ABSTRACT

Title of dissertation: COMPUTATIONAL MID-LEVEL VISION: FROM BORDER OWNERSHIP TO CATEGORICAL OBJECT RECOGNITION

Ching Lik Teo, Doctor of Philosophy, 2015

Dissertation directed by: Professor Yiannis Aloimonos
Department of Computer Science

Since it was proposed in 1890 by Christian von Ehrenfels, Gestalt psychology has remained a key school of thought that explains how one perceives the world ("the whole") from the sum of its individual components ("the parts") or processes. These processes are aptly summarized in the well known "Rules of Gestalt". In spite of its influence in other fields, the empirical nature of Gestalt rules impedes their widespread adoption in Computer Science. This thesis serves to bridge this apparent divide by making *Mid-level Vision*, or Computer Vision based on Gestalt rules, not only computationally feasible but also practical for real applications. We address the general problem of *figure-ground organization*, where the goal is to separate the foreground (or object) from the background. To do this, we first formulate a fast approach that pairs Structured Random Forests (SRFs) with Gestalt-like features, for both boundary detection and border ownership assignment. We then show how border ownership information is useful for shape-based recognition of object categories. This is done by embedding ownership information into the *image torque*, a grouping operator that detects closure patterns in the image edge, so that we

modulate the operator in an efficient manner for detecting class-specific contours in clutter and occlusion. Next, we show how *symmetry*, an important shape-based regularity in Gestalt psychology, can be detected in clutter and be used for guiding segmentation of symmetric foreground regions. Besides shape and symmetry, *functionality* is another important mid-level cue that supports categorical object recognition. Based on Gibson's principle of affordance, we introduce a fast technique based on a SRF trained with geometric features that provides pixel-accurate affordances of tool parts. Finally, we describe as future work how language can be exploited to "activate" such mid-level processes so that a joint semantic space can be obtained for linking visual concepts to language to solve even more challenging problems in Computer Vision, effectively reducing the so-called "semantic gap" between these two related domains.

COMPUTATIONAL MID-LEVEL VISION: FROM BORDER OWNERSHIP TO CATEGORICAL OBJECT RECOGNITION

by

Ching Lik Teo

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Yiannis Aloimonos, Chair/Advisor
Dr. Cornelia Fermüller, Co-Advisor
Professor David Jacobs
Professor Donald Perlis
Professor Timothy Horiuchi, Dean's representative

Acknowledgments

This thesis would not have been possible without the generous guidance and support of my advisors: Prof. Yiannis Aloimonos and Dr. Cornelia Fermüller, during my 5 years with the Computer Vision Lab. I am grateful for the opportunity and time spent discussing these works (and more), many of which came to form the basis of this dissertation.

I would like to thank my colleagues as well: F. Barranco, A. Ecins, A. Myers and Y. Yang for many thoughtful discussions, encouragements and fun times that make life in the lab more interesting and memorable.

Finally, I am grateful for the sacrifices and support from my wife, my daughter and family for all the weekends and nights spent working for another paper deadline or demo. This thesis is dedicated to you all.

There are many other people whom I cannot fully list who have influenced me during my time here in Maryland. To them all, thank you.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

This thesis proposes several techniques that invokes *Mid-Level Vision* as a central paradigm to solve related problems in Computer Vision. These problems can be broadly grouped into the area of *figure-ground organization*, where the goal is to determine, given a 2D image, which parts of the image belong to the foreground (or object) and which parts belong to the background. Numerous tasks in Computer Vision directly or indirectly use figure-ground organization: e.g. scene understanding [294], object proposals [3] and occlusion boundaries detection [109]. In this chapter, we will introduce the problem and explain its importance in linking high-level (semantic) information with low-level visual signals. Next, we argue for the use of mid-level vision from both biological and computational perspectives to efficiently solve this problem. Finally, we survey previous works in related areas and show how this thesis contributes in advancing the state-of-the-art.

1.1 Figure-ground Organization and Bridging the Semantic Gap

Look at the images in Fig. 1.1. Input scenes are shown on the left with predictions of parts of the scene on the right showing the names (labels) assigned to individual segments. These images illustrate the process of *scene understanding*,

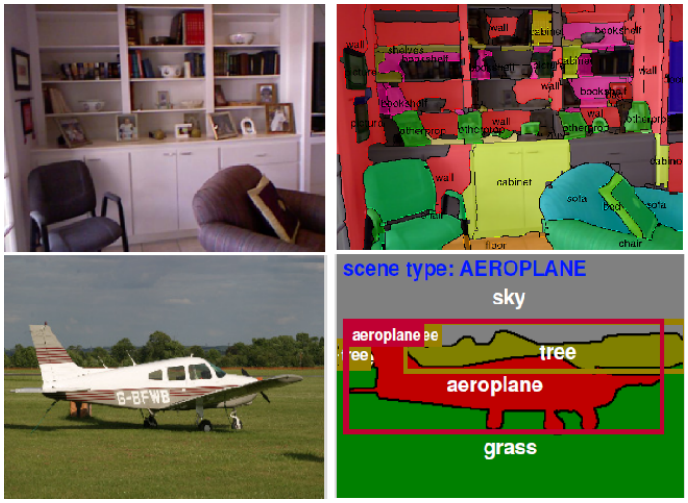


Figure 1.1: A typical scene understanding task. (Left) Input images and (right) predicted results from [93] (above) and [301] (below).

that is, the assignment of semantic labels to scene parts. Although easy for humans, it remains a difficult problem in Computer Vision as many sub-processes remain unresolved. These include (ordered in terms of increasing complexity): 1) boundary detection, 2) foreground-background segmentation, 3) object detection and recognition , 4) semantic segmentation of objects and scene entities and 5) perceptual organization. These problems are often addressed individually in Computer Vision with different approaches (and names) e.g. [8,62,113,193]. Their main goal, however, is the same: to mimic the capabilities of humans in understanding and perceiving the visual world by assigning semantic labels (names) to their corresponding parts. The exact processes involved in converting visual signals to a visual percept of the world is still an open (and complex) research problem. Behavioral and neurological studies from primates, however, have indicated that the process of *figure-ground organization* (FGO) (processes 1-3) provides important input for high-level visual

processes [43, 194, 306] (processes 4 and 5). Recent neurological findings (fMRI and EEG) have also revealed feedback connections between the visual cortex and higher brain areas involved in memory and words [57, 58, 90]. These connections are most active when participants are tasked to describe an image or a moving video, and are co-located with regions of high saliency.

These findings and observations show that the FGO problem is central for linking high-level knowledge (e.g. words, semantic labels, relationships etc.) of objects, scenes and other entities in the world with low-level visual signals (e.g. edges, pixels etc.) for achieving a visual percept (understanding) of the world. In other words, solving the FGO problem bridges the so-called "semantic gap" or "pixels-to-predicates" [96, 186, 289] problem, which is well known in the field of Artificial Intelligence. This is also the main underlying goal that drives and links up the different works in this thesis.

Central to the FGO problem is how one begins the actual process of selecting the "pixels" so that meaning or predicates can be appropriately assigned to them. In this thesis, we draw inspiration from the field of Gestalt psychology, and propose to use "Mid-Level Vision" in our approaches. We introduce these ideas in the next section.

1.2 Mid-Level Vision for Figure-ground Organization

"Mid-level Vision" is a computational paradigm that exploits the use of so-called "mid-level" visual cues to guide certain visual processes. Such cues are derived

or inspired from Gestalt psychology, proposed in 1890 by von Ehrenfels which lead to several seminal works and branches of thought notably by Wertheimer (1912) and Köhler (1920). The basis idea is intuitive: from a set of Gestalt principles that combines information from low-level visual signals, we are able to perceive the world in all its complexity and nuances [135]. These principles or "rules of Gestalt" cover a wide area: 1) shape, 2) proximity, 3) motion, 4) symmetry and 5) common-fate to name a few. Since it deals with how such cues can be derived from low-level signals to guide higher-level visual processes such as recognition or semantic understanding, the term "mid-level" vision was used by several works [25, 55, 66, 138, 209]. The FGO problem, as noted in a recent survey [285], is a central problem in Gestalt psychology that uses mid-level cues: convexity, symmetry, lower-region, extremal edges, motion synchrony etc. In this thesis, we show further that it is possible to compute efficiently a subset of such cues via mid-level *operators* and to use these cues for distinguishing the *ownership* of an occlusion boundary. In the next two sections, we motivate the use of mid-level vision as a paradigm for solving the FGO problem from both biological and computational perspectives.

1.2.1 Biological Motivations: Psychological and Neurological Evidences

The earliest evidence for Gestalt in FGO comes from the famous "vase-face" illusion of Rubin (1915) (Fig. 1.2 (left)), where depending on the viewer's interpretation, the percept shifts between two faces and a central vase in the image. This switch in interpretation depends on which side the boundary is perceived to be

Figure 1.2: Visual illusions demonstrating Gestalt principles. (Left) Switching of the "vase-face" percept of Rubin (1915). (Right) Illusionary contours of Kanizsa (1976) [120].

"owned" by the figure (foreground) and points to the existence of a neural circuitry that encodes this ownership information. This process is known in the literature as *border ownership* assignment (BOWN). Related to BOWN is the process known as *contour continuation* (CCONT), which is made famous by Kanizsa's illusionary contours [120] that complete an otherwise incomplete shape (Fig. 1.2 (right)). The importance of BOWN and CCONT for FGO is further demonstrated by Bregman (1981) [30] where a figure, initially showing only some unrelated fragments changes instantly to recognizable letters with the simple addition of a dark occluding foreground (Fig. 1.3).

Although psychological evidences supporting BOWN and CCONT processes are plentiful [99, 130, 207, 215, 225], it was from recordings of the primate visual cortex that demonstrate the existence of specific neurons that encode BOWN and CCONT. von der Heydt et al. [284] showed that the V2 primate visual cortex responds to specific illusionary contours stimuli. This was followed by several models of neural circuitry [103, 223] based on the suggestion that cells in V2 are responsible

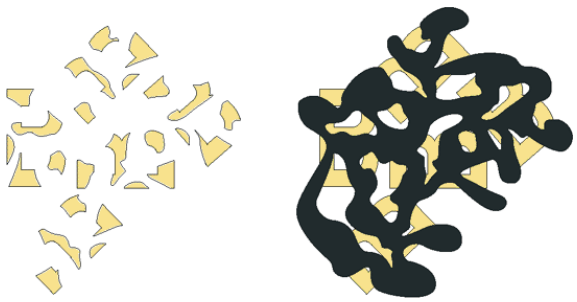


Figure 1.3: Bregman's illusion (1981) [30]. Indiscernible fragments in the background instantly appear as the letters 'B' with the addition of a dark foreground.

for encoding *occlusion* boundaries, that is, regions where objects meet with one another or when objects meet with the background. The neural mechanisms of BOWN were discovered by Zhou et al. [308] that showed that V2 and V4 (and to a lesser extend V1) cells encode border ownership selectively. Very recent works showed that such cells encode not only ownership but also depth ordering [226] within a very short amount of time [262] (75ms of the onset of the stimulus). There are also studies that provide neurological evidence for the interplay of BOWN and CCONT for FGO, where feedback from higher visual areas guides the process of *context integration* [43, 117]: the combination of cues indicative of ownership (usually far [306]) and continuity (which are more local [102]).

1.2.2 Computational Motivations: Gestalt in Computer Vision

Several computational models for FGO, based on the biological and neurological evidences presented above, have been proposed by several authors [37, 75, 147, 200, 219, 257, 291]. One of the most recent (and complete) models is suggested by Kogo et al. [136], that uses local cues suggestive of ordinal depth to determine own-

ership along occlusion boundaries and allow localized spreading to simulate illusionary contours. Besides these biologically motivated models which seek to mimic the neural circuitry, we describe here other computational models that embed Gestalt principles in Computer Vision related tasks.

Most of the works in Computer Vision dealt with embedding the Gestalt principle of proximity, good continuation and similarity by using an explicit or implicit Markovian assumption, modeling the problem as a Markov Random Field (MRF) [79] or (more recently) as a Conditional Random Field (CRF) [151]. Williams and Jacobs [291] used convexity to implement a stochastic field for contour completion, and this was recently extended by [11] using Tangent Bundle Theory. The principles of closure and proximity were explored in a probabilistic contour framework by Elder et al. [59, 60]. The same principles were applied to extracting closed regions by Stahl and Wang [256] using a grouping cost that captured these two principles within a graph optimization framework, which was extended to include the extraction of symmetrical regions [257]. More recently, multiple cues have been employed to extract closed contours from images [202] and extended to extract closed regions from videos [167].

Variational methods such as snakes [124] and level-sets [213] have also been employed with success in capturing salient regions of images and videos displaying different forms of prior information. Cootes et al. [40] introduced active shape models for representing the shape prior in terms of an average shape and basis vectors that account for shape variability. This lead to several other works [31, 277] that embed this model within a variational energy functional. Instead of an average shape,

Rousson and Paragios [239] proposed a novel cost function that constrains an implicit surface to evolve towards a predefined shape prior in a variational framework. Extending this, Cremers et al. [44] further showed that by defining a binary shape functional, embedding the shape prior becomes a convex problem which results in globally optimal solutions.

Scale-space theory [176], image pyramids [32] and CRFs have also been employed to capture longer range contextual relationships. Henkel [104] proposed to group edges in scale-space for the purpose of segmenting coherent regions. Latecki and Lakamper [153] showed improved grouping results by matching multiscaled shape fragments that corresponds to foreground object parts. He et al. [100] introduced multiscale random fields to enforce segmentation labeling consistency. Following this, Cour et al. [42] introduced a novel multiscale spectral clustering technique via graph decomposition so that scale and grouping constraints are enforced to generate coherent segments. Along similar lines, Latecki et al. [154] proposed multiscale random fields with a novel combinatory append operator that enables efficient optimization for the task of detecting class-specific contours in clutter. More recently, Shotton et al. [246] showed improved recognition of object classes by training class-specific local classifiers with multiscale contour fragments via a modified chamfer matching formulation. Kohli et al. [137] proposed a novel set of potential functions that capture even more long range relationships in CRFs and showed improved scene segmentation results in occlusion and clutter. Finally, the recent work of Arbelaez et al. [8] combines both multiscale segmentation and object recognition within a single framework. The approach uses a multiscale spectral clustering technique to

produce segmentation candidates that are then combined in an efficient manner to form reasonable *object proposals*. Object proposals is currently a very popular research area and we defer the discussion of such works to Chapter 2.

1.3 Other Related Works

As related works for specific tasks are discussed in their individual chapters, we focus here on works from two areas in Computer Vision: 1) Scene Understanding and 2) Vision and Language, which are not directly addressed by this thesis but are nonetheless important for solving the problem fully.

1.3.1 Scene Understanding

Scene understanding has remained one of the key open research areas in Computer Vision. Early works define "understanding" as a vision system that achieves two key tasks: 1) segmentation into coherent parts and 2) entity (object or foreground) recognition of each part and the entire scene. Numerous works have been proposed that addressed each task either in combination or together [23, 26, 68, 93, 203, 247, 292, 297] and we review briefly here the most notable works. Olivia and Torralba [210] defined a scene "gist" measure to broadly classify an entire image into several scene types, based on the notion of a "spatial envelope" or an "attribute" of the scene. Along similar lines, using the recently introduced "SUN attributes" dataset, Patterson et al. [222] analyzed and introduced attributes descriptors for scene-based classification. More recently, Zhou et al. [307] used state-

of-the-art deep Convolutional Neural Networks (CNN) to learn attributes-based features for scenes.

Most other works focused on accurate segmentation and labeling of scene entities. Sali and Ullman [242] combined class specific contour fragments using a model-based approach to detect and segment target objects. Similarly, Kumar et al. [149] combined state-of-the-art pictorial structures with MRFs so that object specific segments are extracted from the image. Levin and Weiss [165] extended this work by combining top-down object specific contours to produce object specific bottom-up segmentations by training a CRF that takes into account both contour (high-level) and edges (low-level) information. Cao and Li [33] proposed a generative model by learning latent topic models of multiscale object patches, enabling simultaneous detection and segmentation during inference. More recently, by using a textual model, Li et al. [168] demonstrated a novel generative model that is capable of performing simultaneous segmentation, classification and captioning of unseen images with textual tags (see §1.3.2 for other works that combines language and vision). By embedding responses of pre-trained object detectors as additional potentials into a CRF, Ladicky et al. [150] improved the final scene segmentation compared to using low-level information alone. Yao et al. [301] extended this work by embedding a shape prior potential over superpixel segments to further improve the segmentation results.

Recognizing the scene only by labeling its segmented entities, however, is only a small part of scene understanding [295]. Understanding how these entities relate with each other so that it can be useful, for e.g. an active agent to navigate in the

scene, is an important next step. Several recent works in scene understanding have pursued this research direction that attempts to predict a 3D room layout from 2D images [92, 101, 303]. Other works view the problem as generating an image "parse", similar to approaches in natural language processing (NLP), which we discuss next.

1.3.2 Vision and Language

Although language and vision are completely different modalities, they encode often complementary information that differ only in terms of their semantic content. Language, or text, are often used to describe non-visual and to a lesser extent visual entities in an image or video. A key research problem is how one can leverage on language to improve visual processing (and in NLP, to use vision to improve linguistic processing). Works in the Computer Vision community are mainly focused on using language as a form of contextual information that reduces ambiguous/uncertain information from visual inputs. [84] use high-level contextual knowledge of objects in scenes (position, color, etc.) to induce a visual saliency map that represents the most likely locations of objects in the scene. This work uses high-level information of the target to influence directly (via learned parameters) the weights of the saliency algorithm so that the target is more accurately detected than bottom-up methods. In one of the earliest works, [56] showed how nouns can provide constraints that improve image segmentation. This is done by imposing constraints on the nouns (objects) that are likely to co-exist: e.g. `sky` and `plane` are more likely to exist together than `water` and `bus`. [91] extended this work with the addition of prepositions

to enforce spatial constraints in recognizing objects from segmented images. The work of [4] addresses the object search problem in clutter by encoding predefined relationships on likely occurring locations, object co-occurrences, visual and shape cues into a graphical model that guides a mobile robot in selecting the next place to search. The input is a 2.5D Kinect point cloud and they used a max-margin learning approach over several object classes in two different kinds of environments: office and home. Another interesting work that uses contextual information comes from [160] where they create a "object-graph" that encodes relationships of known object categories together with unfamiliar/unknown categories. The goal is to recognize, in a weak sense, via the similarity of the graphs, the common categories of such unknown objects and perhaps assign a more descriptive label to them.

There has also been several works that integrate linguistic information for the purpose of describing visual scenes/images using natural text, which can also be viewed as a manifestation of the scene understanding process. [13] processed news captions to discover names associated with faces in the images, and [119] extended this work to associate poses detected from images with the verbs in the captions. Both approaches use annotated examples from a limited news caption corpus to learn a joint image-text model so that one can annotate new unknown images with textual information easily. Tu et al. [279] view the of scene understanding as analogous to parsing a sentence in NLP, except that the grammar and entities are visual. More recently, [300] proposed a "image to text" parser that combines noisy detections from visual detectors using learned rules to generate a reasonable textual description of the scene. Along the same lines, [148] constructs a model of a image parse

consisting of objects, their attributes (e.g. color, texture), spatial relationships into a CRF. Inference over the CRF results in the most likely combinations of these components so that a reasonable descriptive paragraph of the image is generated. The work of [65] attempts to "generate" sentences by first learning from a set of human annotated examples, and producing the *same* sentence if both images and sentence share common properties in terms of their triplets: (Nouns-Verbs-Scenes). In another work, [253] views the problem of parsing an image containing super-pixel segmentations within the same framework of parsing a parallel textual description of the image and proposes a recursive neural network (RNN) to model the key components that make up the image and text. By training the network using a structured max-margin learning approach, the model is able to optimally parse both images and text for segmentation in both domains. Very recently, Karpathy and Li [122] introduced a multimodal Recurrent Neural Network that generates a sentence by conditioning the network on objects detected via a separately trained CNN over the input image.

1.4 Contributions of this Thesis

Given the breadth, scope and complexity of the FGO problem, this thesis focuses on a smaller subset of related problems that we believe provide important contributions to a final complete solution. Unlike previous works (§1.2.2) that use Gestalt as a high-level prior containing contextual information, we are motivated from a more biological perspective (§1.2.1) that uses mid-level vision to extract

and organize low-level visual signals before feeding them to higher-level visual areas (e.g. TE, TEO) [146]. Specifically, we propose efficient computational methods that detect Gestalt (cues or representations) from real images and demonstrate their usefulness in higher-level visual tasks: e.g. recognition and segmentation.

Chapter 2 introduces a computationally efficient method for *border ownership* assignment, which is a central subproblem in FGO. Our approach leverages on a state-of-the-art classifier termed the Structured Random Forest (SRF) [140] trained over local and global ownership cues to predict both boundaries and ownership in real-time. We demonstrate the usefulness of detecting border ownership in two areas: 1) Extracting foreground object locations using a mid-level grouping operator, termed the *image torque* [209], that is biased towards closure and 2) layered segmentation of a scene from an event-based camera that mimics the human retina, known as the Dynamic Vision Sensor (DVS) [173].

Next, in Chapter 3, we use border ownership information for enhancing *categorical recognition* of objects/parts that share common contour fragments. This is achieved by embedding a novel shape-based descriptor with ownership information followed by modulating the image torque so that it becomes sensitive to the target contours. Compared to other approaches, we show the advantage of using a mid-level approach for this task in terms of handling clutter, occlusion and noisy contours.

In Chapter 4, we detect *symmetry*, specifically reflection and curved reflection symmetries and use it to extract symmetrical regions in real images. For reflection (bilateral) symmetry detection, we propose to detect, using a fast histogram com-

parison of local edges, potential *symmetry attention* points from which we extract object-centric segments for a more detailed localization of the symmetry axis. For curved reflection symmetry detection, we propose a fast approach by training a SRF classifier sensitive to local symmetry cues. Using the detected symmetries, we embed a symmetry prior into a Markov Random Field (MRF) representation of the image edge so that symmetrical regions can be extracted via graph-cuts [27].

We describe in Chapter 5, a fast approach for detecting part-based functionality, or *affordances* [82] of tools from local geometric features. Affordances can be seen as an innate object-level attribute that generalizes recognition to larger classes of objects (even unseen ones). For this work, we train a SRF paired with such features that provides pixel-accurate predictions of the target affordance.

As was noted earlier in §1.1, the purpose of this thesis is to provide solutions that ultimately bridge the semantic gap. In Chapter 6, we conclude this thesis by suggesting potential research directions that exploit language with approaches presented in the precedent chapters. Specifically, we present ideas that link language and mid-level visual representations in a common canonical space so that the appropriate mid-level concepts: e.g. ownership, symmetry, affordances, can be appropriately activated via linguistic cues.

Chapter 2: Assigning Border Ownership in 2D images

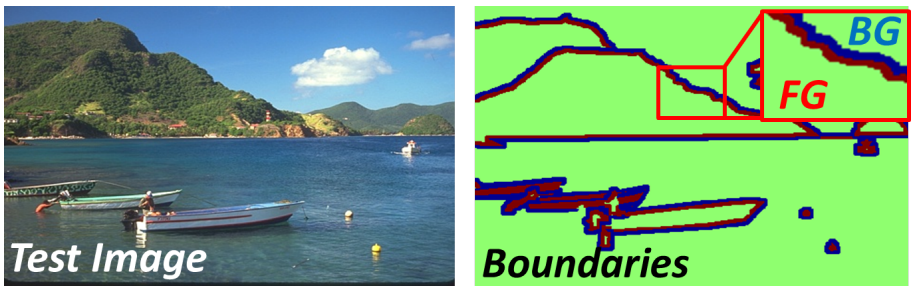


Figure 2.1: Illustrating the border ownership assignment problem. (Left) Input image and (right) boundaries and *border ownership* of foreground (red) and background (blue) regions.

In this chapter, we propose a fast solution for *border ownership* assignment (BOWN)[1], that is, we determine given the input boundaries: places where objects meet with each other or with the background, which "side" of the boundary belongs to the foreground (object) and which side belongs to the background (Fig. 2.1 (left)). As was noted in Chapter 1, cells sensitive to BOWN have been discovered in areas V2 and V4 [308] and they fire within a very short interval [262]. From a computational perspective, BOWN is one of the key *mid-level* processes for solving the figure-ground organization (FGO) problem in that it provides important ordinal depth

[1]This work was published in [268]. Full results, code and videos are available online http://www.umiacs.umd.edu/~cteo/BOWN_SRF/

information and can be regarded as a preprocessing step for higher level tasks such as foreground-background segmentation [238, 283], semantic segmentation [93] and object proposals [34], and is also closely related to selective attention [43]. In spite of this crucial role, BOWN has remained largely ignored by the computational vision community [146], with only two recent works: Ren et al. [231] and Leichter and Lindenbaum [162] proposing computational approaches that address this problem. Unlike these two works that first detect boundaries followed by a separate ownership assignment step, our approach predicts both boundaries and ownership directly from the input RGB image. In addition to state-of-the-art ownership predictions compared to [162, 231] over two datasets: BSDS [193] and the NYU Depth V2 [206], our method runs in *real-time*: $\approx$0.1s for a 320×240 image compared to 15s in [162].

