneer P3-DX base together with a custom made frame on which a Kinect RGB-Depth is attached via a Directed Perception PTU-D46 pan-tilt unit (PTU). The robot’s software runs over the Robot Operating System (ROS) with appropriate interfaces implemented to send the Kinect sensor data (only the RGB) to Matlab for processing by the proposed method. The robot is tasked to perform random movements using either the base or the PTU while observing a table that contains cluttered objects. The goal is to detect the presence of Mug-like objects in clutter while inducing changes in viewpoint and occlusion from the movements. We performed three different collections of the Kinect RGB-Depth data with differing amount of clutter per dataset, with around 1000 frames per sequence. For the target model, we used the same Mug codons learned from the ETHZ-Shapes dataset, together with the same meta-parameters described above. For this dataset, we evaluated frames at intervals of 10 frames, yielding around 300 frames that are considered for evaluation.

We show some results from sample frames of the dataset in Fig. 17: final edge weights $W_{D_{sc}}$ and the predicted objects with centers marked as crosses. The su-
Figure 17: Sample detection results for mugs from four frames of the robot collected dataset. Top row: $W_{D_{gt}}$, where red means higher values. Middle row: Modulated torque showing the top 2 object detections (red and green crosses). Bottom row: RGB frames overlaid with the detection results. Frames 3 and 4 show detections of mugs under partial occlusion.

changes in viewpoints (mainly scale and translation) and occlusions as the robot moves, our approach is able to consistently detect the presence of multiple instances of mugs on the table. Equally important is the fact that the models were obtained from an entirely different dataset, which highlights the generalizability of a contour-based object recognition approach. By counting the number of times the approach returns the correct locations (with greater than 50% overlap) with the groundtruth, the approach achieves a detection rate of nearly 78% over the entire dataset. The main limitation with this setup is the fact that Matlab over ROS is not optimized. Optimizing the code and making it available for use in ROS is currently part of our future work.
5 Summary and Future Work

We have presented a new mid-level approach of contour-based categorical object recognition that exploits several useful properties of the image torque. Firstly, we use the torque’s intrinsic edge grouping mechanism to find initial proto-object locations and their supporting edges. Using these proto-centers, we augment the shape-context descriptor in two ways: 1) with boundary ownership information for matching in clutter and 2) applying FFT over the angular bins to estimate a global orientation transform for rotational invariance. Finally, by compactly representing the extracted edges using codons of varying lengths, we describe a multi-scale approach of matching partial supporting edges with the model that modulates the torque operator towards the target object class. We evaluated the approach over three challenging datasets: 1) the newly introduced UMD Hand-Manipulation dataset, 2) the ETHZ-Shapes dataset and 3) a dataset collected by a moving robot observing a table with clutter, that highlights the approach ability to handle occlusions, partial matches and orientation changes over large variations in environmental conditions which makes it suitable for practical robotic applications.

This work has shown only but one possibility of using the image torque in an interesting way. We plan to investigate how one can tune the torque operator based on more generic mid to high-level description of the target category – e.g. measures of elongatedness, ratio between principle axis etc. instead of a shape outline used here. In addition, we are investigating how one can incorporate higher-level grouping information of parts of objects to improve the performance of the approach for targets that share significant amount of contours.

6 Acknowledgements

The support of the European Union under the grant Poeticon++ in the Cognitive Systems program, and the support of the National Science foundation under an INSPIRE grant in the Science of Learning Activities program and a grant in the Cyberphysical Systems program are gratefully acknowledged.

References