2.1 Introduction

Look at the two images in Fig. 2.2 with highlighted boundaries on the right. These are regions in the image where objects meet with one another or with the background. Humans are able to interpret complex scenes such as these and predict their approximate depth orderings with relative ease by integrating both bottom-up and top-down cues. In recent years, so-called boundary detectors have become very popular tools. These detectors use local cues, such as brightness, color, texture, gradients and simple features [193] in image patches to distinguish edge points likely at boundaries of surfaces from others. More recent approaches also include globalization processes using long-range relations of image points [6]. However, the image

Figure 2.2: Example results of predicted boundaries (blue) and their ownership (red: foreground, yellow: background) from real-world images: BSDS (above) and NYU Depth V2 (below).

structure in the regions next to an occlusion edge can be used for more than boundary indication; it also encodes information about the relative depth about the edge's two adjacent regions, and to which of the regions the edge belongs to. It has been shown that image cues, such as the convexity of the edge [121], the edge junctions, contrast, or the gradient in the intensity and the texture carry this information [218]. In this work, we focus on detecting classes of bottom-up cues that indicate *border ownership* from 2D image, an important mid-level process for solving the FGO problem that was discussed in Chapter 1. Fig. 2.2 shows example predictions using our proposed approach with their accuracy scores over two popular datasets: the Berkeley Segmentation (BSDS) and the NYU Depth V2 (NYU-Depth) [193, 206]. The prediction accuracy not only is state-of-the-art, but outperforms previous approaches [162, 231]. Our method exploits two novel features derived from findings

in human psychophysics to determine the ownership of a boundary. The first one, known as *extremal edges* or image *folds* [133], captures how changes in the shading of pixels near real boundaries differ between foreground and background. It was shown in [228] that such folds exist in a variety of environments.

The second feature detects Gestalt-like groupings of *mid-level* cues. Specifically, we introduce a new multi-scale grouping mechanism that implements the concept of contour closure, and common patterns such as radial and spiral textures. Since such patterns occur naturally in images, we expect the differences in the distribution of these patterns to be indicative of border ownership. Finally, by embedding these features within a Structured Random Forest (SRF), we are able to predict border ownership in *real-time*, ≈ 0.1s for a 320×240 image. Notably, our method predicts both boundary *and* ownership together in a single step. Compared to previous works that considered border ownership determination as a separate step independent of boundary detection, our single-step approach is not only faster but also more accurate.

2.2 Related Works

Determining border ownership accurately in images involves several related works in computer vision which can be classified into two different areas: 1) depth ordering prediction and 2) object proposals. We briefly review each area in relation to the current work.

Depth ordering prediction. Perceiving ordinal depth from 2D images has been

tackled as early as the classical "Blocks World" of Roberts [237]. Hoeim et al. [109] revisited the problem by combining numerous local and global cues: color, gradients, junctions, textures, sky above ground etc. into a large conditional random field (CRF) for recovering occlusion boundaries and depth ordering in a 2D image. The CRF weights were obtained from training data to ensure consistency of depth across different segments, which were merged in an iterative process from an initial over-segmentation. Along similar lines, Saxena et al. [244] imposed simple geometric constraints to estimate plane parameters related to the 3D location and orientation of each image patch to create a 3D pop-out of the image. Ren et al. [231] considered local convexity and junction cues and integrated them into a CRF to predict border ownership on *Pb* boundaries [193]. Leichter and Lindenbaum [162] followed up by computing distributions of ownership cues in ordinal depth: parallelity, image folds, lower-region etc. over curves, T-junctions and image segments. Stein and Hebert [261] further imposed motion constraints to detect occlusion boundaries consistently across video frames.

Object proposals. A recent trend in computer vision is to detect from an image, object-like regions in the foreground. Early works [3, 61] combined several "objectness" cues to train detectors. However, the applicability of such methods are limited as cue detection and integration is computationally expensive. Recently, Cheng et al. [34] introduced a surprisingly simple technique using binarized gradient norms of images that is able to produce high quality proposals at a fraction of the time of previous methods. The Gestalt concept of *closure* has been exploited by Nishigaki et al. [209, 296] in detecting object like regions via a mid-level grouping

operator termed "image torque". Similarly, using a SRF based structured edge (SE) detector [51], Zitnick et al. [310] counts the number of contours that enter and exit a bounding box region to determine if there is enough closure within the proposed region.

Although many of these works have considered the border ownership problem implicitly in their problem formulation, it is often considered as an independent pixel-wise classification step over predicted input boundaries [162,231] or segmentations [109,261]. In order to ensure prediction consistency over larger scales, CRFs are often used at the expense of computation time. Our approach, by contrast, considers border ownership and boundary detection within a single SRF where consistency over multiple scales are enforced using structured output labels. Our approach is therefore self-contained: we predict both boundaries and border ownership in one single step unlike previous approaches that require further optimizations using a CRF. Consequently, our approach affords us to predict border ownership in real-time.

2.3 Approach

Our approach of determining border ownership via SRFs consists of two key components: 1) Features derived from ownership cues and 2) Imposing border ownership structure in the SRF. We describe these two components in the sections that follow.

2.3.1 Border ownership cues

We use some local cues reported in prior works [52, 81, 162, 216, 218, 231] that were shown to be important in determining border ownership and some new cues. Specifically, we use: 1) shape (convexity/concavity), 2) image folds or extremal edges derived from spectral properties of boundary patches and 3) Gestalt-like grouping features. In addition, our choice of features was influenced by how efficient we can extract them from local patches.

2.3.1.1 HoG-like descriptors

As reported in several previous works, shape cues such as local convexity and concavity of contours are important features that are indicative of foreground objects: the foreground ownership of a boundary tends to be on the concave side [216]. To capture this cue within a local patch, we construct a HoG-like descriptor [47] of image gradients where we quantize the gradient directions into 4 orientation bins. In addition, we use the gradient magnitude as an indicator for good boundary localization. The HoG-like descriptor of gradient orientations captures roughly the local shape of the patch, while its magnitude tells us how likely this patch should contain a real boundary. Notably, as shown in Fig. 2.3 (A), we see that the histograms for typical convex and concave patches are different. For efficiency, we compute these features in terms of "channels" [50] per image patch. Given a patch $\mathbf{P}$ of size $N \times N$, this results in a $N^2 \times 5$ feature vector per patch.

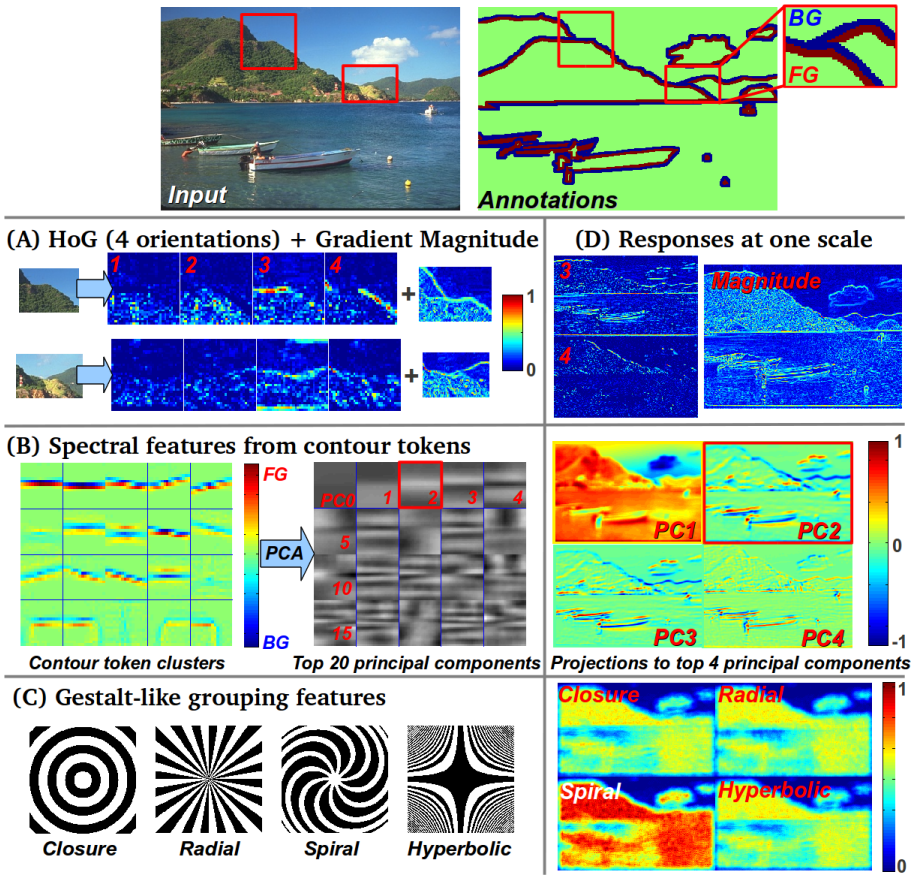


Figure 2.3: Border ownership cues used. (Top) Input image and annotations (red: foreground, blue: background) with example patches boxed. (Below) (A) Local shape (HoG + gradient magnitude) showing four discrete orientations, (B) Spectral features derived via PCA from 20 oriented token clusters (foreground at lower half) and their principal components with extremal edge cues in PC2 (boxed) and (C) Gestalt-like grouping target patterns: closure, radial, spiral and hyperbolic. (D) Corresponding responses at one scale for each of the features. See text for details.

2.3.1.2 Extremal edges from PCA of contour tokens

Extermal edges, or image folds have been known for some time as one of the strongest border ownership cues [52, 81]. Huggins et al. [115] have shown that extremal edges can be reliably detected by computing the so-called shadow flow field in controlled environments. Recently, [228] have shown that extremal edges exists in natural images by performing a principal component analysis (PCA) of aligned oriented boundary image patches. Their key insight is that extremal edges account, after step edges, for most of the gray-level illumination variance at such regions. Motivated by this insight, we derived the basis functions using PCA oriented along so called contour fragments or Sketch Tokens [174] which are similar to shapemes [231] as shown in Fig. 2.3 (B). Since each contour token has an orientation determined by its foreground and background labels, we first orientate all patches so that the background and foreground occupy the top and bottom halves of the patch (using the center pixel as a reference) respectively. Clustering these orientated tokens produces a set of C token centers to which we then apply PCA over the S supporting patches, $\mathcal{P}_c = \{\mathbf{P}_1, \ldots, \mathbf{P}_S\}, c \in \{1 \ldots C\}$. By applying PCA over each $\mathcal{P}_c$, we learn a separate orthonormal basis corresponding to each token center. Specifically, given the $N^2 \times S$ data matrix $\mathbf{X}$ that contains at each column a vectorized (and demeaned) $\mathbf{P}_c$, we apply Singular Value Decomposition on its covariance matrix $\Sigma_{\mathbf{X}}$ to obtain a set of orthonormal basis spanned by the eigenvectors (columns) of $\mathbf{U}$:

$$\Sigma_{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{U}^{-1} \tag{2.1}$$

where we keep the top K eigenvectors, $u_k \in \mathbf{U}$, corresponding to the top K eigenvalues in $\mathbf{S}$ to obtain the projection matrix $\mathbf{W}_c = [u_1, \ldots, u_K]$. $\mathbf{W}_c$ represents a new basis that accounts for most of the variance per contour token center. As features, we reproject $\mathbf{X}$ to obtain $\mathbf{Y}_{K \times S} = \mathbf{W}_c^T \mathbf{X}$, the coordinates of each patch $\mathbf{P}_c$ in the new basis. This yields a feature vector of dimensions $N^2 \times K$. We show in Fig. 2.3 (D-middle) the spectral features derived from the first four principal components (PC). Of note are the responses for PC2-PC4 which exhibit a large response only along real boundaries with positive values encoding foreground ownership and negative values encoding background ownership. In §2.4.2, we show further that PC2 exhibits the characteristics of extremal edges.

2.3.1.3 Gestalt-like grouping features

Gestalt psychologists have developed a set of well-known rules of "Gestalt" that suggests how humans perceive the world from 2D images. Gestalt rules deal with groupings of low-level features (e.g. edges), and can be regarded as a form of *mid-level* cue that captures the holistic properties of individual visual parts. These properties can then be used to organize these visual parts into more meaningful entities that serve as input to higher level processes: e.g. segmentation, recognition etc. In this work, we leverage on specific grouping patterns: 1) closure, 2) radial, 3) spiral and 4) hyperbolic (Fig. 2.3 (C)). Such patterns are useful for border ownership determination because foreground objects tend to exhibit different grouping patterns compared to the background [217], and such patterns have been observed in area

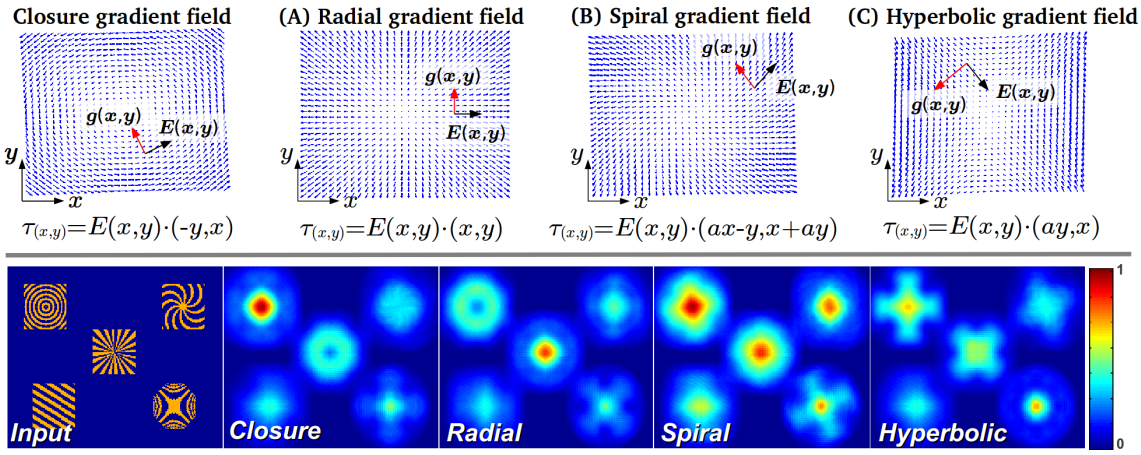


Figure 2.4: Generalizing the image torque for different Gestalt groupings. (Top) By rewriting $\tau_{(x,y)}$ in terms of a scalar product, we are able to generalize the image torque so that it becomes sensitive to: A) radial, B) spiral and C) hyperbolic patterns. (Bottom) Test toy image with different target patterns and their maximum responses over different scales. Notice the selective nature for each target pattern.

V4 of macaques [78]. Closure, one of the strongest cues used in foreground object proposals tasks, is detected in this work by computing the image "torque" [209], $\tau_{\mathbf{P}}$, associated at each patch (Fig. 2.4 (Top-left)). The image torque is so-termed because it is analogous to the torque formulation known in physics, which is the cross-product between a tangential "force" vector $\vec{F}_q$ and its corresponding displacement vector $\vec{d}_{pq}$ where p denotes the center pixel in $\mathbf{P}$ and q an edge pixel in $\mathbf{P}$. The image torque for each edge point q is thus defined as $\tau_{pq} = \vec{F}_q \times \vec{d}_{pq}$. Summing up all $q \in \mathbf{P}$ and normalizing with the patch size yields $\tau_{\mathbf{P}}$:

$$\tau_{\mathbf{P}} = \frac{1}{2|N|} \sum_{q \in \mathbf{P}} \tau_{pq} = \frac{1}{2|N|} \sum_{q \in \mathbf{P}} \left(\vec{F}_q \times \vec{d}_{pq} \right) \tag{2.2}$$

In practice, we search over several scales $s \in \{5, 6, \cdots, N\}$ within $\mathbf{P}$ and retain the maximum torque response over all scales. An alternative derivation for $\tau_{\mathbf{P}}$ is to view the detection of closure patterns as detecting iso-contours corresponding to circles in the image. In general, we consider the patterns we want to detect as the iso-contours of some function f. For example circles are the iso-contours of the function $f(x, y) = x^2 + y^2$. We are interested in the tangent lines of these iso-contours, $g(x, y)$. Given the 2D gradient field, $\nabla f(x, y) = (f_x, f_y)$, the corresponding tangent vectors perpendicular to the gradient field are thus $g(x, y) = (-f_y, f_x)$. From the iso-contour equation of circles, it follows that the closure tangent vectors are $g(x, y) = (-y, x)$. Given an input test patch $\mathbf{P}$, we first determine its gradient field, denoted as $\nabla P(x, y) = (P_x, P_y), (x, y) \in \mathbf{P}$, and their edges (tangent vectors) as $E(x, y) = (-P_y, P_x)$. If a closure pattern exists in $E(x, y)$, then the edges must align well with tangent vectors $g(x, y)$. A simple measure of alignment for a point $(x, y) \in \mathbf{P}$ is thus the scalar product between $E(x, y)$ and $g(x, y)$:

$$\tau_{(x,y)} = E(x, y) \cdot g(x, y) = (-P_y, P_x) \cdot (-y, x) \tag{2.3}$$

which is equivalent to the definition of τ_{pq} for point q. Replacing τ_{pq} in eq. (2.2) with eq. (2.3) yields exactly the same results. The key insight from eq. (2.3) is that we are now able to modify $g(x, y)$ so that eq. (2.3) is sensitive to different patterns in the image. As we show in Appendix A, by writing different target iso-contour equations, we are able to detect different Gestalt patterns using the same formulation. We show some sample responses using different $g(x, y)$ in Fig. 2.4 (Bottom) for four patterns: closure, radial, spiral and hyperbolic. For efficiency, we have implemented eq. (2.2)

as a convolution operation so that their responses can be used directly as features of size $N^2 \times 4$ for training the SRF. Additionally, the responses of the Gestalt features for an example input image are shown in Fig. 2.3 (D-below). We note that because the background (e.g. sky) tends to be textureless, all the features have a small response. Notably, we observe that the strongest response occurs for the spiral pattern, which is localized in the forested foreground region.

2.3.2 Border ownership assignment via SRF

We use an extension of the Random Forest (RF) classifier [107], termed the Structured Random Forest (SRF). Similar to the RF, a SRF is an ensemble learning technique that combines t decision trees, $(T_1, \cdots, T_t)$, trained over random permutations of the data to prevent overfitting. The key difference is that in general, SRFs are able to learn a mapping between inputs of arbitrary complexity (e.g. strings, segmentations, relationships etc.) and similarly complex outputs. Due to their flexibility in representation, SRFs have been used successfully in a variety of computer vision tasks such as boundary detection [51] and semantic scene segmentation [140]. See [45] for a comprehensive review of RFs and their applications. In this work, we show that a SRF can be used as a border ownership classifier by imposing a spatial border ownership structure in the output labels (Fig. 2.5). Similar to [51], we assume that only the target output labels are structured (borders with ownership labels) while the inputs are non-structured (feature vectors derived from image patches).

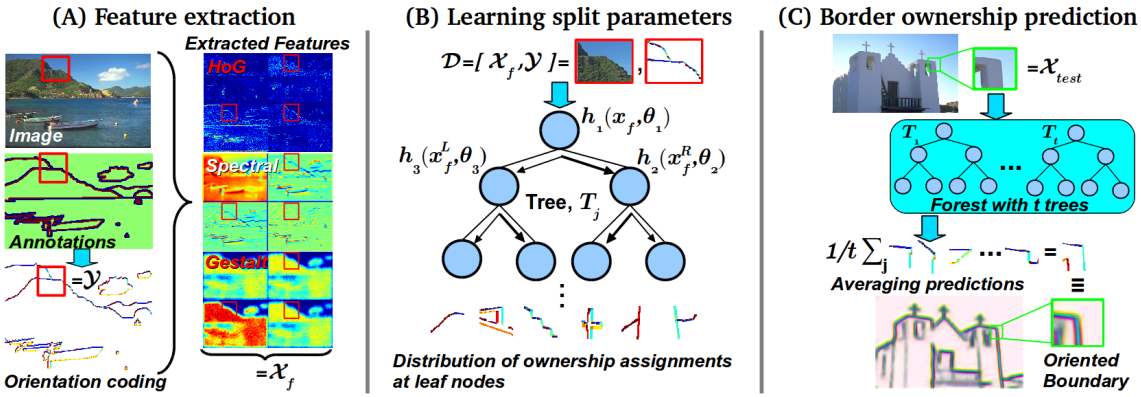


Figure 2.5: Training a SRF for border ownership assignment. (A) Example image with extracted features $x_f \in \mathcal{X}_f$ and ground truth annotations from the highlighted patch. We derive an orientation coding, $\mathcal{Y}$, from the annotations. (B) By mapping $\mathcal{Y}$ to discrete labels, we determine the optimal split parameters θ associated with each split function $h(x_f, \theta)$ that send features x_f either to the left or right child. The leaf nodes store a distribution of border ownership structured labels. (C) During inference, a test patch is assigned to a leaf node within a tree that contains a prediction of the border ownership. Averaging the prediction over all t trees yields the final ownership prediction. We then convert the orientation code into an oriented boundary (blue) that encodes the foreground (red) and background (yellow) predictions.

Let us denote the input as $\mathcal{X}_f$ composed of features $x_f \in \mathcal{X}_f$ derived from a training patch $\mathbf{P}$. The target output is a structured label $\mathcal{Y} = \mathbb{Z}^{N \times N}$ that contains the *orientation* coded annotation of the border ownership. Using a 8 way local neighborhood system, this amounts to 8 different possible orientations of border ownership (Fig. 2.5 (A-bottom)) that each decision tree will predict. The goal of training a SRF (or a RF in general) is to determine, for the i^{th} split (internal) node, the parameters θ_i associated with a binary split function $h(x_f, \theta_i) \in \{0, 1\}$ so that if $h(\cdot) = 1$ we send x_f to the left child or to the right child otherwise. We define $h(x_f, \theta_i)$ to be an indicator function with $\theta_i = (k, \rho)$ and $h(x_f, \theta_i) = \mathbf{1}[x_f(k) < \rho]$, where k is the feature dimension corresponding to one of the features described in §2.3.1. Following [80], we select at most $\sqrt{k}$ feature elements for evaluation. ρ is the learned decision threshold that splits the data $\mathcal{D}_i \subset \mathcal{X}_f \times \mathcal{Y}$ at node i into $\mathcal{D}_i^L$ and $\mathcal{D}_i^R$ for the left and right child nodes respectively. ρ is based on maximizing a standard information gain criterion M_i:

$$M_i = H(\mathcal{D}_i) - \sum_{o \in \{L,R\}} \frac{|\mathcal{D}_i^o|}{|\mathcal{D}_i|} H(\mathcal{D}_i^o) \tag{2.4}$$

We use the Gini impurity measure: $H(\mathcal{D}_i) = \sum_y c_y(1 - c_y)$ with c_y denoting the proportion of features in $\mathcal{D}_i$ with ownership label $y \in \mathcal{Y}$. For non-structured $\mathcal{Y}$, computing eq. (2.4) is straightforward. In the case of structured labels, we first compute an intermediate mapping $\Pi : \mathcal{Y} \mapsto \mathcal{L}$ of structured labels into discrete labels $l \in \mathcal{L}$ following [51] that allows us to compute eq. (2.4) directly. $\mathcal{L}$ is a set of labels that corresponds to different types of possible contour token centers (see §2.3.1.2), and this means that we can reuse the results from the feature extraction

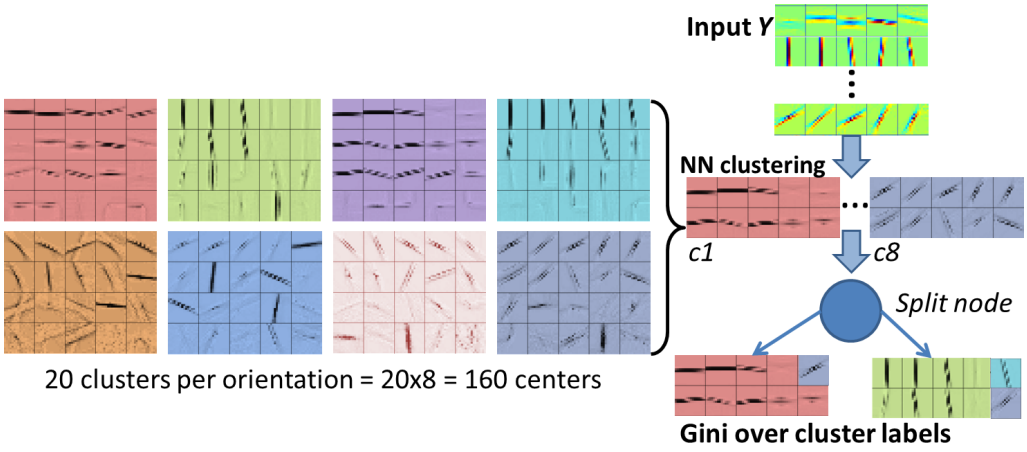


Figure 2.6: Computing the Gini impurity measure, $H(\mathcal{D}_i)$, using cluster labels. (Left) Using cluster centers derived from the 8 discrete orientation coding, we first apply nearest neighboring clustering (NN) over the input structured annotations $y \in \mathcal{Y}$. The labels assigned to each y are then used to compute $H(\mathcal{D}_i)$ so that we improve the purity of $\mathcal{D}_i$ at each split node i.

step during training for added efficiency. Specifically, we apply a nearest neighbor (NN) clustering for each data point y so that it is assigned to one of the cluster labels that each contour token center is associated with. We can then apply eq. 2.4 over these cluster labels l so that we split $\mathcal{D}_i$ appropriately (Fig. 2.6).

The process is repeated with the remaining data $\mathcal{D}^o, o \in \{L, R\}$ at both child nodes until a terminating criterion is satisfied. Common terminating criteria are: 1) maximum depth of tree d_t is reached, 2) a minimum input $|\mathcal{D}|$ is achieved or 3) the gain in M_i is too small. The leaf nodes of each tree after training thus contain the predicted local ownership orientation decision y (Fig. 2.5 (B)). Note that unlike the RF, where a prediction is performed independently per pixel, the SRF enforces spatial consistency in the structured labels at the leaf nodes so that

the final predictions do not change too much along boundaries. In order to account for scale variations, we further sample patches from three (original, half and double) different resolutions of the input image. During inference, we sample test patches densely (at the original resolution) over the entire image and classify them using all t decision trees in the SRF. The final ownership label at each pixel is determined by averaging the predicted orientation labels across all t trees, producing an orientation code that we convert directly into an oriented boundary representation (Fig. 2.5 (C)).

2.4 Experiments

2.4.1 Datasets, baselines and evaluation procedure

We evaluate the performance of border ownership assignment over two publicly available datasets containing real world images: 1) The Berkeley Segmentation Dataset (BSDS) [193] and 2) The NYU Depth V2 (NYU-Depth) dataset [206]. For BSDS, we use a separate subset of 200 labeled images (obtained from the training subset of BSDS-300) that contains ownership annotations. As this dataset was used by the two baseline approaches: 1) Global-CRF of Ren et al. [231] and 2) 2.1D-CRF of Leichter and Lindenbaum [162], the results we report in §2.4.3 are directly comparable. We use the same test/train split as both baselines, with 100 images for training and 100 images for testing. The NYU-depth dataset consists of 1449 RGB-Depth images taken from a variety of indoor environments. The training set consists of 795 images while the remaining 654 images are used for testing. All images in the

dataset are hand annotated with 1000+ object class labels. Following [93], we select the top 35 most frequent object labels (excluding flat surfaces such as walls, floors and ceilings) in order to automatically generate a large number of ownership labels along the boundaries of these objects, using the depth information to produce the ground truth labels for the entire dataset. Compared to BSDS, where only 36.1% of boundary pixels have ownership annotations, we increase the annotation density to nearly 50% in NYU-Depth. Several examples of the input data, ground truths and results are shown in Fig. 2.8.

We report the same accuracy evaluation metric used in [231] and [162], where we count the number of correctly classified border ownership pixels against the ground truth. This is computed via a bipartite graph matching to determine the closest correspondences between the predicted border ownership pixels and the ground truth. Predictions that were not matched are not considered. Following [162], we set this threshold to 0.75% of the image diagonal. The parameters used for training the SRF are the same for both datasets. We use patch sizes of $N = 16$ with $C = 20$ token cluster centers (per direction). 200,000 patches are randomly sampled from the training images. We retain the top $K = 5$ principal components for generating the spectral features. We train a SRF with $t = 16$ decision trees and we limit all trees to have a maximum tree depth of $d_t = 64$ levels.

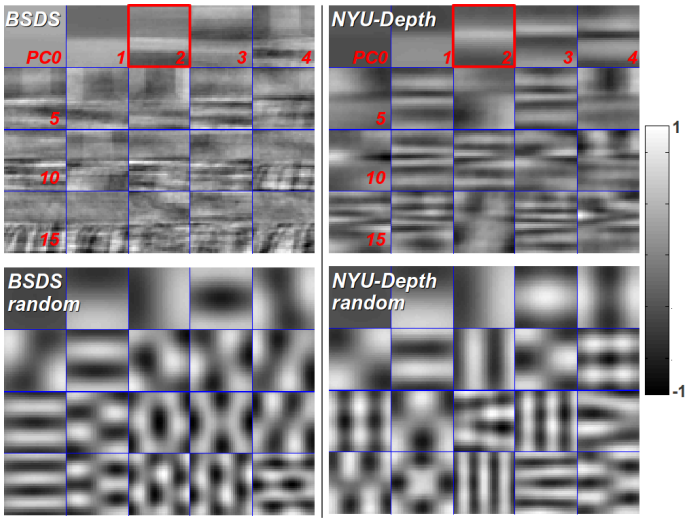


Figure 2.7: Top 20 principal components for BSDS (left) and NYU-Depth (right) for a particular token cluster center. (Bottom row) Components derived from random patches in each dataset.

2.4.2 Comparing spectral components

Before we present evaluation results of the approach, we first perform an analysis of the spectral components produced by applying PCA over clustered token patches in both the indoor (NYU-Depth) and outdoor (BSDS) datasets. We show in Fig. 2.7 a visual comparison of the top 20 principal components (PC) obtained from one token cluster center: horizontal with the background at the top half and the foreground at the lower half of each patch, baselined against components derived from random patches (bottom row). In both datasets, we sampled 500,000 patches. We make four observations. First, the top component (PC1) is the same for both BSDS and NYU-Depth, which is a step edge. The second component (PC2, boxed in Fig. 2.7) exhibits the distinctive signature of extremal edges: with a shading

on the lower-half (foreground) and no shading in the top-half (background). This confirms the observations made by Ramenahalli et al. [228] on the basis of a much smaller number of images (585), and confirm that extremal edges are present across different scenes and environments. Second, we note that the intensity variation in PC2 from NYU-Depth appears "smoother" across the foreground region compared to BSDS. This seems to indicate that extremal edges are more stable in the indoor NYU-Depth dataset. One possible explanation would be that the structured lighting in indoor environments supports the existence of extremal edges better than the diffused lighting common in outdoor situations. Third, we note that other ownership cues such as T-junctions and parallel structures are also captured within the top PCs of both datasets (e.g. PC6 and PC9). Finally, as none of the PCs from random patches exhibit the signature of extremal edges (or other ownership cues), this further confirms that the spectral features we use are unique along true object boundaries.

2.4.3 Results

We perform a series of quantitative ablation studies over different features sets in both datasets and compared their performance with the baselines Global-CRF and 2.1D-CRF in the BSDS dataset. In a second experiment, we also applied the basis functions learned from NYU-Depth (indoor) over the BSDS dataset in order to validate our observations in §2.4.2 that the spectral components from the indoor NYU-Depth scenes are more informative than those obtained from BSDS

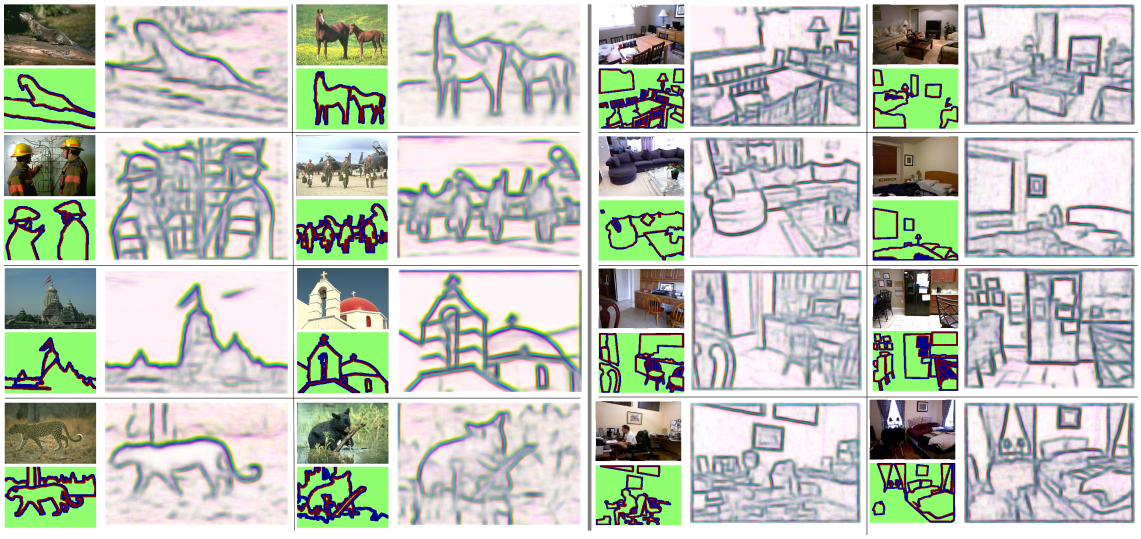


Figure 2.8: Example results from both BSDS (left panel) and NYU-Depth (right panel) datasets. Eight results per dataset: (Top-left counterclockwise): images, ground truth labels (red: foreground, blue: background) and ownership prediction (red: foreground, yellow: background, blue: boundaries).

Feature set	BSDS	NYU-Depth
HoG	72.0%	66.0%
+ Spectral (no contour tokens)	73.1% (72.0%)	67.0% (65.6%)
+ Spectral (contour tokens)	74.0% (72.3%)	68.1% (66.7%)
+ Gestalt patterns	74.4% (72.7%)	**68.4**% (66.7%)
All features + Spectral (NYU)	**74.7**% (72.8%)	-
Global-CRF [231]	69.1%	-
2.1D-CRF [162]	68.9%	-

Table 2.1: Border ownership prediction accuracy for various ablations compared with the baselines (last two rows). '+' denotes the addition of new features to those above the current row. Numbers in parenthesis denote the use of the single feature for prediction.

Method	BSDS-500	NYU-Depth
Our approach	0.73,0.74,0.76	0.63,0.64,0.60
gPb-owt-ucm [6]	0.73,**0.76**,0.73	0.63,0.66,0.56
SE [51]	0.73,0.75,**0.77** (SE-SS)	**0.65**,**0.67**,**0.65** (SE-RGB)

Table 2.2: Boundary prediction accuracy. The numbers reported in each cell are [ODS, OIS, AP] following [6]. Results for gPb-owt-ucm and SE are reproduced from [51].

(outdoor). The full results are summarized in Table 2.1. We show the contribution for individual features, as well as the improvements when the feature is used with other cues. As a point of reference, we note that for BSDS, we are classifying over 18,000 pixels, while we are approaching 2,500,000 pixels for NYU-Depth. Finally, since our approach predicts boundaries in addition to ownership, we evaluate its boundary prediction accuracy in a third experiment (Table 2.2).

Ablation studies of different features. The first four rows in Table 2.1 summarize the mean accuracy of border ownership assignment when different combinations of feature sets are used. The general trend is that with more cues used, the ownership prediction improves for both datasets. We note that the results confirm the usefulness of learning separate basis functions corresponding to different contour token centers (third row), where there is around 1% improvement in accuracy over the case where no contour tokens are used (second row). For the latter, we simply learned a basis over 8 ownership orientations. We also show the contribution of individual features in parenthesis. Of interest is that Gestalt-like features perform on par with spectral features in the NYU-Depth dataset while they have a larger

individual influence in BSDS. A likely explanation is that most indoor man-made objects are *textureless* compared to outdoor environments. Additional experiments with more controlled environments have to be done to confirm this hypothesis.

Applying NYU-depth (indoor) spectral features to BSDS dataset. In the second experiment, we applied the basis functions obtained from NYU-Depth to the BSDS dataset. This results in a slight improvement to 72.8% of its individual contribution. Due to this small degree of improvement, more experiments with a more careful selection of indoor patches should be performed to confirm our hypothesis in §2.4.2. Nonetheless, we note that combining NYU-Depth spectral features with other features yield the best overall prediction accuracy for BSDS (74.7%) in all experiments.

Comparison with state-of-the-art. The prediction accuracy of the proposed SRF border ownership assignment outperforms previous state-of-the-art results: 1) Global-CRF and 2) 2.1D-CRF by at least 2% even using simple HoG-like (shape) features in the BSDS dataset. The performance when all features are combined is even more significant: $> 5\%$ or around 900 pixels that were reclassified correctly. Compared to 2.1D-CRF with a reported mean run-time of 15s, inference using the SRF is $\approx$100 times faster (0.1s).

Boundary prediction accuracy. Our approach (using all features) produces reasonable *boundary* (not ownership) predictions that are comparable with state-of-the-art boundary detectors: gPb-owt-ucm [6] and structured edges (SE) [51] when evaluated over the larger BSDS-500 [6] and NYU-Depth datasets (Table 2.2). Since our approach evaluates test patches at the original resolution without any depth

information, we compared the closest variants of SE: SE-SS (single scale) and SE-RGB (no depth) in BSDS-500 and NYU-Depth respectively. Ablations of features produce insignificant deviations from these results, which shows that the proposed features are more suitable for ownership than boundary prediction. Furthermore, these results are even more significant since our approach is trained on a smaller subset of ownership labels in both datasets.

2.5 Applications of Border Ownership

We demonstrate in this section, two extensions of our proposed border ownership assignment approach. First, we show how ownership information can be used to guide the image torque (closure) operator [209] (§ 2.3.1.3) for the object proposal task, that is, for detecting potential foreground objects. Second, we demonstrate the generalizability of our approach to a completely different sensor, a neuromorphic event-based camera known as the Dynamic Vision Sensor (DVS) [173].

2.5.1 Guiding image torque using ownership information

The image torque closure operator, introduced by Nishigaki et al. [209], is defined as the cross product between a tangential "force" vector $\vec{F}_q$ along an edge point q and its corresponding displacement vector $\vec{d}_{pq}$ towards the center of the patch p. [209] used this operator to detect closed regions, under the reasonable assumption that a region that is closed tends to correspond to an object, and this corresponds to the key Gestalt principle of closure. However, as we will show in Chapter 3,

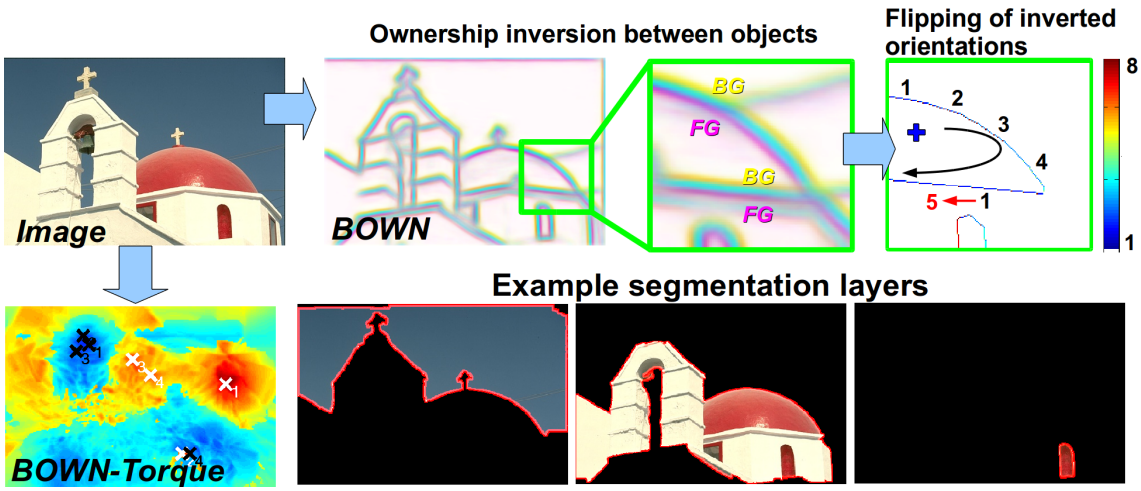


Figure 2.9: Using the torque operator with ownership information for object proposals. (Top row) Given the input image and extracted border ownership (BOWN) information, objects at different depth layers will experience an inversion at shared boundaries (left-boxed). By enforcing an ownership orientation (clockwise), we are able to invert ownerships that do not correspond to the torque center, denoted as a blue '+' (right-boxed). (Bottom row) We then use this ownership-guided torque to select only the *negative* torque points (black crosses) to extract layered segments using the fixation segmentation approach of [199].

this assumption is often violated in real images, where clutter and occlusions result in many wrong closure groupings from *background* boundaries. We present here a simple extension to further improve the results of [209] by imposing an additional *ownership* constraint so that torque groups only the *foreground* side of boundary. The key insight is that ownership information encodes a *directed edge*, $\vec{O}_q$ at each boundary pixel q. Replacing $\vec{F}_q$ with $\vec{O}_q$ in the torque definition therefore encourages (by selecting the correct direction or polarity) foreground boundaries to be grouped: $\tau^O_{pq} = \vec{O}_q \times \vec{d}_{pq}$ (Fig. 2.9).

An important point to note, and is often overlooked, is why would one need a grouping mechanism such as torque when we already have ownership information to begin with? The reason is although ownership indicates the foreground and background regions along the boundaries, it provides only *relative* and local ordinal depth information. Unlike segmenting an object which requires larger and more global cues, we note that in many cases, especially in complex scenes with multiple scene depths, *inversion* of the ownership occurs along shared boundaries between such objects (Fig. 2.9 (top-left)). Such a situation makes it impossible to apply a straightforward approach to segment the object directly from ownership information. The ownership *orientation*, however, encoded by $\vec{O}_q$ (clockwise or counter-clockwise) should remain *consistent* throughout. To do this, we first extract the ownership orientation codings surrounding a particular torque center. We consider only ownership furthest from the center while removing repeats. Next, we select as a start/end point which has orientation '1' (FG below/BG above). Since the foreground object must be closed, a perfect closure would have the following (target)

sequence $[1 \rightarrow 2 \rightarrow 3 \rightarrow \cdots \rightarrow 8 \rightarrow 1]$. Such a situation, however, only occurs in uncluttered, unoccluded foreground. In most cases, inversions will occur within the sequence and we mark them so that when we compute torque, the orientations along such locations are flipped intentionally (Fig. 2.9 (top-right)).

Using the ownership-guided torque in this way presents a robust and elegant mechanism for locating objects at different depth layers. Furthermore, we retain all the advantages of torque: 1) speed, 2) robustness to noisy edges (wrong ownership predictions) and 3) the maximum response provides both the scale and centroid (fixation points) of the object. It also limits the selection of correct torque points: we only need to search for a single torque polarity (which in this case, is negative for the clockwise direction). Finally, this approach models closely neurological studies that show that border ownership tends to have longer contextual range from higher-level regions ([43,306]), and the ownership-guided torque is a computationally viable method to realize this.

Since torque produces fixation points at the maxima over multiple scales, we use them in a fixation-based segmentation [199] to extract the final object segments. We show and compare in Fig. 2.10 segmentation results of ownership-guided torque τ^{O}_{pq} with standard torque τ_{pq}. The key observation is that segmentations extracted from ownership-guided torque are usually the foreground objects compared to the segments from standard torque that segments both foreground and background regions. Using ownership-guided torque also produces more consistent foreground segments with little leakage to the background unlike results from standard torque. Finally, we note that these quantitative results are extremely promising

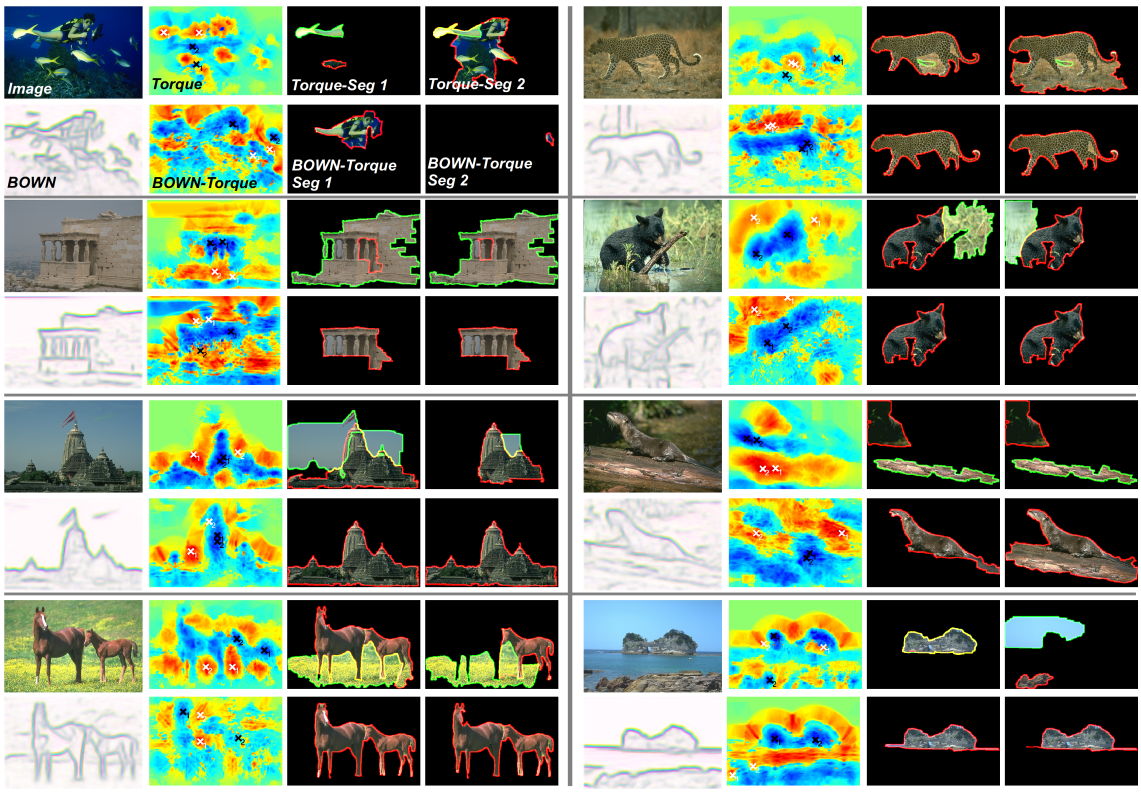


Figure 2.10: Some results of applying ownership-guided torque vs. standard torque. 4 results per column: (L-R) (T):Input RGB image, standard torque output and segments; (B): Border ownership predictions, ownership-guided torque and segments. We show only the segments produced by the top 2 torque points. For standard torque, we show segments from both torque maxima (green) and minima (red) since we do not have ownership constraints unlike ownership-guided torque where we show only the segments corresponding to torque minima.

and indicates further research directions into using ownership information better for foreground-background segmentation.

2.5.2 Predicting boundaries and ownership from DVS

The Dynamic Vision Sensor (DVS), is a neuromorphic camera that mimics the human retina [173] in terms of its design and output. It belongs to a group of novel sensors known as *event-based* cameras, so named because the camera does not create a full image frame using a global clock used in conventional CCD cameras. In conventional *frame-based* cameras, the CCD sensor captures images at a known frame rate (related to the ISO film speed), and reconstructs the entire image after a preset time. This is not what the retina does. Instead, the DVS outputs asynchronous *events*, $ev(x, y, t, p)$ parameterized by its (x, y) spatial location within the 128×128 sensor; the *timestamp*, t of when the event occurred and the *polarity*, $p \in [+1, -1]$ of the events. The sign of p is based on whether the log of the intensity at the same pixel location increases or decreases beyond a fixed global threshold compared to the previous event, $ev(x, y, t-1, p)$.

Due to the event-based nature of such cameras, most Computer Vision techniques in existence are not suitable to handle such input. This is because the key assumption, that we have the entire 2D image or 3D frames (for videos), is simply not available for such cameras. This is the key computational motivation for this work[2]. From a biological perspective, the formation of edges to form object-centric contours

[2]Joint work with Francisco Barranco and was published in [9]. Code, data and more results are available at http://www.umiacs.umd.edu/research/POETICON/DVSContours/

with ownership information models processes in the visual cortex [102,146,284] that are responsible for illusionary contours and ownership assignment (see §1.2.1).

Our approach is similar to what has been described above and in [268], with two important novelties, we: 1) use DVS event-based features and 2) present a *sequential* SRF that improves the prediction as more events are observed. We describe these two innovations next.

2.5.2.1 Event-based features from DVS

Several different event-based features derived from the DVS data are used here. They were selected based on: 1) the ownership and boundary information they capture and 2) their ease of computation. These features can be broadly grouped into four categories and are illustrated in Fig. 2.11, which we describe in the next few paragraphs.

Event temporal information. We show in Fig. 2.11-A the timestamp of the last event triggered for every pixel, measured in terms of *relative* time (ms) with respect to the onset of the event.

Event-based orientation. The events are grouped into eight discrete spatial orientations (from 0 to π). Fig. 2.11-B shows the map of orientations for different spatial locations. For every new event, its timestamp is first compared to the average timestamp of the events in the neighborhood. If the difference exceeds 10 ms, the event is considered an outlier and is discarded. A winner-takes-all strategy is then used to obtain the most likely orientation for the new event which we admit if the

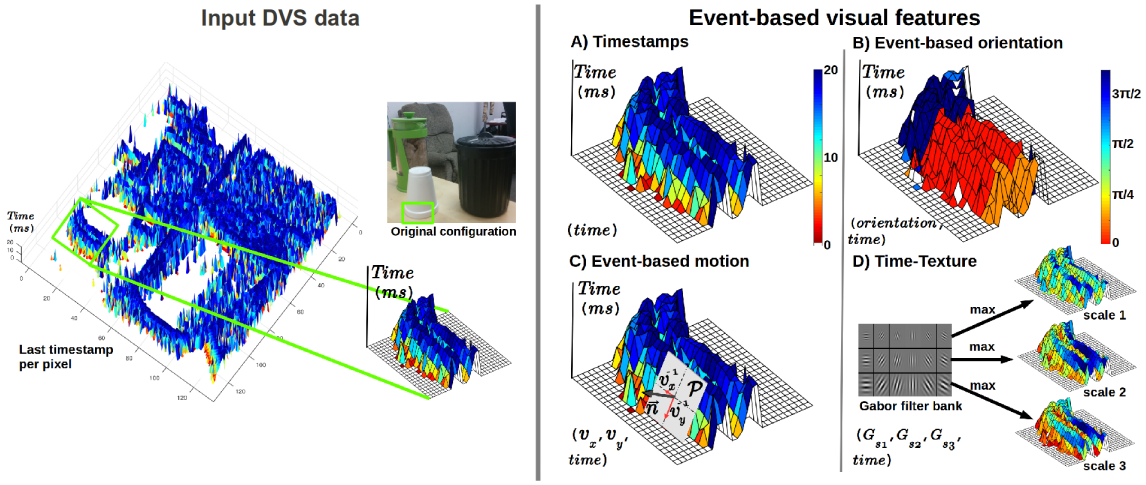


Figure 2.11: Event-based visual features. (Left panel) A 3D spatial representation that encodes the timestamp of the last event in the z axis (after 20 ms) for every pixel in the DVS sensor. The image on the top-left shows the original configuration of the scene (captured with a conventional camera). We show more features derived from the highlighted patch (boxed) in the right panel: A) the last timestamp (time); B) event-based orientation (orientation, time); C) event-based motion estimation, $\nabla\mathcal{T}_e$, computed by fitting local 5×5 planes to the surface $\mathcal{T}_e$ (horizontal component of the motion v_x, vertical component of the motion v_y, time); D) event-based time-texture, obtained from the maximum responses per scale of a bank of Gabor filters with 6 orientations and 3 scales (max response at 1^{st} scale G_{s1}, max response at 2^{nd} scale G_{s2}, max response at 3^{rd} scale G_{s3}, time). Figure adapted from [9].

difference between the new orientation and previous orientation exceeds 2 orientation bins.

Event-based motion estimation. Following [12], we used a function $\mathcal{T}_e$ that assigns to every position the timestamp of its last event. This function locally defines a surface of size 5×5 pixels. The spatial derivatives of this surface provide the speed and direction of the local motion. Specifically, the gradient vector $\nabla \mathcal{T}_e = \left(v_x^{-1}, v_y^{-1}\right)^T$ gives the inverse of the image velocity. In practice, the function $\mathcal{T}_e$ is approximated by fitting a local plane $\mathcal{P}$ (with normal vector $\vec{n}$) to the last timestamp for every location, as illustrated in Fig. 2.11-C. Additionally, a regularization of the data is performed simultaneously along with the plane fitting process. For each new event that is reasonably close (< 0.2 pixels), $\mathcal{P}$ is updated within a time interval of 7.5 ms.

Event-based time-texture. Instead of intensity texture gradients as used on images, we use a map of the timestamps of the last event triggered at every pixel. This map defines a *time-texture* surface and we apply a bank of Gabor filters, using 6 orientations and 3 scales. The three feature maps depicted in Fig. 2.11-D correspond to the maximum response over all orientations at every location, for each of the three spatial scales considered.

All these feature maps are estimated using short time intervals of 20 ms. It is important to note that all feature processing is event-driven: with every new event, all the feature maps and their timestamps are updated with respect to the new event.

2.5.2.2 A sequential SRF for continuous DVS data

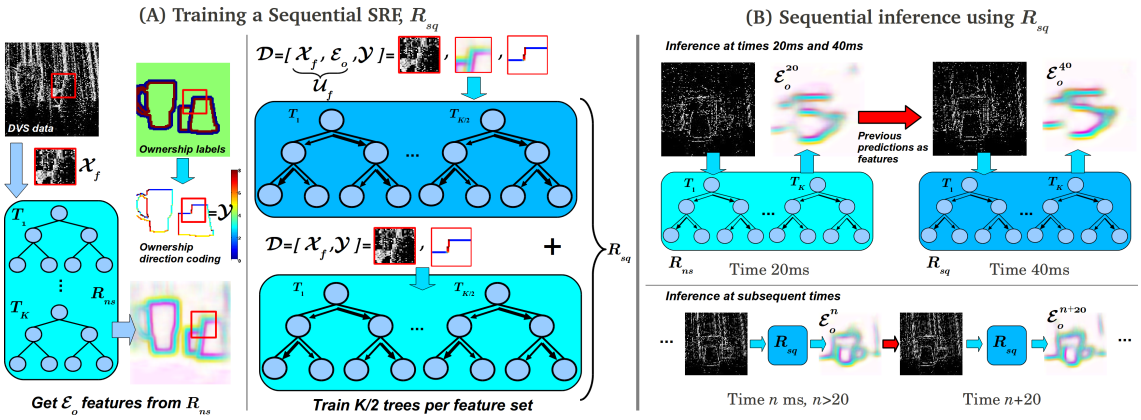


Figure 2.12: Extending a non-sequential SRF (R_{ns}) to a sequential SRF (R_{sq}) given a sequence of DVS input features. (A) Training R_{sq}. We train R_{sq} in exactly the same way as R_{ns} with one key difference, we first run R_{ns} over the training data to provide initial predictions $\mathcal{E}_o$ (left panel) which is then used as an augmented training feature set $\mathcal{U}_f = \mathcal{E}_o \times \mathcal{X}_f$ for learning weights in R_{sq} (right panel). (B) Inference from sequential data. (Top panel) For the first DVS data at 20ms , we use R_{ns} to predict $\mathcal{E}_o^{20}$: the boundaries and their ownership labels. Using the augmented input feature patch, the sequential R_{sq} is then used to produce $\mathcal{E}_o^{40}$ for the second DVS data at 40ms. (Bottom panel) The process is repeated for all subsequent DVS data using R_{sq}.

In practice, as events from DVS arrive in a continuous fashion, our (non-sequential) SRF-based approach for boundary and ownership prediction, R_{ns}, should benefit from more observations of the scene. We describe here an extension that takes advantage of new DVS data to refine the results further using a *sequential* SRF,

R_{sq}. We do this by augmenting the existing DVS features used with the output predictions of the *previous* time's DVS data as shown in Fig. 2.12. Specifically, denoting $n = 20$ as the first DVS data at 20ms, we use the existing non-sequential SRF (R_{ns} in Fig. 2.12 (B)) to produce a prediction of the data, $\mathcal{E}_o^{20}$. For subsequent DVS times, $n + 20$, we augment the input DVS features $\mathcal{X}_f^{n+20}$ with the previous data's prediction $\mathcal{E}_o^n$ to obtain a larger feature set $\mathcal{U}_f^{n+20} = \mathcal{E}_o^n \times \mathcal{X}_f^{n+20}$ which we then pass into a *sequential* SRF R_{sq}. R_{sq} is a SRF that contain $K/2$ trees trained using DVS features $\mathcal{X}_f$ and $K/2$ ($K = 8$) trees trained using $\mathcal{U}_f$ (Fig. 2.12 (A)) which during inference produces two predictions: $\mathcal{E}_o$ from the DVS features and $\mathcal{E}_{osq}$ from the augmented features. $\mathcal{E}_o$ is exactly what R_{ns} predicts for the current DVS data (with half the number of decision trees) while $\mathcal{E}_{osq}$ is a prediction that takes into account the results from the previous time's DVS data. By choosing $w_f \in [0, 1]$, a weight factor that combines these two predictions, the final prediction is thus defined as $\mathcal{E}_o^{n+20} = w_f \mathcal{E}_{osq} + (1 - w_f)\mathcal{E}_o$.

2.5.2.3 Boundary and ownership results

In order to validate our results, we ran a series of feature ablations studies over different DVS sequences that varies in terms of the number of objects (depth-layers), background and motion of the camera. The datasets used are summarized in Table 2.3. Note that two sequences: "NewObj-NewBG" and "Complex-C" sequences are used for testing only. All sequences are used to evaluate R_{ns} except "Complex-C" which is used solely to evaluate R_{sq}. The various feature ablations are selected by

Sequence	Description	# Objects/ # Layers/ # Motion/{# Train | # Test}
Rotation	Mainly rotational motion	1/1/1/{20 | 20}
Translation	Mainly translational motion	1/1/1/{18 | 18}
Zoom	Mainly zoom motion	1/1/1/{18 | 18}
Complex	Up to 3 objects and clutter, different backgrounds	3/3/3/{74 | 53}
NewObj-NewBG	Only for testing: new objects and backgrounds	3/3/3/{- | 47}
Complex-C	Mainly translation + rotation	3/3/2/{-|1}

Table 2.3: Descriptions of DVS sequences used. Note that "NewObj-NewBG" is a held out testing sequence and "Complex-C" is used only for testing the sequential SRF

training the SRF with a subset of the DVS features that capture certain properties that are sensitive to boundary prediction and ownership assignment. We first train separate SRFs that used each feature subset (§2.5.2.1) separately: [Timestamp (TS) only], [Motion Only], [Orientation (Orient) Only] and [Time-Texture Only]. Next, we train a SRF that uses all features together [All features].

We evaluate the performance of our approach by reporting the F-measure over Precision and Recall (P-R) for assessing ownership and boundary accuracy. For boundaries, we use the standard evaluation procedure from the Berkeley Segmentation Dataset [193] to generate P-R curves and report the maximal F-score (ODS) per DVS sequence. For ownership, we compute its F-score, F_{own}, by first matching ownership predictions that are no further than 0.4% of the image diagonal to the groundtruth (same as [162]), and we consider a pixel to have the correct ownership when it's orientation code is less than 90 degrees from the groundtruth. As a

Feature ablations	Rotation	Translation	Zoom	Complex	NewObj-NewBG
Timestamp Only	**0.394**, 0.641, **0.517**	0.308, **0.591**, **0.449**	0.239, 0.498, 0.368	**0.289**, 0.494, 0.391	0.185, 0.366, 0.276
Motion Only	0.307, 0.558, 0.433	0.271, 0.492, 0.381	0.251, 0.475, 0.363	0.267, 0.478, 0.373	**0.207**, 0.392, 0.300
Orientation Only	0.321, 0.570, 0.445	**0.323**, 0.536, 0.429	0.243, 0.494, 0.368	0.279, 0.471, 0.375	0.200, 0.363, 0.282
Time-Texture Only	0.268, 0.552, 0.410	0.197, 0.512, 0.354	0.223, 0.492, 0.358	0.258, 0.460, 0.359	0.206, 0.395, 0.300
All features	0.373, **0.661**, **0.517**	0.313, 0.578, 0.445	**0.268**, **0.523**, **0.395**	0.287, **0.502**, **0.394**	0.204, **0.406**, **0.305**
Baseline	–, 0.218, –	–, 0.237, –	–, 0.344, –	–, 0.273, –	–, 0.302, –

Table 2.4: Performance evaluation of feature ablations over different DVS sequences. For every dataset and ablation, each cell reports the $\{F_{own}$, ODS, $F_c\}$ scores.

final measure of the *combined* performance of ownership assignment and boundary accuracy, we report the average of these two scores, denoted as F_c. Since this is the first approach that detects boundaries from DVS data, there is no other methods to compare with. However, we have created a baseline that groups events using their timestamps. This simple method connects edges to create long contours, if they appear in spatial proximity within a small time interval, and if their orientations match. Moreover, it applies non-maximum suppression as in the Canny edge operator, to make boundaries cleaner.

The evaluation results are summarized in Fig. 2.13 (boundary P-R curves) and in Table 2.4. We show quantitative results using R_{ns} and R_{sq} in Figs. 2.14 and 2.15 as well.

2.5.2.4 Discussion

Boundary prediction accuracy, ODS. From the results, it is clear that our approach significantly outperforms the baseline predictions, producing much better

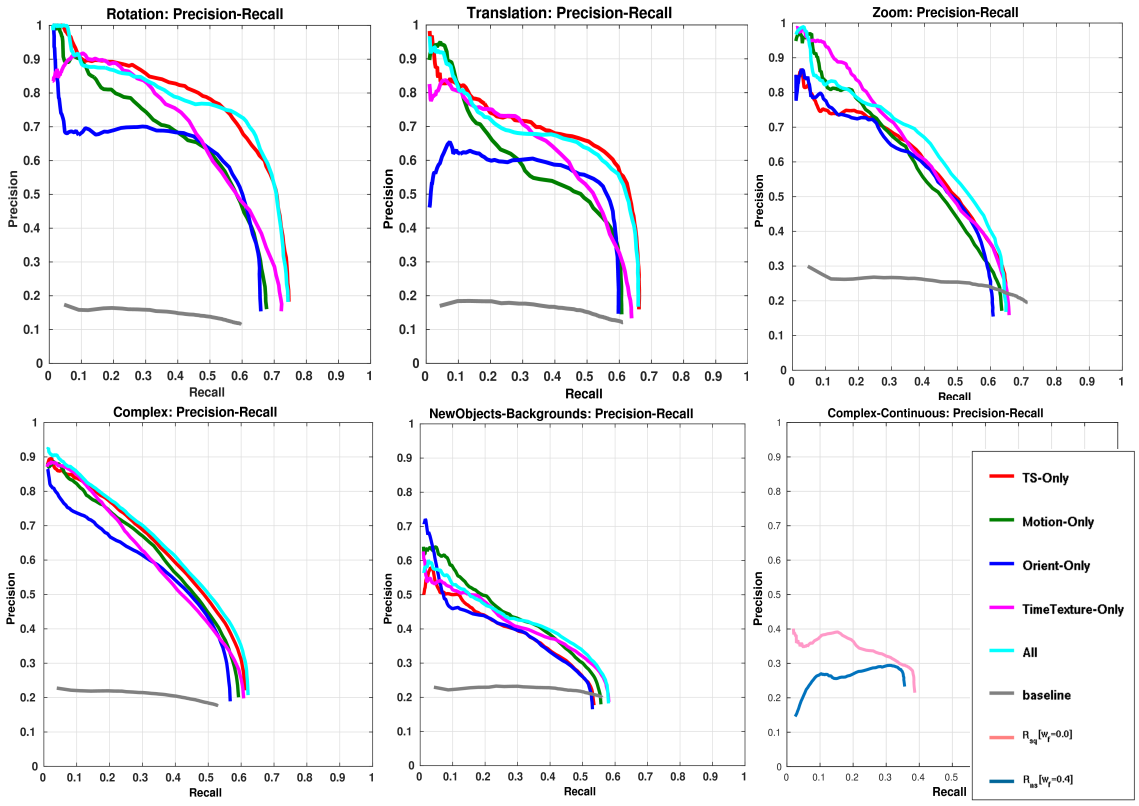


Figure 2.13: Precision-Recall of boundary prediction accuracy for all DVS sequences. Top row (L-R): "Rotation", "Translation" and "Zoom". Bottom row (L-R): "Complex", "NewObj-NewBG" and "Complex-C". See text for details.

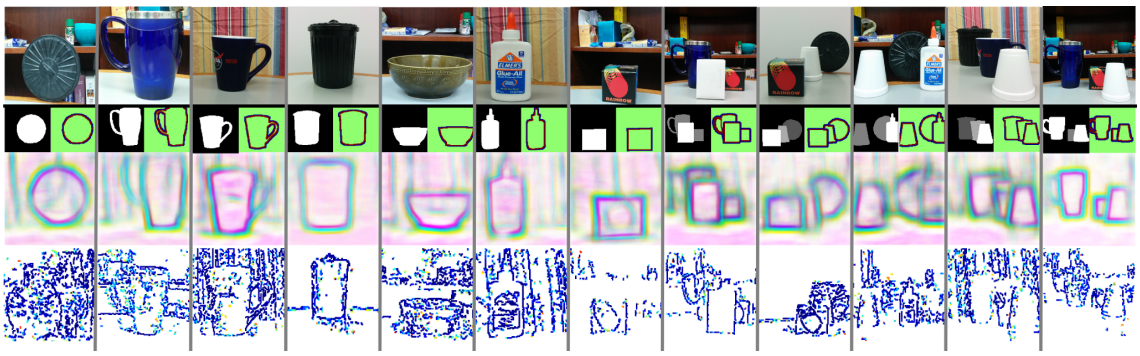


Figure 2.14: Example results using R_{ns}(Top to bottom): Original scene configuration; Hand annotated segmentation and border ownership groundtruths; Predicted boundaries (blue) and ownership (red: foreground, yellow: background) from DVS data; Baseline contours.

SRF	Complex-Continuous
R_{ns} [$w_f = 0.0$]	0.181, **0.324**, 0.252
R_{sq} [$w_f = 0.4$]	**0.212**, 0.310, **0.261**

Figure 2.15: (Left panel) How three different values of w_f affect the final predictions using the sequential SRF, R_{sq}. (Left to right) DVS data from the first 120ms of the "Complex-Continuous" sequence. (Top to Bottom) $w_f = \{0.0, 0.4, 0.7\}$. Notice that the predictions retain more history with increasing w_f, while a small value of w_f (top) is comparatively more noisy. The image on the right shows the final configuration at the end of the sequence. (Right panel) Evaluation results comparing R_{ns} (non-sequential SRF) with R_{sq}: each cell encodes $\{F_{own}$, ODS, $F_c\}$ scores.

boundaries that are closer to the groundtruth. Moving on to the individual features, we first note that Timestamp (TS) is an extremely strong feature that predicts the spatial location of the object (motion) boundary, yielding the highest ODS scores in all sequences (with the exception of "NewObj-NewBG"). This highlights the importance of further studies into the use of the event timestamps, which is a unique feature of the DVS camera, not present in conventional sensors. Next, we note that in "NewObj-NewBG", time textures yield the most accurate results which may indicate some form of invariance under challenging scenarios not captured by other features. Further experiments with more precise motions, however, are needed to confirm this. Finally, we note that using all features together improves boundary ownership in all sequences except "Translation" (where TS remains the best).

Ownership assignment accuracy, F_{own}. We first note that the best results are obtained by different features for different sequences (motions). This shows that ownership assignment compared to boundary prediction is more complicated to capture from the features we investigated and no single feature accurately predicts ownership reliably across different motions (sequences). Interestingly, we note that even though the combination of all features do not yield the best accuracy, it consistently produces one of the top results which shows the advantage of using the SRF to determine the best feature combination. This also highlights another issue: the dependency of the motion pattern on the 3D motion. We believe that a possible approach in a practical application would be to selectively use features for border ownership characterization, according to the general predominant 3D motion, i.e. depending on the kind of motion (predominant parallel translation, zoom,

or rotation) we can use specific SRF classifiers tuned for the predicted motion.

Overall performance, F_c. We note that in spite of the selectivity of features for boundary prediction and/or ownership assignment, the best results (with the exception of "Translation") are obtained when all features are used. This confirms that our choice of features is *balanced* in terms of these two performance criteria and the SRF is trained to make the optimal selection to this end.

Qualitative comparisons. From Fig. 2.14, we note that qualitatively, not only are our predictions much cleaner and smoother than the baseline, we are able to generate these predictions in *real-time* which is a key requirement for event-based approaches.

Results using R_{sq}. We illustrate the effects of using three different values of w_f in Fig. 2.15 (left) over the "Complex-C" sequence: a small value of w_f results in more noisy predictions while a large one retains more temporal history, some of which are propagated to the subsequent DVS times. We have determined that a value of $w_f = 0.3$ to 0.4 provides reasonable predictions that removes temporally inconsistent predictions while reinforcing the strongest predictions over time. This is confirmed experimentally as shown in Fig. 2.15 (right) where the sequential variant of the SRF, R_{sq}, outperforms the non-sequential variant R_{ns} (by setting $w_f = 0.0$) in the combined F-score, F_c. Most of the improvement is derived from improving ownership accuracy, at the slight expanse of boundary accuracy (due to the blurring of edges across time), which is also observed in their corresponding P-R curves (Fig. 2.13 (bottom-right)).

2.6 Conclusions

In this chapter, we have described a real-time approach for simultaneous boundary and border ownership prediction using a Structured Random Forest (SRF) classifier. Our results are state-of-the-art for two modalities: RGB images from conventional CCD cameras and event-based features from DVS. We also described a simple and elegant approach for recovering foreground object segments by reformulating the image torque grouping operator using ownership information. Key to the success of our approach are local and global ownership cues that were efficiently extracted from the input RGB/DVS data. For RGB images, we used well-known cues such as convexity/concavity, extremal edges and Gestalt-like patterns while for DVS data we exploit a variety of time-based features that are indicative of boundary regions.

In the next chapter, we move beyond boundaries and border ownership to a higher-level visual task: shape-based recognition of objects, where we use ownership information to improve recognition and matching of contour fragments in clutter before we modulate the torque grouping operator towards the target.

Chapter 3: Contour-Based Categorical Object Recognition

In this chapter, we propose a method for detecting generic classes of objects from their representative contours in cluttered environments[1]. The approach uses the image torque closure operator [209] to group edges into contours which likely correspond to object boundaries. This 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, we apply the torque to simple edges to extract likely fixation locations of objects. Using the torque's output, a novel contour-based descriptor is created that extends the shape context descriptor [10] to include *border 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

[1]This work was published in [271] and further extended to include contour fragments and rotational invariance in [269]. Full results, code and datasets are available at http://www.umiacs.umd.edu/research/POETICON/contour_based_recognition/

and viewpoint changes. Compared with current state of the art approaches, our approach is able to detect the target with less 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.

3.1 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, is still 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?

As we have noted earlier in Chapter 1, 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 to extract foreground regions from background (§1.1). 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

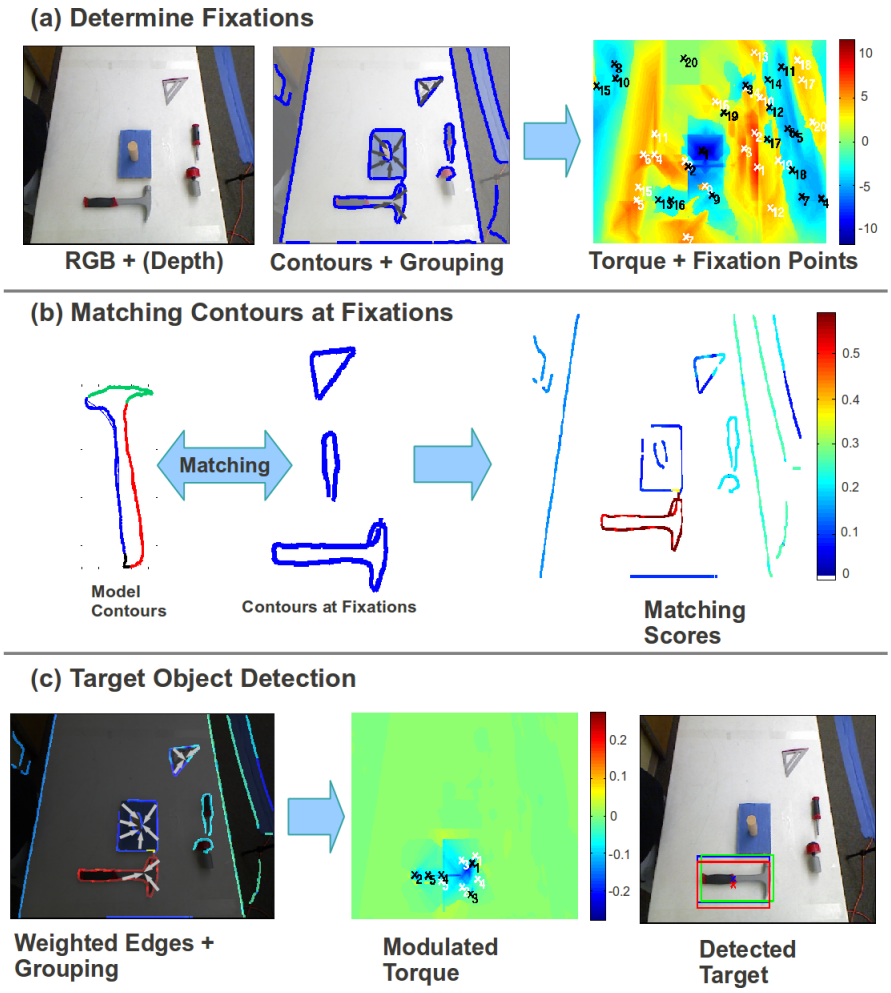


Figure 3.1: From mid-level contour grouping to object recognition. (a) Attention based contour grouping: by grouping contours that support the presence of an object, a set of initial fixation points are used for the recognition step. (b) Contour based recognition at fixation points: using the supporting 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.

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 mid-level vision operator to implement these principles. This operator groups edges within regions of different size 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 (Fig. 3.1(a)), it can be used to find in parallel potential object locations. Then by tuning it to object characteristic edges (Fig. 3.1(b)) to group boundary edges of objects, even when only parts of the object are visible, it can be used to locate and identify specific objects (Fig. 3.1(c)).

The main advantages of using contour information for recognition are that they are: 1) extremely easy to obtain and process using recent state-of-the-art edge detectors [51,174] 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 [193]. 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 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 quickly recognizes 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 – e.g. a `Hammer` class in the UMD Hand-Manipulation dataset or a `Bottle` class in the ETHZ-Shapes dataset (§3.4), 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. 3.2):

1) *Inaccurate and noisy (broken) edges.* Since edge detection in 2D or 2.5D images depends inherently 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 (Fig. 3.2(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 (Fig. 3.2(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, for example in [127,198]. Edge grouping techniques, however, will still fail when considerable clutter (see issue 3 below) occurs and when broken edges predominate.

2) *Boundary detection and border ownership.* In order to distinguish between contours belonging to one object, a key challenge addressed in Chapter 2 of the thesis, is to determine who "owns" the edge. Once the ownership is determined, we can assign an orientation to the contour (Fig. 3.2(b)), which makes it more

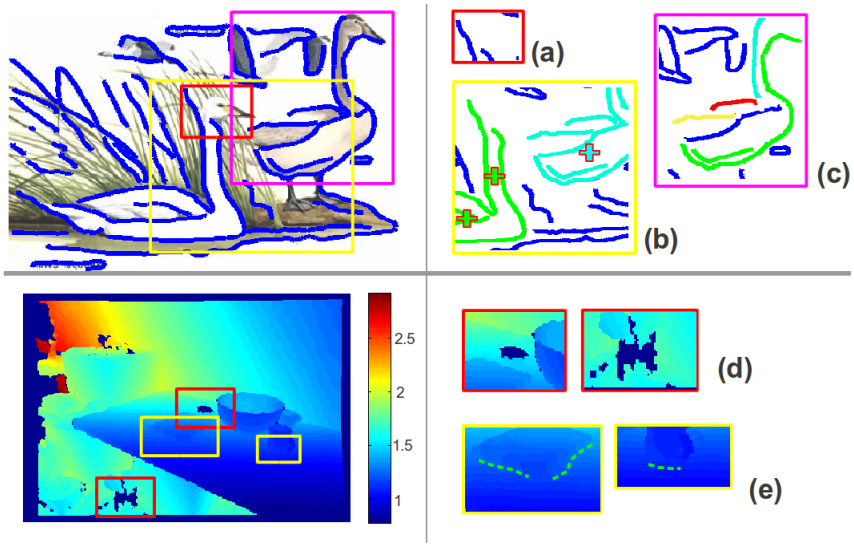


Figure 3.2: Challenges of contour-based categorical object recognition. (Top Panel) 2D images. (a) Noisy edges: some edges on the head are missing. (b) Border ownership between two targets, with support marked as '+'. (c) Detecting partial contours in clutter. (Bottom Panel) 2.5D images. (d) Noise and 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.

discriminative.

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 (Fig. 3.2(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 [187, 234] 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.

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 [212, 246] 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, [188] proposed a discriminative sparse coding that learns a class specific dictionary that detects object specific contours within clutter. [161] 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. [72] approximates them with straight adjacent fragments for part-based matching. Similarly, [229] uses curves instead of straight lines, which are more discriminative, together with a novel scoring function. More recently, [288] 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. [164] uses simple features based on orientations and pairwise interactions to create a local descriptor for matching. [254] 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 [234] and used in a discriminative framework in [141]. [276] proposed using a novel descriptor known as the "chordiogram" to encode relative angles of

boundaries obtained from an initial super-pixel segmentation step. In [185], 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. [190] 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 [211] used a kernel based Support Vector Machine (SVM) [41] with a hough voting approach to detect object specific contours. The recent work of Hariharan et al. [97] combines the outputs trained "poselet" detectors [24] with gPb edges [6] to detect so-called "semantic contours" in images. Instead of hand designed features, the recent work of Bertasius et al. [15] introduced a novel multi-scale bifurcated deep network that detects object-level boundaries. A recent extension [16] extends this network to aid in the detection of semantic contours, with results far surpassing that of [97].

The work of [106] 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 against 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, [111] 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 in-

dicates 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. [118] 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, [171] 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, [298] 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 extends [271] 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 used the full 2.5D information for object recognition. See [94] for an extensive review. Approaches that used local 2.5D descriptors, such HONV or FPFH [241, 266] exploits local geometry and surface normals as their main features. To improve their discriminatory power, [20] used

hierarchical kernel descriptors to produce larger patch based features and trained a linear SVM for 2.5D object recognition. A more recent extension [21] 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 [20].

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 border ownership at all, which is a powerful cue for contour discrimination (see §2.1). Even among approaches that use object centers, either explicitly [161] or implicitly [276] 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 border 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).

3.3 Approach

The proposed approach consists of several steps and is summarized in Fig. 3.3. Prior to detection, contours of the target model are obtained from annotated ground truth of the training set (Fig. 3.3(a)). As the ground truth consists of contours of varying sizes and scales, we first apply Generalized Procrustes Analysis [87] to align the contours of the objects of the same category. Next, motivated by the classical work on representing contours compactly using *codons* [233], 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 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 cu-

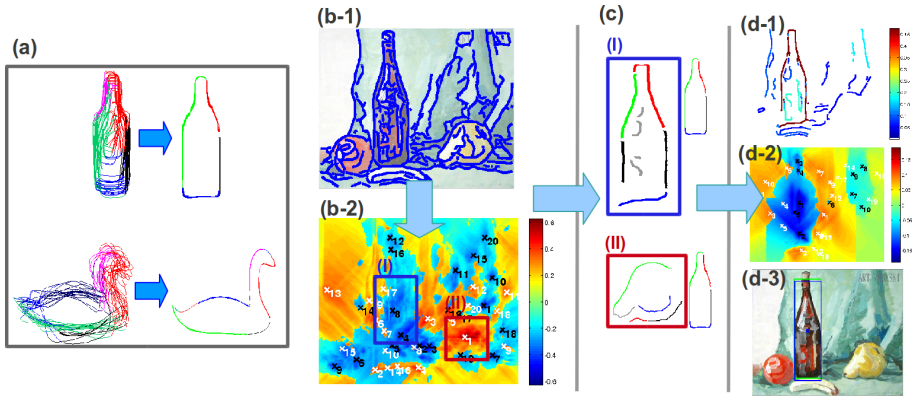


Figure 3.3: Overview of proposed approach. (a) Example model contours fragments (codons) obtained via EM clustering of annotated training data: `Bottles`(top) and `Swans`(below). (b-1) Input image + edge map. (b-2) Original torque value map with detected proto-objects centers $\mathcal{P}_c$: sorted 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.

mulative 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) using any standard edge detection technique (Fig. 3.3(b-1)). We detail the remaining steps in the sections that follow (Fig. 3.3(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 border 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.

3.3.1 Contour completion using image torque

The image torque [209] 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. Formally, the value of image torque[2] , τ_{pq} of an edge pixel q within a discrete image patch with center p is defined as:

$$\tau_{pq} = \vec{r}_{pq} \times \vec{F}_q \tag{3.1}$$

where $\vec{r}_{pq}$ is the displacement vector from p to q and $\vec{F}_q$ is the tangent vector[3] 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 eq. (3.4)). 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:

$$\tau_P = \frac{1}{2|P|} \sum_{q \in E(P)} \tau_{pq} \tag{3.2}$$

We compute eq. (3.2) over multiple scales, $s \in \mathcal{S}$ for every image point, and we extract the largest τ_P over all scales to create a two-dimensional torque *value map*, T_I (Fig. 3.3(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

[2]Here with a slight abuse of notation, we describe $\vec{r}_{pq}$ and $\vec{F}_q$ as two-dimensional vectors, and denote the cross product of these two dimensional vectors as the signed scalar magnitude of the resulting vector obtained by cross-multiplying these vectors. Writing $\vec{r}_{pq}$ and $\vec{F}_q$ 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.

[3]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. As we have shown in §2.5.1, this direction is fixed given border ownership, which is not considered here.

in Fig. 3.3(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 extrema center, $p_c \in \mathcal{P}_c$, we can also compute the torque *contribution* per edge pixel, $\tau_{p_c q_i}$, via eq. (3.1). 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 Fig. 3.3(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 eq. (3.2) 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 extrema 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 eq. (3.1), we apply additional weights w_{pq} that measure the absolute difference in depth values between an edge point q and

the center p:

$$\tau_{d_{pq}} = \vec{r}_{pq} \times (w_{pq}\vec{F}_q) \tag{3.3}$$

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 $\text{abs}(\cdot)$ denotes the absolute value. The torque of an image patch with depth information is similarly derived via $\tau_{d_{pq}}$ using eq. (3.2).

In practice, we use an efficient implementation[4] via the method of Summed Area Tables [46] (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 8 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 [209], 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 towards 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.

[4]Code available online at http://www.umiacs.umd.edu/research/SRVC/NSF-project/

3.3.2 Torque shape context descriptor

Let us return to eq. (3.1), 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:

$$\tau_{pq}^{\omega} = \vec{r}_{pq} \times \vec{f}(\vec{F}_q) \tag{3.4}$$

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_{pq}^{ω} in eq. (3.12) 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 [10]. 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_i^{sc}, is defined as a coarse histogram of the relative coordinates of the remaining $n-1$ points:

$$h_i^{sc}(k) = \#\left[q_j \neq q_i : (q_j - q_i) \in \text{bin}(k)\right], j \neq i \tag{3.5}$$

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 k^{th} bin in the histogram in log-polar space centered over the i^{th} 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 (Fig. 3.4 (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 (Fig. 3.4(Middle-**r2**)).

To overcome these two shortcomings, we introduce two enhancements to the shape context descriptor by: 1) Embedding *border ownership* information through image torque to create a more robust descriptor, termed the *torque* shape context (Fig. 3.4(right)), that can better match contour fragments (§3.3.2.1) 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 descriptor's angular bins via the Fast Fourier Transform (FFT) (§3.3.2.2). Finally, we show how the torque shape context descriptor is matched efficiently via dynamic programming in §3.3.2.3.

3.3.2.1 Robust contour fragment matching from border 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

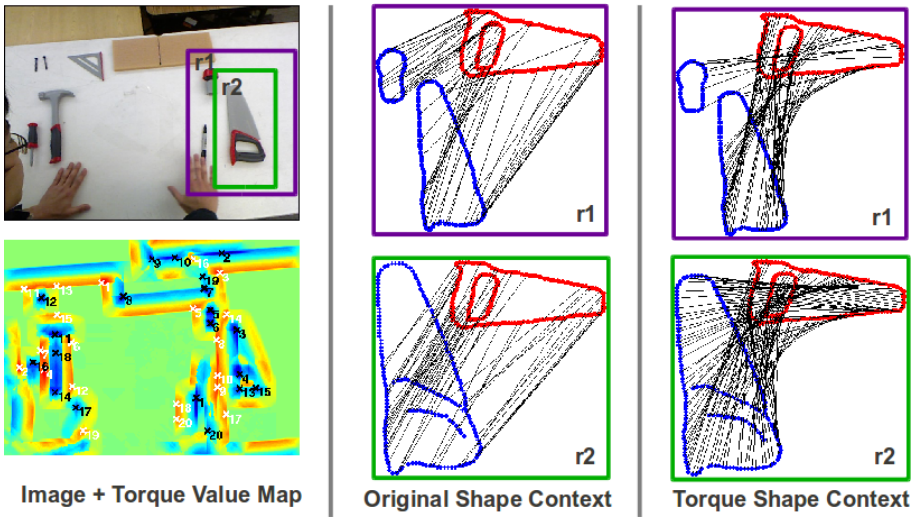


Figure 3.4: Why shape context is insufficient for matching contour fragments in clutter. (Left panel) Input image with torque value map. The two torque fixations considered are boxed as **r1** and **r2** and the model is `Saw`. Model points are red and test points are blue. Black lines indicate correspondences. (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**: Less points from `Borer` object are matched due to border ownership embedding. **r2**: Rotational invariance enables matching of rotated `Saw` to the model.

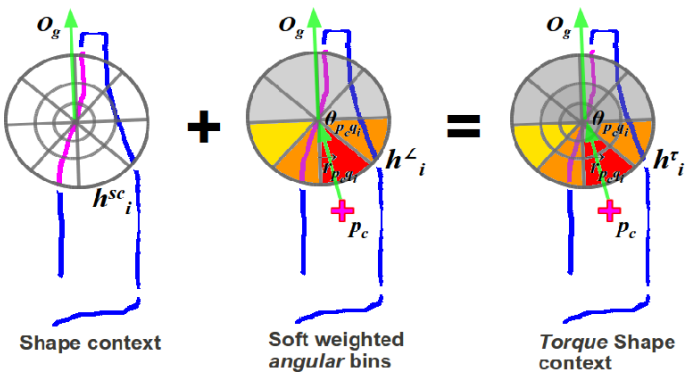


Figure 3.5: Constructing the torque shape context: Selected codon highlighted with respective p_c, $\vec{r}_{p_cq_i}$ and $\theta_{p_cq_i}$ from torque. h_i^{τ} is constructed by adding soft-weighted counts from angular bins in $h_i^{\angle}$ that are intersected with $\vec{r}_{p_cq_i}$ (oriented along O_g, see Fig. 3.9). 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}.

of the shape context histogram additional information that indicates the *location* of p_c, i.e. 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 (Fig. 3.5):

$$\begin{aligned} h_i^{\tau}(k) &= h_i^{sc}(k) + h_i^{\angle}(k) \\ &= h_i^{sc}(k) + \mathcal{K}(\angle\text{bin}(k) \equiv \theta_{p_cq_i}) \end{aligned} \tag{3.6}$$

where $\angle\text{bin}(k)$ denotes the *angular* bins in the shape context histogram in $h_i^{sc}(k)$. $\theta_{p_cq_i}$ is the angle that vector $\vec{r}_{p_cq_i}$ makes with respect to O_g within the coordinate system of the shape context, as shown in Fig. 3.5. $\mathcal{K}(\cdot)$ is a normalized "truncated"

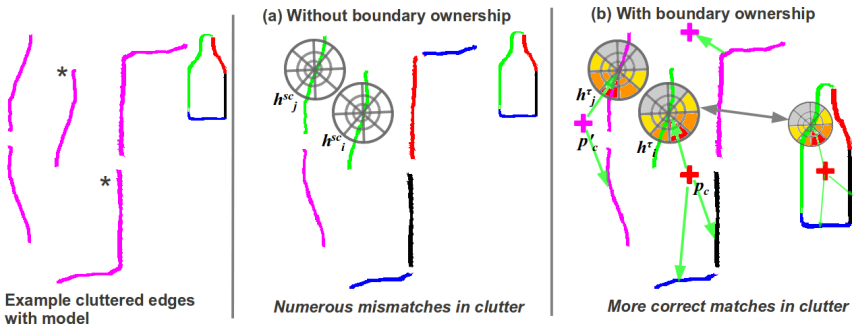


Figure 3.6: Using border ownership information for robust matching in clutter. (Left) Codons in clutter to be matched with the model `Bottle`. The two codons marked with $*$ are correct. (a) Using only shape context, many mismatches occur because of similar histograms in clutter, e.g. h_i^{sc} and h_j^{sc}. Codon colors code for corresponding matches with the model codons. (b) Using the border 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.

Gaussian $N(\theta_{p_cq_i}, \sigma_{\mathcal{K}}^2)\big|_{\theta_{p_cq_i}-\pi/2}^{\theta_{p_cq_i}+\pi/2}$ 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 [5]. What $\mathcal{K}(\cdot)$ does is to weigh angular bins in $h_i(k)$ nearest to $\theta_{p_cq_i}$ more than those angular bins that are not aligned towards $\theta_{p_cq_i}$. 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:

[5] 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 $N(\theta_{p_cq_i}, \sigma_{\mathcal{K}}^2)$ 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.

1) As $\mathcal{K}(\cdot)$ is active only on the side facing p_c, the set of torque shape contexts $\{h_i^\tau | q_i \in \mathcal{Q}_{p_c}\}$ encodes effectively 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, Fig. 3.6 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* against 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 in 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) (Fig. 3.7).

We illustrate the advantages of using $h_i^\tau(k)$ using real cluttered data in Fig. 3.8 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.

3.3.2.2 Rotational invariance via the Fast Fourier Transform

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,

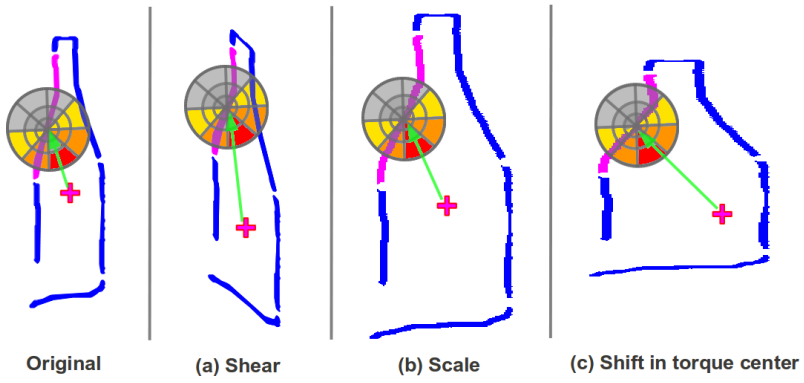


Figure 3.7: Robustness against deformations. The selected torque shape context remains stable for deformations induced by (a) shearing, (b) scale changes and (c) shifts in the torque center.

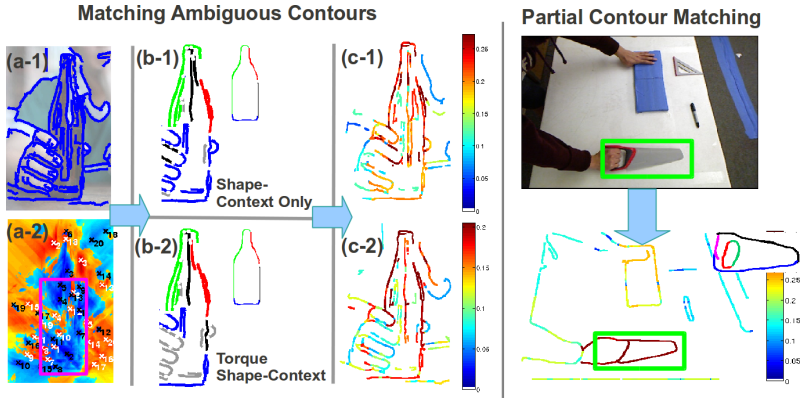


Figure 3.8: (Left) Matching in clutter using torque shape context. (a-1) Input image + edge map. (a-2) Selected proto-object center boxed. (b) Comparing the matches to the model codons with (b-1) shape context and (b-2) torque shape context. Notice that fingers and noisy edges do not have the correct 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).

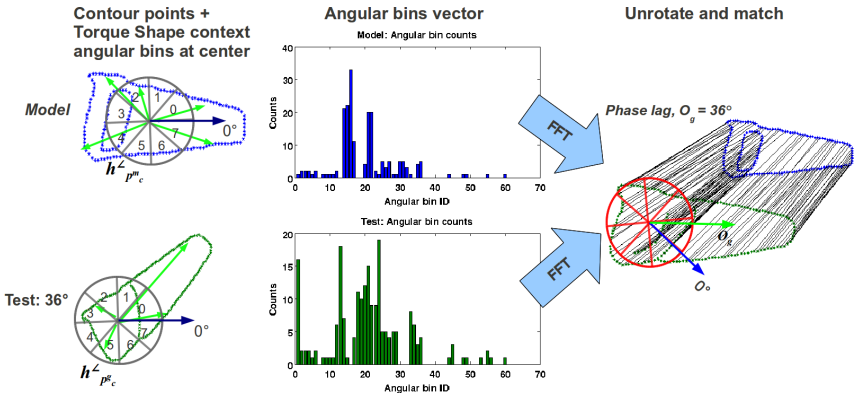


Figure 3.9: Estimating the phase lag O_g from the angular bins of the torque shape context at the torque center. (Left + Middle) Using the angular bins vectors (numbers indicate the bin ID) from the model and the test edges, we estimate the phase lag O_g from the FFT of the two signals. (Right) Using O_g, we "unrotate" the test edges before matching the descriptors (black lines indicate correspondences).

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 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 (Fig. 3.9).

The key idea is to apply a 1D Fast Fourier Transform (FFT) over a 1D vector, $\vec{a}_g = \langle h^{\angle}_{p_c^g}(1), \cdots, h^{\angle}_{p_c^g}(\kappa) \rangle$ 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 captures succinctly the struc-

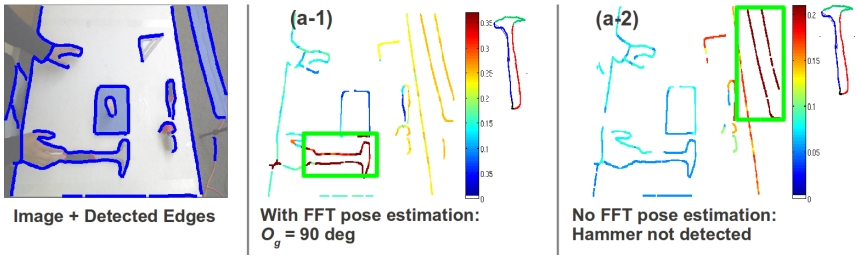


Figure 3.10: Effects of using FFT to estimate O_g on matching accuracy: (a-1) With pose estimation. (a-2) Without pose estimation. The hammer is much better localized and scored using the torque shape context when O_g is applied.

ture of the edge points while it is robust against changes in scale and translation. We obtain in a similar fashion $\vec{a}_m$, from $h^{\angle}_{p^m_c}(k)$ of the model. The cross-correlation between the two discrete signals, $(\vec{a}_g \star \vec{a}_m)[\nu]$ is then obtained via FFT to determine the most significant phase lag $O_g = \text{argmax}_\nu(\vec{a}_g \star \vec{a}_m)[\nu]$ 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 (§3.3.3). 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 Fig. 3.10. 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 §3.4.1 quantitative results that highlight the importance of this procedure over a challeng-

ing hand manipulation dataset in improving the recognition of tools that are often occluded and placed at random orientations.

3.3.2.3 Matching of torque shape context descriptors

Following [10], we compare the torque shape context defined in eq. (3.6) using the χ^2 statistic. We use the dynamic programming method of [273] 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):

$$C_\phi(\mathcal{G}', \mathcal{M}) = \gamma_{sc} C_{sc}(\mathcal{G}', \mathcal{M}) + \gamma_\angle C_\angle(\mathcal{G}', \mathcal{M}) \tag{3.7}$$

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 eq. (3.6)) and the angular bin histograms (second term in eq. (3.6)) 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 SC and angular components of the torque shape context histogram for the i^{th} test point and corresponding $\phi(i)$ model point as g_i^t and $m_{\phi(i)}^t$, with $t \in \{sc, \angle\}$ respectively. The matching costs $C_t(\cdot), t \in \{sc, \angle\}$ for these two components are similarly defined as:

$$C_t(\mathcal{G}', \mathcal{M}) = \sum_{i=1}^{n'} \chi^2(g_i^t, m_{\phi(i)}^t) \tag{3.8}$$

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 $m_{\phi(i)}^t$

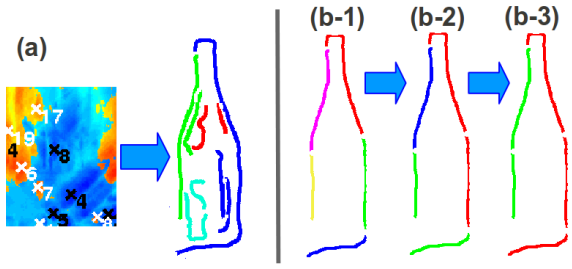


Figure 3.11: Multi-scale edge matching: (a) Detail of p_8 from Fig. 3.3(b-2). Neighboring torques p_c with their supporting edges (in similar colors) are combined. (b-1) to (b-3) Increasing scales of combining neighboring codons together for matching.

in the model. χ^2 is defined for two sets of shape context histograms centered at $(g_i^t, m_{\phi(i)}^t)$ as:

$$\chi^2(g_i^t, m_{\phi(i)}^t) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i^t(k) - h_{\phi(i)}^t(k)]^2}{h_i^t(k) + h_{\phi(i)}^t(k)} \tag{3.9}$$

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 $\mathcal{G}'$:

$$D_{sc}^{\tau}(\mathcal{G}', \mathcal{M}) = \frac{1}{n'}\sum_{i=1}^{n'} C_{\phi(i)}(\mathcal{G}', \mathcal{M}) \tag{3.10}$$

Since D_{sc}^{τ} is a local measure of similarity of partial edge fragments, we show in §3.3.3 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.

3.3.3 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 Fig. 3.11. 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[6] 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 eq. (3.10) 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

[6] the codons are indexed in a clockwise direction.

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 §3.3.2.2 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 [187, 234] 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:

$$W_{D^{\tau}_{sc}}(r_i) = \beta_c + \beta_f(\exp(-E_{D^{\tau}_{sc}}(r_i)/(2\sigma))), \tag{3.11}$$

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 eq. (3.11) to the scale at which $\vec{F}_q$ was detected, we obtain the *modulated* image torque that is sensitive to the target object class:

$$\tau^{\omega}_{pq} = \vec{r}_{pq} \times (W_{D^{\tau}_{sc}}(q)\vec{F}_q) \tag{3.12}$$

where $\vec{f}(\vec{F}_q) = W_{D^{\tau}_{sc}}(q)\vec{F}_q$ as in eq. (3.4). Finally we can compute the modulated

torque per patch P by replacing τ_{pq} in eq. (3.2) with τ^{ω}_{pq}:

$$\tau^{\omega}_{P} = \frac{1}{2|P|} \sum_{q \in E(P)} \tau^{\omega}_{pq} \tag{3.13}$$

For 2.5D images, we add in the depth constraint $w_{pq} = \text{abs}(I_d(p) - I_d(q))$ similarly as described in §3.3.1 to redefine the modulated torque with depth information:

$$\tau^{\omega}_{d_{pq}} = \vec{r}_{pq} \times (w_{pq} W_{D^{\tau}_{sc}}(q) \vec{F}_q) \tag{3.14}$$

and we define the modulated torque per patch in the same way as in eq. (3.13) 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 with the model will be detected as well: e.g apples and oranges (round), sticks and rods (elongated) etc. 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 provide a summary of the algorithm in Appendix B. When the depth image I_d is available, the algorithm remains the same except we replace eqns. (3.2) and (3.13) with eqns. (3.3) and (3.14) 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 contour matching step.

3.4 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 [111] 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 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.

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. t_c, the threshold to select the strongest edges, is set to the 50^{th} percentile of the ranked torque contribution scores from the grouped edges. The number of codon neighbors to combine, $\mathcal{J}$, is set to 3 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 [193] to derive I_e. For computing torque, we search over image patches with sizes ranging from 3 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 320×240 image are around $\sim$15 seconds using a Matlab

implementation running on a Core i7 2.4GHz 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 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, τ_P^m (replacing τ_{pq} in eq. (3.2) with τ_{pq}^m from eq. (3.4)) with τ_P.

3.4.1 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 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 (§3.3.2.2) in improving the recognition accuracy.

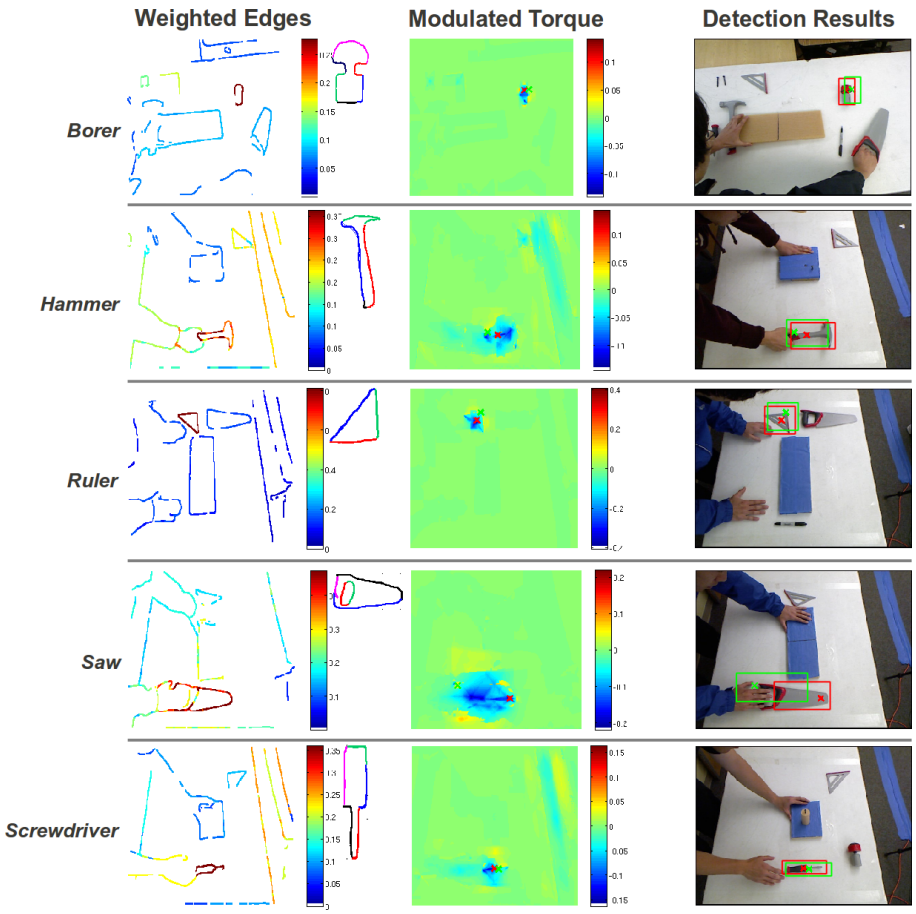
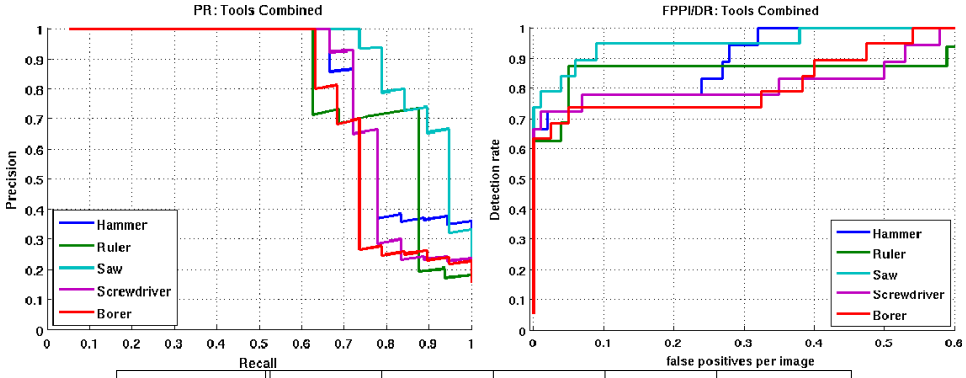


Figure 3.12: Detection results from five sample frames of the UMD Hand-Manipulation dataset. (Rows) Target object class: (Top to Bottom) `Borer`, `Hammer`, `Ruler`, `Saw`, `Screwdriver`. (Columns) Left: $W_{D^{\tau}_{sc}}$ where red means higher values and target model contours at top-right, Middle: Modulated torque showing the top 2 object detections (red and green crosses), Right: RGB frames overlaid with detection results. Note that for `Hammer` and `Saw`, the objects are partially occluded by the hands.

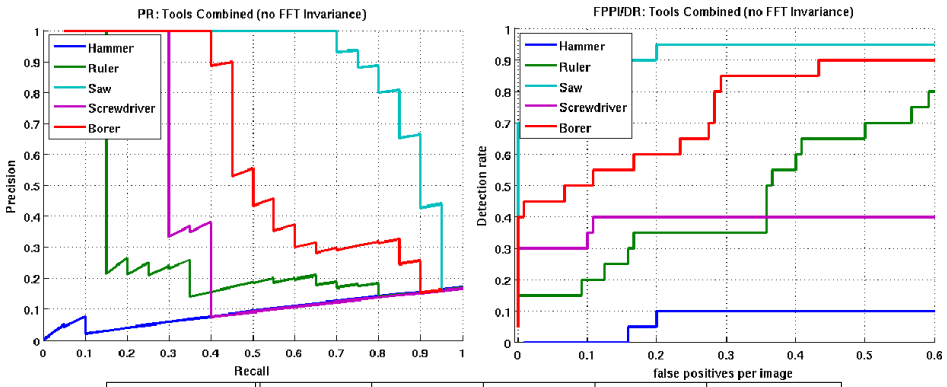
We used the meta-parameters and evaluation procedure as indicated above. For obtaining the target model codons, we used the initial first ten frames and hand annotated the target tool's contours to obtain the model codons. We then evaluated the rest of the sequence at sample intervals of 10 frames each, which yielded a total of around 800 evaluated frames in the entire dataset. We show some results from sample frames of the dataset in Fig. 3.12: final edge weights $W_{D^{\tau}_{sc}}$ and the predicted target objects with centers marked as crosses.



	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

Figure 3.13: (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.

For evaluation, we report the Precision/Recall (PR) rates and corresponding Detection Rate/False Positives Per Image (DR/FPPI) curves. The results are summarized in Fig. 3.13 for all 5 tools considered and compared to Fig. 3.14 where no rotational invariance is applied to the procedure. From the results of the full



	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

Figure 3.14: Performance without incorporating rotational invariance via FFT: (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.

approach, we are able to localize the target objects in clutter with Average Precision (AP) 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 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 contribution of estimating O_g via FFT is also clearly shown in Fig. 3.14 when we note the improvements in AP 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 Fig. 3.12 first column where the model codons are shown on the top right).

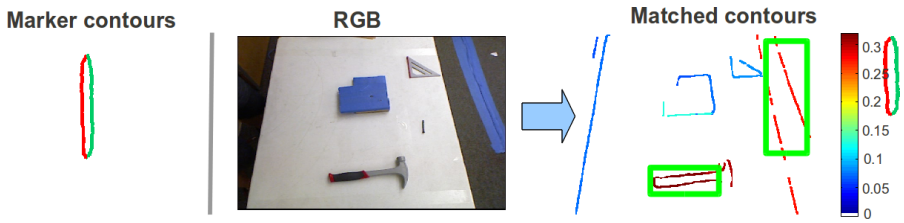


Figure 3.15: When contour information alone is not enough. (Left) Model contours of the `Marker` class. (Right) Contours that have long parallel lines are highlighted (in green boxes): e.g. the handle of the hammer or the sides of a tape.

The decrease in performance of `Borer` (and to a large extent `Screwdriver` as well) compared with other tools as noted in Fig. 3.13 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 Fig. 3.15(left). Due to the small number of contours in the model, such a configuration is highly ambiguous (Fig. 3.15(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.

3.4.2 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 [111] and compared it with the state of the art LINE2D algorithm of [106] 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 [111] that includes the probabilistic occlusion prior, since our approach does not explicitly model it. The dataset consists of eight textureless objects: `{bakingpan, colander, cup, pitcher, saucepan, scissors, shaker, thermos}` placed among other common kitchen objects with 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 [111] 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 Fig. 3.16. The

	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

Table 3.1: Comparing detection rates with our method, LINE2D [106] and rLINE2D [111] at 0.3/0.4/1.0 FPPI over the CMU Kitchen Occlusion dataset.

detection rates at 0.3/0.4/1.0 FPPI are summarized in Table 3.1.

From the DR/FPPI curves, we first note that for all the objects, our method significantly outperforms LINE2D and has a performance that is at least on-par or better than rLINE2D which includes an occlusion model of the target (which our method does not have). Second, from Table 3.1, 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 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 that use gradient orientations only. We show some example detection results with the modulated torque in Fig. 3.17 that illustrate how the approach performs over this dataset.

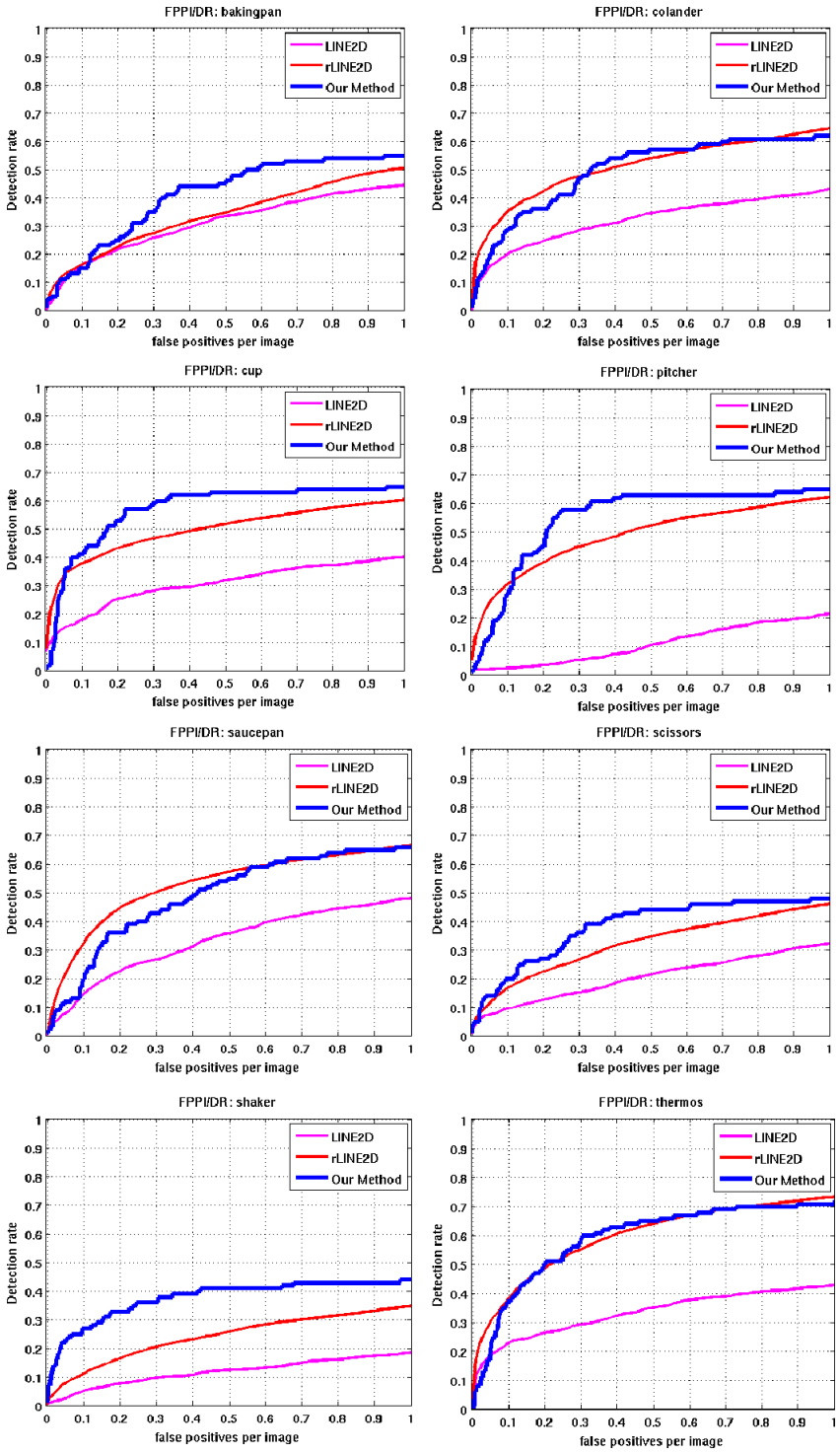


Figure 3.16: DR/FPPI curves comparing our approach with LINE2D [106] and rLINE2D [111] over the eight object categories in the CMU Kitchen Occlusion dataset.

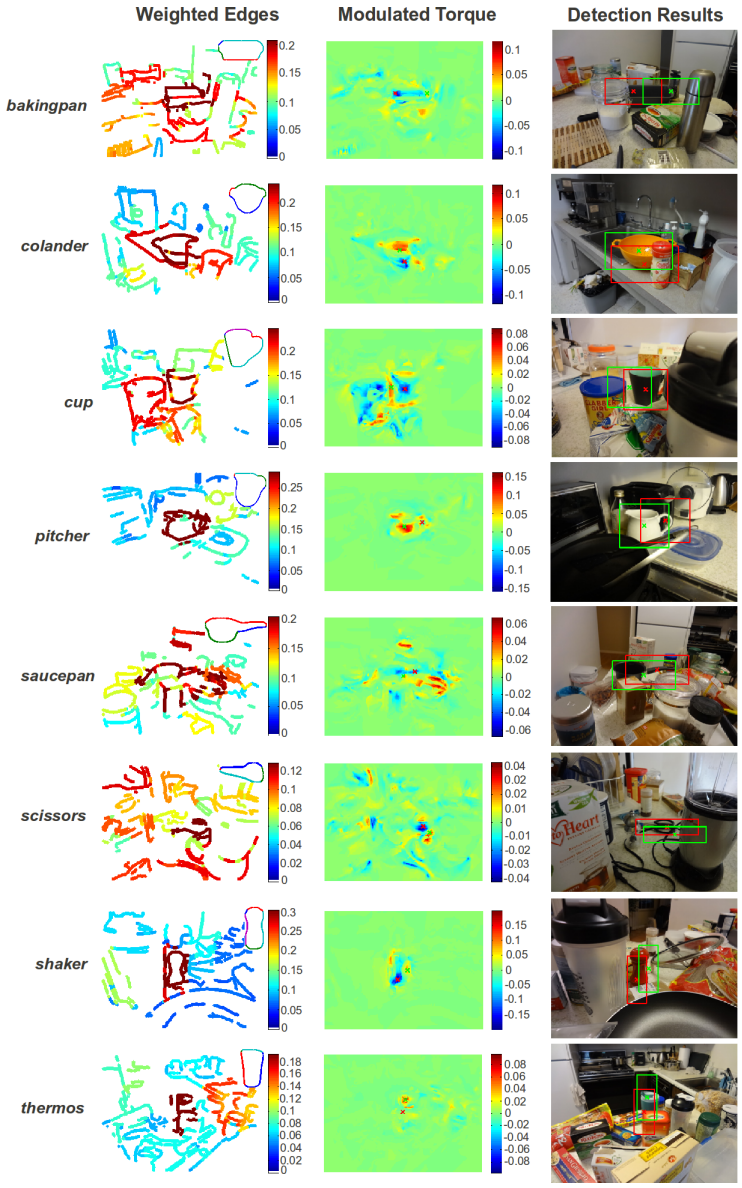


Figure 3.17: Detection results for the eight objects in the CMU Kitchen Occlusion dataset. (Rows) Target object class: (Top to Bottom) `bakingpan`, `colander`, `cup`, `pitcher`, `saucepan`, `scissors`, `shaker`, `thermos`. (Columns) Left: $W_{D^{\tau}_{sc}}$ where red means higher values and target model contours at top-right, Middle: Modulated torque showing the top 2 object detections (red and green crosses), Right: RGB frames overlaid with detection results.

3.4.3 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 background, clutter, scale and viewpoint. We follow the same test/train split procedure as suggested by [254] 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.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our Method	**0.917**	0.931	0.796	**0.888**	0.891	**0.885**
[190]	0.869	0.724	0.742	0.806	0.716	0.771
[254]	0.845	0.916	0.787	**0.888**	0.922	0.872
[187]	0.881	0.920	0.756	0.868	**0.959**	0.877
[288]	0.866	**0.975**	**0.832**	0.843	0.828	0.869

Table 3.2: Comparing interpolated average precision (AP) with the proposed method over the ETHZ-Shapes dataset.

We focus our comparisons with recent state of the art contour-based object detection methods [187, 190, 254, 288]. The Precision/Recall (PR) curves of these methods and their interpolated average precision (AP) are compared with the pro-

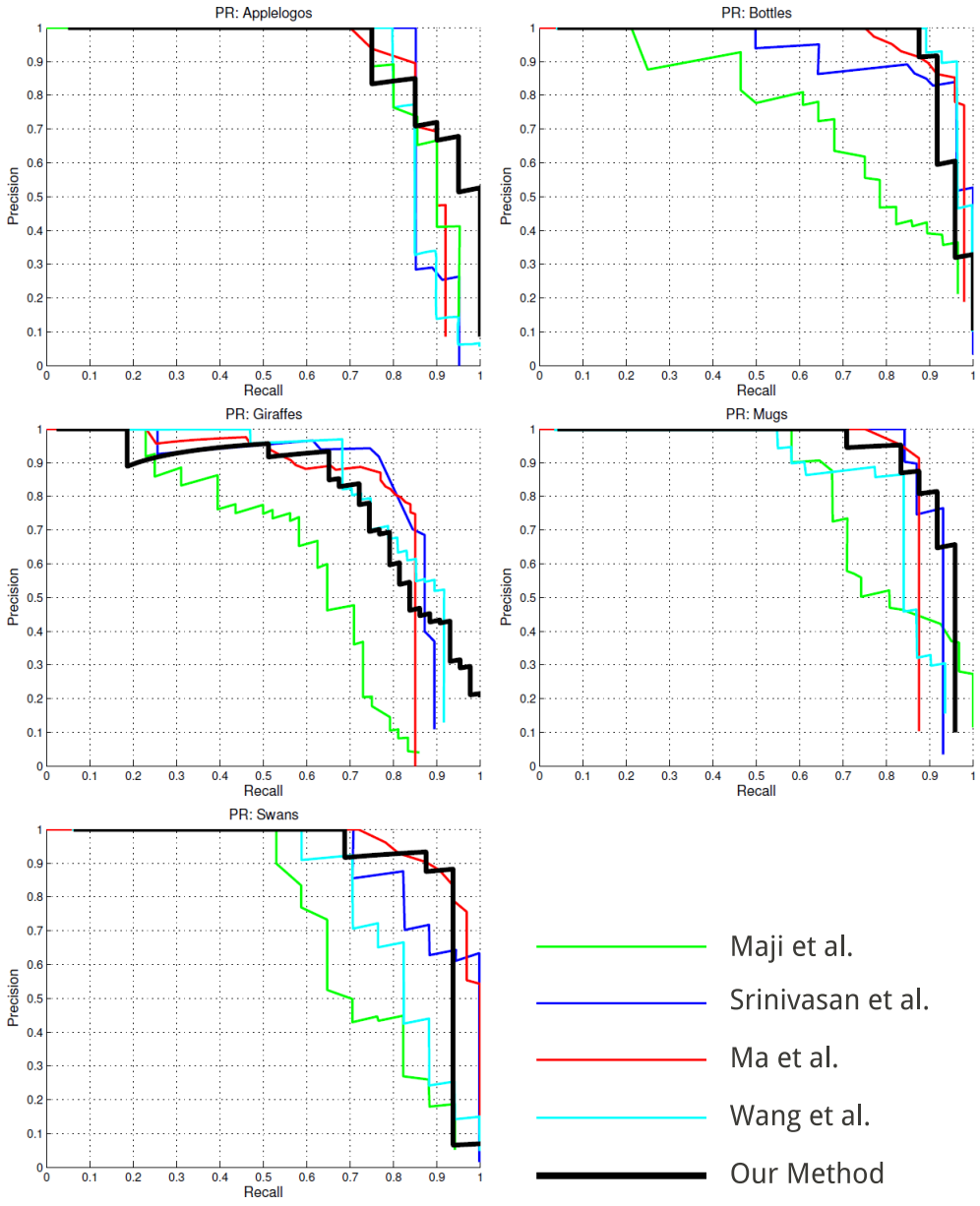


Figure 3.18: Precision/Recall curves comparing [187, 190, 254, 288] to the proposed method over the ETHZ-Shapes dataset.

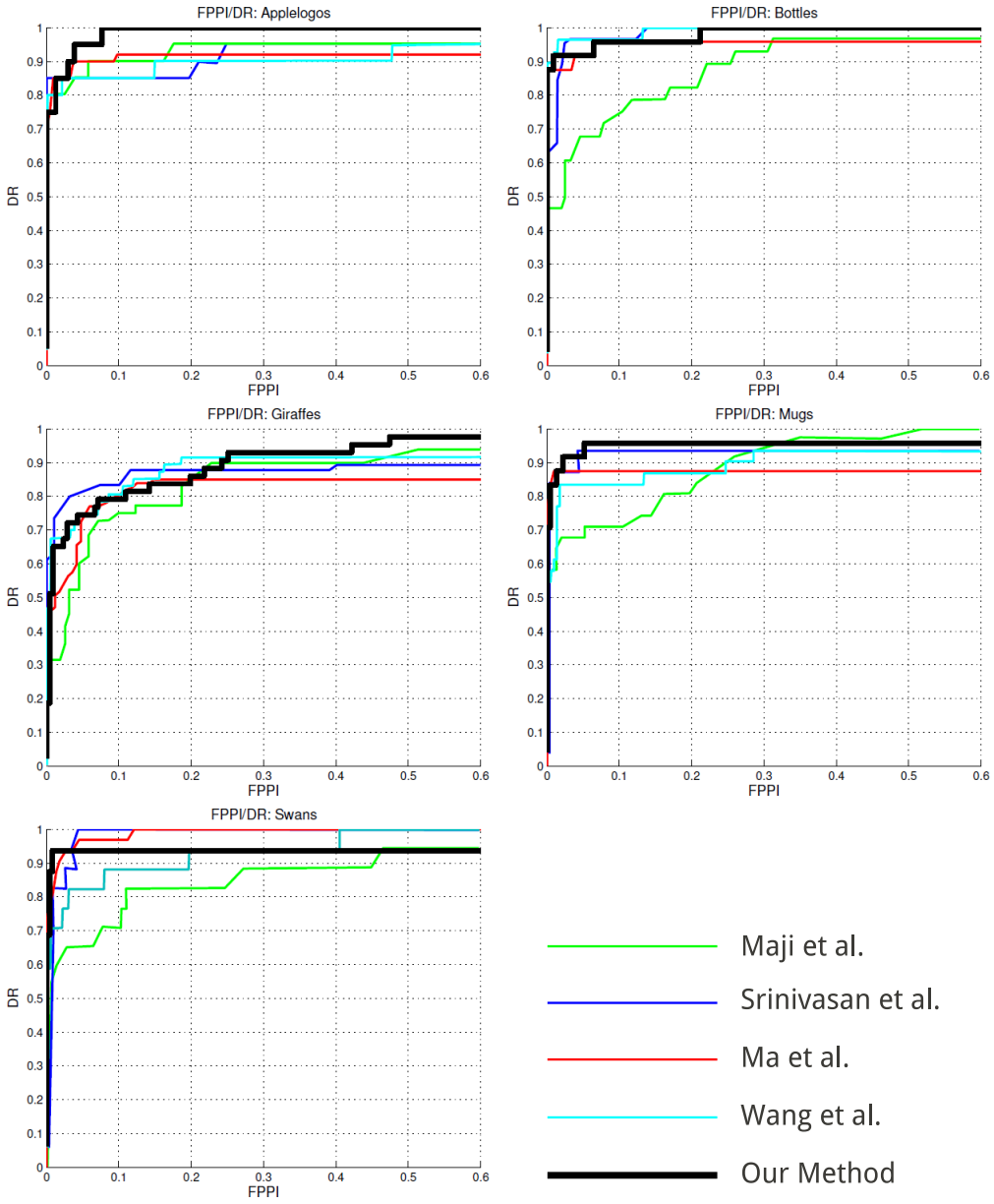


Figure 3.19: Comparison of DR/FPPI curves over the ETHZ-Shapes dataset.

	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**
[190]	0.95/0.95	0.929/0.964	0.896/0.896	0.936/**0.967**	0.882/0.882	0.919/0.932
[254]	0.95/0.95	**1/1**	0.872/0.896	0.936/0.936	**1/1**	0.952/0.956
[187]	0.92/0.92	0.979/0.979	0.854/0.854	0.875/0.875	**1/1**	0.926/0.926
[288]	0.90/0.90	**1/1**	0.92/0.92	0.94/0.94	0.94/0.94	0.940/0.940
[234]	0.933/0.933	0.970/0.970	0.792/0.819	0.846/0.863	0.926/0.926	0.893/0.905
[71]	0.777/0.832	0.798/0.816	0.399/0.445	0.751/0.8	0.632/0.705	0.671/0.72

Table 3.3: Comparing detection rates at 0.3/0.4 FPPI over the ETHZ-Shapes dataset.

posed method in Fig. 3.18 and Table 3.2 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 5 categories, our approach is able to achieve the overall best mean AP among the compared methods, with a small improvement over [187].

In addition, we plot the Detection Rate/False Positives per Image (DR/FPPI) curves in Fig. 3.19. The detection rates at 0.3 and 0.4 FPPI are compared with several reported results in the literature in Table 3.3. 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 in Fig. 3.20 some example results: 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.

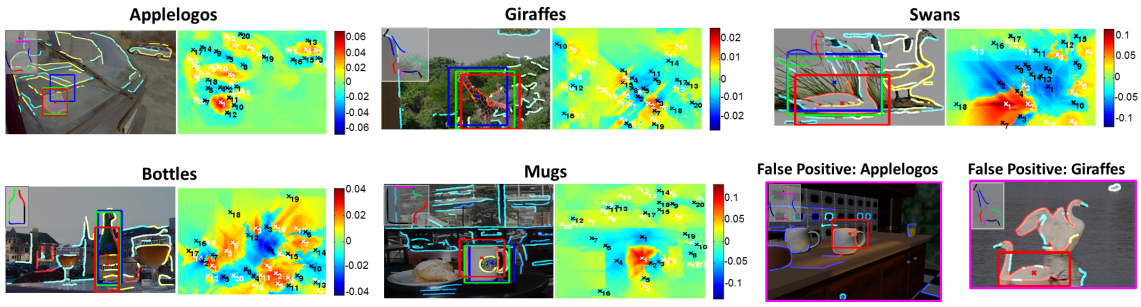


Figure 3.20: Some example detection results with their modulated torque. Edges show values of $W_{D^{\tau}_{sc}}$, green boxes are ground truth, red and blue boxes are the top min/max modulated torque values: Top row (left to right): `Applelogos, Giraffes, Swans`. Bottom row (left to right): `Bottles, Mugs`. False detections of `Applelogos` and `Giraffes`. Best viewed in color.

3.4.4 Object recognition in clutter by a mobile robot

We demonstrate the feasibility of our approach for practical robotic applications on our mobile robot platform (Fig. 3.21(left)). The robot consists of the Adept Pioneer P3-DX base together with a custom made frame on which a Kinect RGB-Depth sensor is attached via a Directed Perception PTU-D46 pan-tilt unit (PTU). The robot's software runs over the Robot Operating System (ROS) [227] with appropriate interfaces implemented to send the Kinect RGB-Depth data 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 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 [271]: we performed three different collec-

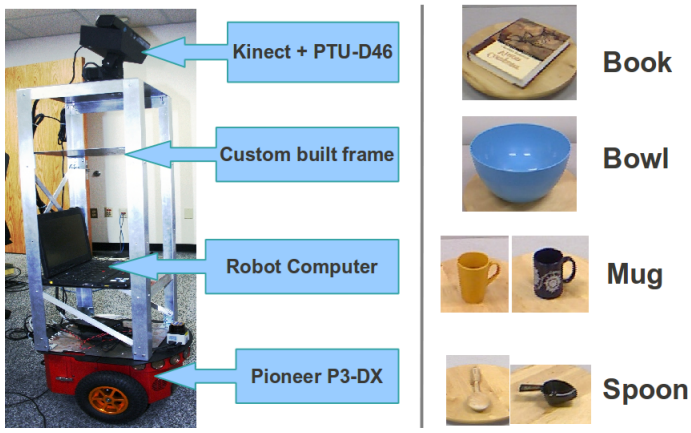


Figure 3.21: (Left) The mobile robot with relevant hardware components highlighted used in the experiment. (Right) The four object categories used. For `Mug` and `Spoon`, two different instances exist in the dataset.

tions of the Kinect RGB-Depth data with differing amount of clutter per dataset, with around 1000 frames per sequence. We focus on detecting four object categories (Fig. 3.21(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 [271] 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 Hierarchical Matching Pursuit (HMP) method of [21] that learns a dictionary of RGB-Depth features for object recognition. A linear SVM classifier is then trained over the features for 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 (eq. (3.12)) with the approach that uses depth information eq. (3.14). 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. Fig. 3.22 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 3.4.

	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 [271]	**0.33**/0.33	0.14/0.14	0.43/0.43	0.17/0.17	0.09/0.09
HMP-RGBDepth [21]	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

Table 3.4: Comparing detection rates of our approach that uses depth information and one that does not use depth information with the baseline Shape-Torque [271] and HMP [21] at 0.3/0.4 FPPI over the UMD-clutter dataset.

From the results, we see that with the exception of `Spoon`, the proposed method with depth information is on par or better than the baseline and standard non-depth approach at both FPPI levels. This shows that given a cluttered environment,

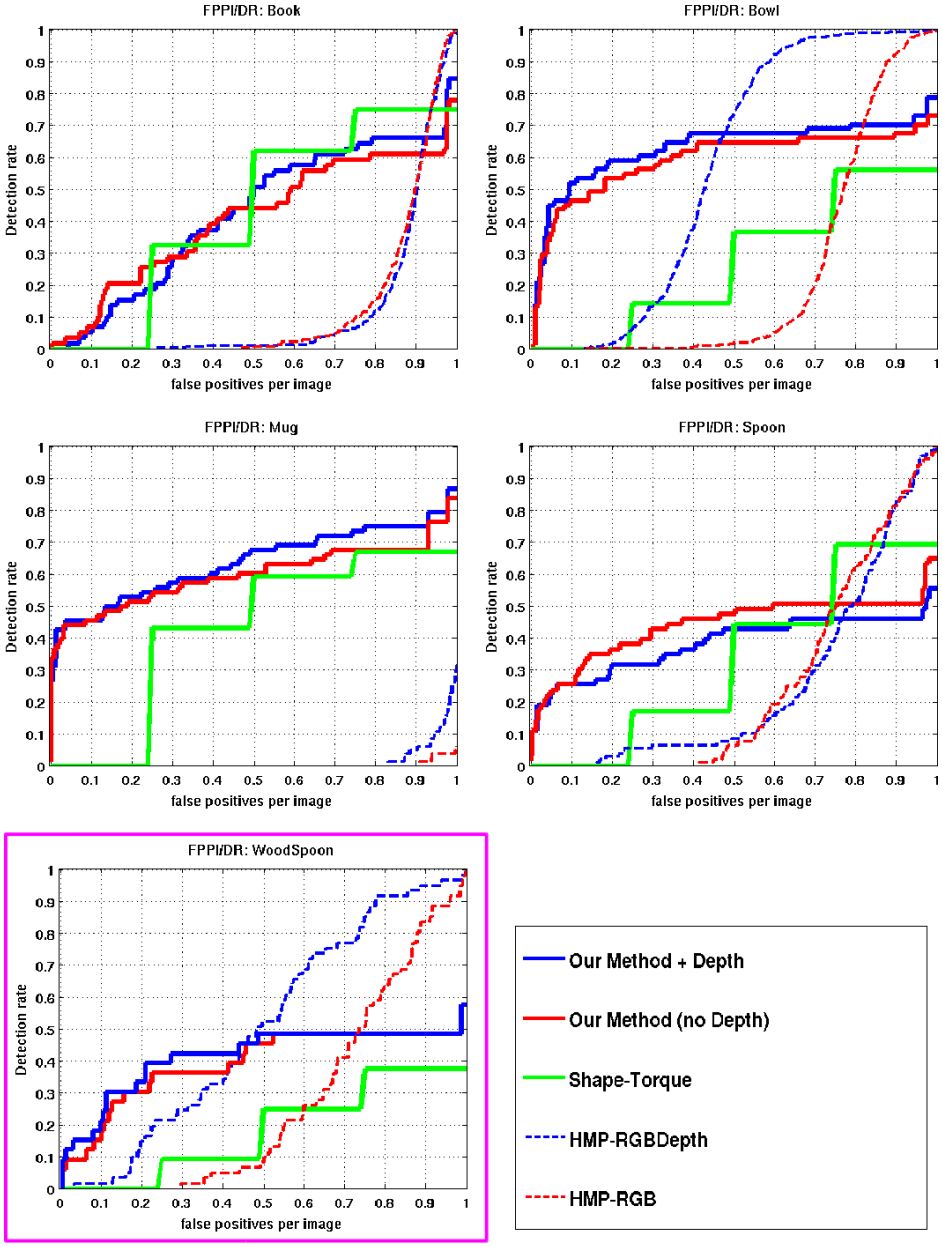


Figure 3.22: DR/FPPI curves of the UMD-clutter data evaluated over four object classes. The DR/FPPI curves for the *Wood Spoon* (boxed) instance is presented to contrast with the results shown for the `Spoon` category, which is affected due to bad depth estimates, see text for details.

using the depth information enables us to reduce the influence of false contour groupings that have very different depth values and hence unlikely to come from the same object. Fig. 3.23 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, e.g. `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 (Fig. 3.22, boxed), where depth information improves the result.

We show some results from sample frames of the dataset in Fig. 3.24. Specifically the figure shows the final edge weights $W_{D^{\tau}_{sc}}$, the modulated torque value map with depth constraint and the predicted objects with centers marked as crosses.

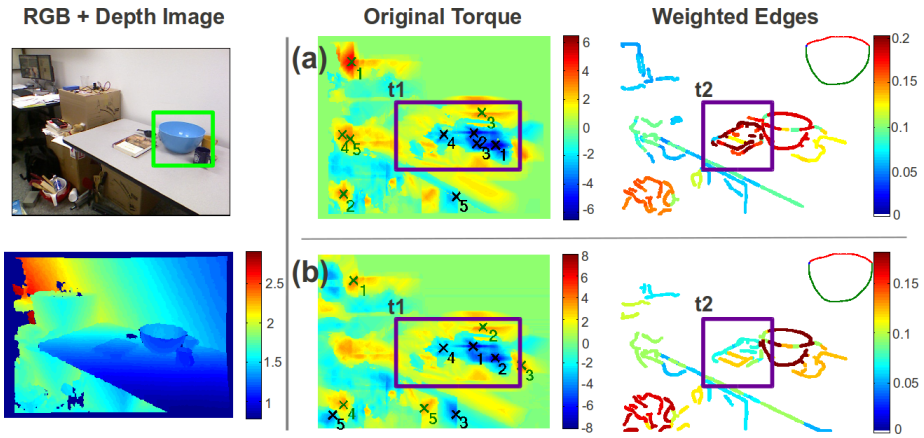


Figure 3.23: How depth information helps in improving the image torque. (Left) Input Kinect RGB-D image with target `Bowl` in green box. (Right) Comparing effects of (a) not using depth information and (b) using depth information. Region **t1**: Using depth information produces more depth consistent 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).

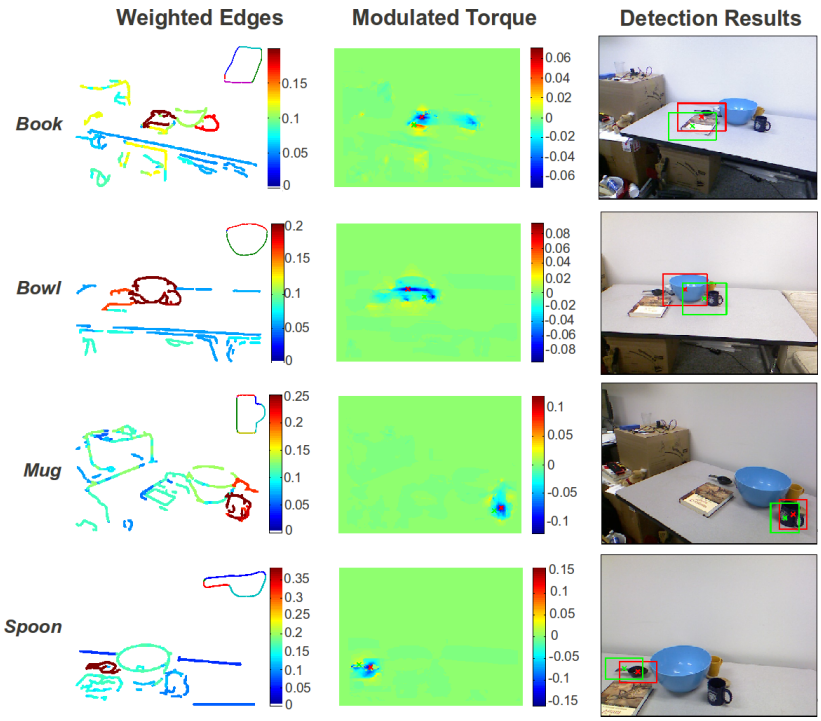


Figure 3.24: Detection results using depth information for the four objects in the UMD-clutter dataset. (Rows) Target object class: (Top to Bottom) `Book`, `Bowl`, `Mug`, `Spoon`. (Columns) Left: $W_{D^{\tau}_{sc}}$ where red means higher values and target model contours at top-right, Middle: Modulated torque with depth constraint showing the top 2 object detections (red and green crosses), Right: RGB frames overlaid with detection results.

3.5 Conclusions

In this chapter, we have presented a Gestalt-based approach to contour-based categorical object recognition that uses the image torque for the selection and grouping of specific target 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 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 border 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, 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, with state-of-the-art performance compared to other contour-based approaches.

The ability to recognize categories of objects using their shape information is just one way, however, of using Gestalt for higher-level visual tasks. Besides shape, *symmetry* and *functionality* are two innate attributes that invoke Gestalt-based recognition of objects. We discuss how these cues can be exploited to solve the FGO problem in the next two chapters.

Chapter 4: Detecting and Segmenting Symmetrical Regions

Symmetry, as one of the key components of Gestalt theory, provides an important mid-level cue that serves as input to higher visual processes such as segmentation. In this chapter, we propose a complete approach that links the detection of: 1) reflection (bilateral) and 2) curved reflection symmetries to produce symmetry-constrained segments of structures/regions in real images with clutter.

For detecting bilateral symmetry, a two stage approach that first detects putative symmetrical locations followed by a more expensive localization step is proposed[1]. We evaluate and compare our bilateral symmetry detector with the state-of-the-art feature based detector of Loy and Eklundh (Loy-Eklundh) [184] over two datasets: 1) Penn State University 2011/2013 symmetry competition datasets (PSU 2011/2013) and 2) the UMD Symmetry dataset. Extensive experiments with various ablations of the approach show that it is able to retrieve more precise bilateral symmetries compared to Loy-Eklundh over most recall values.

For curved symmetry detection, we leverage on patch-based symmetry features to train a Structured Random Forest (SRF) [140] classifier that detects multi-

[1]The detection of bilateral symmetry is currently under revision as [272]. Joint work with Hyoungjune Yi. Full results, datasets and code are available at http://www.umiacs.umd.edu/~cteo/object_symmetry/

scaled curved symmetries in 2D images. Experimental evaluations over two datasets: 1) SYMMAX-300 [278] and 2) NY-Roads [249] show that our SRF-based curved symmetry detector outperforms two state-of-the-art curved symmetry detectors [159, 278] with comparable performance to that of [249].

Next, using these symmetries, we modulate a novel symmetry-constrained foreground-background segmentation by their symmetry scores so that we enforce global symmetrical consistency in the final segmentation. This is achieved by imposing a pairwise symmetry prior that encourages symmetric pixels to have the same labels over a MRF-based representation of the input image edges, and the final segmentation is obtained via graph-cuts. Experimental results over four publicly available datasets containing annotated symmetric structures: 1) SYMSEG-300, 2) BSD-Parts, 3) Weizmann Horse (both from [159]) and 4) NY-roads [249] demonstrate the approach's applicability to different environments with state-of-the-art performance[2].

4.1 Introduction

Symmetry is a universal invariant, an innate attribute, that is ubiquitous and very common in nature. In Gestalt psychology, symmetry is considered as one of the key grouping or *mid-level* cues for explaining human visual perception and its detection facilitates early visual processes such as figure-ground segmentation [53].

[2]The detection of curved symmetry and symmetry-constrained segmentation was published as [270]. Code, data and more results are available at http://www.umiacs.umd.edu/~cteo/SymmetrySegmentation/

From a biological viewpoint, it is well-known that humans are extremely sensitive to reflection symmetry [39, 280]. There is also evidence that humans know about the symmetry of a figure, prior to the onset of recognition. Analysis of eye movement patterns by [182] found that symmetrical patterns are scanned with fixations that mostly fall on only one side of the symmetry axes.

In this chapter, we present approaches for detecting: 1) reflection (bilateral) symmetry and 2) *curved* reflection symmetry. For bilateral symmetry, we are concerned with the detection of larger, *global* symmetry patterns, often associated with foreground objects. On the other hand, curved reflection symmetries consider reflection symmetries that are much more *local* in terms of spatial scale. Such a concept is known in the literature as ridges, ribbons or centerlines [177,249], and is related to the classical Medial Axis Transform of Blum [19]. Equally important is the extraction of *symmetrical regions* that support these symmetries, and a key requirement is that the final extracted regions (segments) has to be symmetric as well. We consider both of these issues together in this chapter and present a complete approach for detecting symmetries and segmenting such symmetrical regions.

4.2 Related Works

4.2.1 Symmetry detection

The detection of various types of symmetry (bilateral/reflection, rotational and translation) from 2D images has a long history in computer vision. See [181] for an up-to-date survey of past and current techniques. The classic voting approach of

"Generalized Symmetry Transform" (GST) of Reisfeld et al. [230] is now largely surpassed by the feature-based method of Loy and Eklundh [184] that used symmetrical SIFT keypoint descriptors [183] for more robust detection of bilateral and rotational symmetries. GST, however, is still widely used, e.g [142, 169]. The main drawback of GST is that because it compares pointwise orientations, it is computationally expensive and fragile, and it works well only in relatively clean images with little noise and textures. By contrast, Loy-Eklundh detects standard SIFT keypoints and derives "mirrored" SIFT counterparts for each key-point. Matches based on mirrored counterparts result in local symmetry axes "particles" that are then used to vote (via their location and orientation) in Hough space. The local maxima within the space, selected via an adaptive threshold method, are therefore indicative of the most consistent symmetry axes associated with the image. An important limitation of [184] is that its performance is heavily dependent on the detectability of SIFT features in the image. Since SIFT keypoints are essentially histograms of local edge orientations, textureless or shadowed regions will not generate a sufficient amount of keypoints. A recent extension by Lee and Liu [158] matched these descriptors within a 3D axis parameter space to detect curved reflection symmetries from keypoints.

Instead of using keypoints, Tsogkas and Kokkinos [278] used Multiple Instance Learning to train a curved symmetry detector, that combines multiscale patch-based feature histograms of intensity, color, texture and spectral cues to obtain state-of-the-art detection performance on a large dataset of real images with clutter. Along similar lines, [35] creates reflected copies of local features for training a symmetry detector based on spectral features.

In the domain of biomedical imaging, most works had focused on detecting centerlines of 3D tubular/cylindrical structures: blood vessels, axons, dendrites and spinal columns [86, 155, 286], and in cartography symmetry was used for detecting road networks [114]. Although these works produce very good centerline predictions, their applicability is often limited to the specific imaging modality (e.g. CT, MRI or brightfield) and the expected size (scale) of the target tubular structures. Recently, Sironi et al. [249] proposed a novel regression-based technique using Regression-Trees (as opposed to classification) that showed state-of-the-art centerline detections in different applications (medical and roads). Their method, however, requires training a large number of regressors to predict the expected scale and location of the tubular structures from the input.

Other methods, [214] and [144] proposed to detect the symmetry of a region based on the phase relationship in spatial harmonics. Gabor and log-gabor filter kernels are applied over the image; regions where the different symmetric frequency components are in phase indicate a possible symmetry axis. This method, however, is extremely sensitive to noise, and the choice of the kernel filter size affects both the scale and quality of the symmetry axes returned. [305] proposed a "Symmetry Distance" measure of shapes, defined by the mean square distance that transforms each point in the original shape to the new (symmetrical) shape. A smaller distance means the shape is more symmetric. The method, however, is dependent on a good selection of initial seed points, and an exhaustive search for various symmetrical shapes limits its applicability to simple scenarios.

4.2.2 Segmenting symmetrical regions

Most previous approaches [142, 170] considered segmentation as a separate, independent step from symmetry detection. [142] used local features to approximate a symmetry axis followed by the fixation-based segmentation method of Mishra et al. [199] to extract regions with no symmetry constraints.

Riklin-Raviv et al. [236] embed symmetry cues dynamically into a level-set functional so that each evolution of the functional improves the symmetric properties of the current segmentation. Like most variational methods, however, the approach requires several iterations and could be stuck at a local minima.

Sun and Bhanu [264] use a region merging approach where homogeneous regions, measured in terms of color and texture, are merged while preserving reflection symmetry. The merging process, however, is sensitive to large variations of color and textures, producing oversegmented (small) regions in these areas. Along similar lines, Levinshtein et al. [166] build an adjacency graph that encodes how superpixels are grouped into symmetrical parts. Lee et al. [159] extend this approach by imposing a more general deformable disc (ellipse) model that encodes affinity of superpixels in curvilinear structures better. Affinity is computed from shape similarity (parameters of the deformable ellipse) and differences in local color and intensity. Since superpixels are grouped in a pairwise manner via dynamic programming, this approach is limited to grouping homogeneous regions that contain a single curved symmetry (no branches).

Fu et al. [77] focused on extracting foreground salient objects exhibiting re-

flective symmetry by first computing a symmetry foreground segmentation map from color-contrast cues and feature-based symmetry-induced homography, which are then set into unary and pairwise terms in a MRF-based segmentation. The accuracy of the final segmentation, however, depends primarily on the initial foreground map, which is formed by combining an feature-based estimate of a global reflection homography with a saliency foreground map based on color-contrast.

In the medical imaging literature, most works consider segmentation as an integral part of centerline detection, where the centerlines and their corresponding radii are solved by the same detector [86, 155, 249]. The final segmentations are therefore a combination of circles or balls located along the centerlines, with no enforcement of global curved symmetry consistency.

4.2.3 Contributions of this work

Our approach, described next, considers the issues of symmetry detection and segmentation of symmetrical regions together so that we produce accurate segmentations of symmetrical structures from robust symmetry detections in clutter.

Compared to Loy-Eklundh that requires textured regions for computing SIFT keypoints, our proposed two-step bilateral symmetry detector uses Gestalt principles to first detect putative symmetrical locations, a symmetry "attention" map, by comparing histograms of Gabor edge responses. Although histograms have better tolerance to noise, since they only encode the number of edges within a discrete set of orientations, they are by no means sufficient to determine the *precise* location, scale

and orientation of the symmetry, because spatial information is lost. In the second step, termed the symmetry "refinement" step, we estimate the location (centroid) and orientation of the supporting symmetry axis within the potential symmetry patches. As a direct 2D search over the image patch over all locations and orientations is computationally expensive, we show that it is possible to decompose the problem into two separate 1D searches, the first in edge orientation space and the second, in the (rectified) location space. To achieve robustness, we use kernel density estimates of edge orientations and edge counts in the first and second 1D search, respectively; and compare their probability distributions using the efficient L1 Earth Movers distance (EMD-L1) measure of [178]. Combining both searches admits a small set of local minima in each image patch that suggests likely locations of possible symmetry axes. We then score each axis via a method similar to Loy and Eklundh [184] by matching local symmetric features using a Hough voting technique. The symmetry axis with the best (highest) score is selected as the final symmetry axis of the image patch.

Compared to other curved symmetry detectors [159, 249, 278], our SRF-based approach learns to associate multiscale symmetrical features with a novel symmetry output annotation structure. The key advantage is that we let the SRF determine from training exemplars the optimum feature combination that predicts the best symmetry axes (location, orientation and scale) without a need to predefine a symmetry or noise model, enabling our approach to work in a large variety of environments and conditions. Additionally, as inference using SRF is extremely efficient, our curved symmetry detector runs in 0.1s (after feature extraction which

takes ≈1min) per 320×240 image.

Compared to works that apply grouping and merging of superpixels or regions [159, 166] or those that iteratively improve the final symmetry segmentation [236, 264], our segmentation approach is not only faster but in addition is able to handle symmetry axes with multiple branches. This is achieved through a foreground-background segmentation of structures/regions supporting these predicted curved symmetries via the addition of a novel pairwise symmetry prior in a Markov Random Field (MRF) representation of the input image edges. Since the symmetry prior is defined locally in the MRF clique, the optimal segmentation can be solved using graph-cuts [27], while handling even convoluted curved symmetry axes with multiple branches. As the predicted symmetries provide an initial measure of how symmetric the region should be, we modulate this prior so that the appropriate amount of symmetry is enforced in the final segmentation, a crucial requirement for natural images that can exhibit approximate symmetries at different scales.

4.3 Robust bilateral symmetry detection

As noted earlier, our bilateral symmetry detector consists of two related steps: 1) generating a "symmetry attention" map to obtain putative symmetry fixation points from which we extract initial object-like segments and 2) a symmetry "refinement" step that localizes the symmetry axes per segment. From an image I, we extract an edge map, I_e, which is used as input to the symmetry computation. In our implementation, we use Gabor filters, and the gPb edge detector [6], and

we retain edges larger than a threshold of 0.07. The output is a set of T symmetry axes $\mathcal{A} = \{A_1, \ldots, A_T\}$ with $A_i = \{(x_i, y_i), \theta_i, v_i\}$ (centroid location, orientation and symmetry axis score) for the i^{th} symmetry axis. We describe these two steps next.

4.3.1 Symmetry Attention

The goal of the symmetry attention stage is to detect efficiently regions in the image that are likely to support a reflection symmetry axis. From the input edge map, I_e, we first determine a symmetry "attention map" (§4.3.1.1), D_{sym}, with the same dimensions as I_e. From this map, we then extract a set of G fixation points (local maxima) which we denote as $\mathcal{F}_{sym} = \{f_1, \ldots, f_G\}$. We assume that each f_m is supported by a unique symmetric region, r_m, which we obtain via a (fast) variant of the fixation-based segmentation (§4.3.1.2) of Mishra et al. [199]. The method computes a closed boundary surrounding f_m, and this boundary contour likely is associated with an object in the image. These "object-centric" segments will then be used in the refinement stage (§4.3.2) to localize the precise locations of the symmetry axes.

4.3.1.1 The symmetry attention map

We introduce a fast and robust technique for detecting approximate symmetries in the image via the formation of an attention map. This map localizes potential (noisy) symmetry locations. This is similar in spirit to computing a symmetry "saliency" map of [77, 142], but our approach differs in the representation

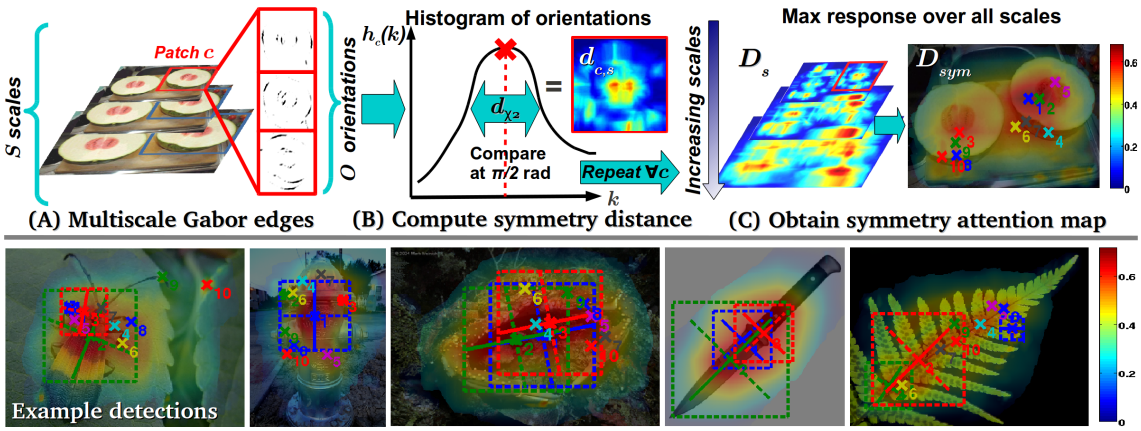


Figure 4.1: (Top) Generating the symmetry attention map: (A) Oriented Gabor edges are detected over several scales, (B) At an image patch c (boxed), we compare histograms $h_{c,s}(k)$ via d_{χ^2} to measure local symmetry, and (C) Normalized detections over scales, D_s, are combined to form a final attention map D_{sym}: larger values indicate stronger localized symmetry. (Bottom) Example top three maxima showing scales (boxes) and orientations.

used: oriented gabor histograms, and speed via integral images [46]. While most symmetry detection methods based on edges consider both the edge locations and their orientations and evaluate perfect symmetry, our method only considers the orientation of edges. Given a possible symmetry axis, the method compares the probability distributions of edges on the two sides of the axis and thus is robust to image deformations and errors.

The steps for generating the attention map are illustrated in Fig. 4.1. Given an input image, I, oriented Gabor edges $I_{e,s}$ are detected over $s \in S = 4$ scales and $O = 16$ orientations. At each scale, s, we compute a histogram $h_{c,s}(k)$ of the orientations of Gabor edges $\mathcal{Q}_p = \{q_1, \ldots, q_n\}$ within a patch c centered at pixel $(x_c, y_c) \in I_{e,s}$ (Fig. 4.1 (A)):

$$h_{c,s}(k) = \#\{\angle\{q_1, \ldots, q_n\} \in \text{bin}(k)\} \tag{4.1}$$

where $\text{bin}(k)$ denotes an orientation bin in the histogram centered at patch c, with $k \in [0, \pi]$ radians. To check for local symmetry at a patch, we check for every possible orientation of the symmetry axis (for O discrete orientations) whether their appropriately adjusted histograms match. Specifically, for the $o \in O$ orientation, we select bins from opposing angles $b_o = h_{c,s}(\{k_1, \ldots, k_{o-1}\})$ and its symmetric counterpart $b'_o = h_{c,s}(\{k_O, \ldots, k_{o+1}\})$ to compute their χ^2 distance:

$$d_{\chi^2}(o) = \chi^2\left(b_o, b'_o\right) \tag{4.2}$$

For example, to detect horizontal symmetries (parallel to the x axis), we would compare bins ranging from $(0, \pi/2]$ and $(\pi, \pi/2]$ (Fig. 4.1 (B)). When the bins on each side are similar, the χ^2 distance would be small, indicating the presence of a

strong symmetry. The final symmetry distance measure, $d_{c,s}$, of a patch is defined as:

$$d_{c,s} = 1 - \underset{o \in O}{\operatorname{argmin}}(d_{\chi^2}(o))/n_c, \tag{4.3}$$

where we select the minimum $d_{\chi^2}(o)$ (and corresponding orientation) normalized by n_c, the number of edges in the current patch. This process is repeated over all pixels in $I_{e,s}$, generating a symmetry attention map D_s that codes the symmetry measure at scale s. The final symmetry attention map, D_{sym}, combines all the different scales into one map. At each point we extract the maximum response over all scales: $D_{sym} = \operatorname{argmax}_{s \in S} D_s$, which yields a measure of symmetry *confidence* at the particular pixel. The set of "fixation points", $\mathcal{F}_{sym} = \{f_1, \ldots, f_G\}$, shown as crosses in Fig. 4.1 (C) and (Bottom) are obtained as the maxima (using non-maxima suppression) applied over D_{sym}. We show the top two orientations per point (usually separated by $\pi/2$ radians) with the bounding box representing the best scale that supports this symmetry.

For an efficient implementation, we use, similar as in [209, 269], the method of summed area tables (integral images). Specifically, since each patch c consists of sums of Gabor edges, we precompute the sum of edges for each possible orientation $o \in O$ at each location in the image in summed area tables. With these summed tables of Gabor responses per orientation, R_o, we then can compute the total Gabor response as an orientation histogram, $h_{c,s}(o)$, for one orientation within a rectangular patch c of size $W_c \times H_c$ (width, height) and centered at pixel (x_c, y_c) as the

sum/difference of four tables as:

$$h_{c,s}(o) = R_o(x_c - \frac{W_c}{2}, y_c - \frac{H_c}{2}) + R_o(x_c + \frac{W_c}{2}, y_c + \frac{H_c}{2}) \\ - R_o(x_c + \frac{W_c}{2}, y_c - \frac{H_c}{2}) - R_o(x_c - \frac{W_c}{2}, y_c + \frac{H_c}{2}) \tag{4.4}$$

Repeating eq. (4.4) O times for each R_o allows us to compute the orientation histogram of a patch per scale, $h_{c,s}$, in constant time.

4.3.1.2 Fixation-based segmentation

Given the set of G symmetry fixation points $\mathcal{F}_{sym} = \{f_1, \ldots, f_G\}$, we seek to determine at this step the set of G supporting regions $\mathcal{R}_{sym} = \{r_1, \ldots, r_G\}$ associated with every fixation point. Specifically, for the m^{th} fixation point $f_m \in \mathcal{F}_{sym}$, we want to determine the region, $r_m \in \mathcal{R}_{sym}$, with the appropriate size that would explain the putative symmetry centered at f_m. Extracting r_m has three key purposes. First, it suggests an appropriate *scale* that the potential symmetry axis should be detected at the refinement stage. As we have argued, this is a critical aspect that has been largely ignored by the majority of past works but is key to the proper definition of symmetry. Secondly, since we are interested in *object-centric* symmetries in the image, we require a procedure that is able to extract, in addition to the right scale, also a region that is suggestive of a potential object. In this work, we assume that an object-like region must satisfy the simple criterion that most of the edges must be closed via the Gestalt principle of *closure* so that an appropriate neighborhood that surrounds the fixation point f_m is found. Thirdly, by limiting the refinement step to within the set of regions in $\mathcal{R}_{sym}$, we significantly reduce the search space

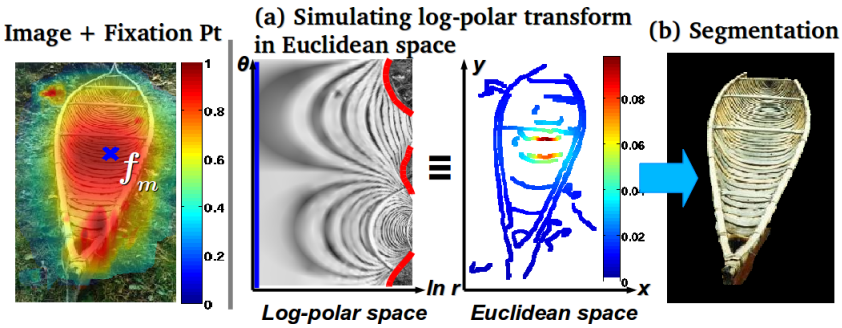


Figure 4.2: Segmenting a region given the symmetry fixation point f_m (blue cross). (a) Weighing edges by $\kappa = \frac{1}{d_q}$ with respect to f_m (red means larger κ) simulates a log-polar transformation of the image. (b) The final segmentation.

for detecting the symmetry axis since each $r_m \subseteq I$. This speeds up the approach compared to an exhaustive search over the entire image and reduces the number of false positives as well.

We use a fixation-based segmentation procedure that is similar to the method proposed by Mishra et al. [199] as it seeks to segment a closed region that surrounds the input fixation point. We use a log-polar transformation, instead of a polar transform. In Appendix C we show that we can simulate the transformation into the log-polar coordinates by weighing the pairwise terms in the standard graph-cuts energy function [28] by a factor $\kappa = \frac{1}{d_q}$ where d_q is the Euclidean distance (in pixels) from the fixation center f_m to an edge point q (Fig. 4.2 (a)). Additionally, this saves computation time needed for the coordinate transformation, leading to a faster segmentation procedure. Finally, unlike [199] which uses color information as further pairwise constraints for the segmentation, we use only edge pixels from I_e. This provides a better segmentation for cases where there is large variation in color

information within the object itself.

Note that a straightforward alternative that uses the bounding box associated with each f_m to determine a region would be a rough approximation of the desired object segment. In §4.6.2.2 we show experimentally that using the segmentation improved the overall performance for images with single symmetric regions, but using the bounding box regions produced the best performance for multiple symmetric regions.

4.3.2 Symmetry Refinement

Having obtained the set of regions $\mathcal{R}_{sym} = \{r_1, \ldots, r_G\}$ from the symmetry attention map, the goal of the symmetry refinement step is to detect, for each $r_m \in \mathcal{R}_{sym}$ with dimensions $X_{r_m} \times Y_{r_m}$, the final symmetry axis $A_m = \{(x_m, y_m), \theta_m, v_m\}$ parameterized by its centroid, orientation, and symmetry score, respectively. Unlike approaches that use local feature matches (e.g. [184, 230]) to determine these parameters, we use a robust approach based on comparing *statistics* of the edges present in the image. A similar idea was developed in [263], where gradient orientation histograms were compared using a FFT based technique to find the direction of the orientation axis. In contrast to [263], our approach uses two steps of comparing edge statistics, to find both the orientation and the position of the symmetry axis, and it is not applied to the whole image, but searches over the regions provided in the attention step (§4.3.1). We first compare probability density functions (pdf)s derived using kernel density estimates of edge orientations, $p_d(\theta)$, then probability

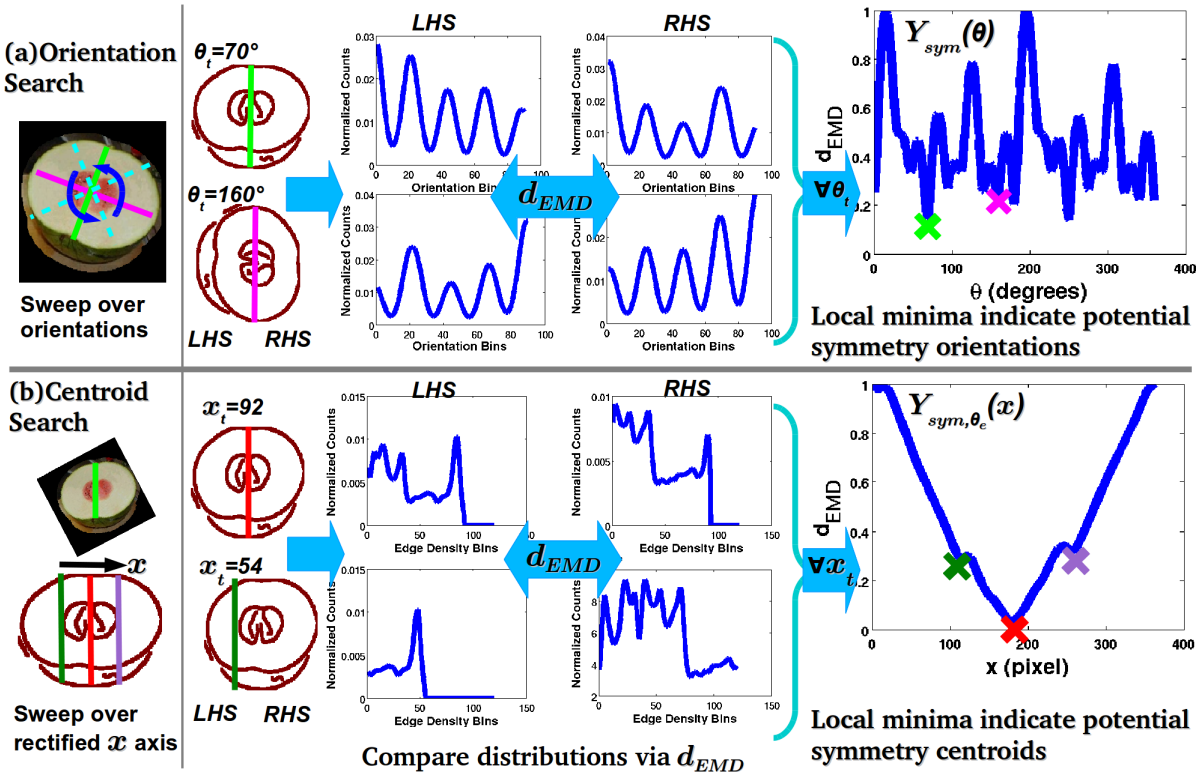


Figure 4.3: Overview of the refinement step. (a) Comparing pdfs of edge orientations $p_d(\theta)$ (middle) and their local minima marked by crosses (right). (b) Given a selected orientation, we compare pdfs of edge density $p_{d,\theta_e}(x)$ (middle) and the final set of possible symmetry axes as local minima as crosses (right).

distributions of edge counts, $p_{d,\theta}(x)$, over the rectified image at angle θ, as shown in Fig. 4.3. This effectively reduces the original (expensive) 2D search for the symmetry solution in each r_m into two separate 1D searches. A justification for the separation of orientation from translation (centroid location) in computing the reflection symmetry is given in Appendix D. We describe next these two steps in detail (§4.3.2.1 and 4.3.2.2) and how the final symmetry axis per segment is obtained (§4.3.2.3).

4.3.2.1 1D search over orientations

We proceed first by taking each r_m and dilating it by a small factor $\delta_e = 10$ pixels so that the expanded segment captures the necessary edges in I_e that are likely to support the symmetry. Let us denote the set of M edge points within the expanded segment as $\mathcal{W}_{r_m} = \{w_1, \ldots w_M\}$ and their corresponding edge orientations as $\Theta_{r_m} = \{\theta_1, \ldots, \theta_M\}$, with $\theta_j \in [0, \pi]$ radians. We then estimate the edge orientation pdf $p_d(\theta)$ using kernel density estimates as:

$$p_d(\theta) = \frac{1}{M\beta} \sum_{j=1}^{M} K_N \left(\frac{\theta - \theta_j}{\beta} \right) \tag{4.5}$$

where K_N is the standard normal kernel. β is the bandwidth parameter that we derived from the number of modes of the data. We determine from Θ_{r_m} which orientations have significant (e.g. greater than $t_\theta = 50\%$ of the largest edge orientation bin in Θ_{r_m}) counts in the data to set a reasonable value for β. Next, by sweeping through the orientation space, we check at each test orientation, $\theta_t \in [0, \pi]$, whether the potential symmetry axis separates Θ_{r_m} into two distributions $p_d(\theta_t)$ and $p'_d(\theta_t)$ via an efficient implementation of the EMD-L1 distance measure [178][3]:

$$d_{EMD}(\theta_t) = EMD(p_d(\theta_t), p'_d(\theta_t)) \tag{4.6}$$

[3]Other statistical measures based on entropy, such as the Jensen-Shannon divergence [175], or the Bhattacharyya distance were tried but the EMD-L1 distance was found to give the highest accuracy.

where $\{p_d(\theta_t), p'_d(\theta_t)\}$ are derived from $p_d(\theta)$ as:

$$\begin{cases} p_d(\theta_t) = p_d(i)|_{i=\theta_t - \pi/2}^{i=\theta_t} \\ p'_d(\theta_t) = p_d(i)|_{i=\theta_t + \epsilon_\theta}^{i=\theta_t + \epsilon_\theta + \pi/2} \end{cases} \tag{4.7}$$

and $p_d(\theta) = p_d(\theta) \pm \pi$ due to the periodicity of the orientation angles. We set $\epsilon_\theta = \pi/180$ to obtain a search resolution of 1 degree. Sweeping through the orientation space gives us a 1D score function defined over θ, $Y_{sym}(\theta)$, that codes the symmetrical distance d_{EMD} at each evaluated θ (Fig. 4.3 (a)). The set of J local minima obtained from $Y_{sym}(\theta)$, $\mathcal{Y}_\theta = \{\theta_1, \ldots, \theta_J\}$ represents the top J orientations for the potential symmetry axis within r_m. We therefore focus our search for the centroid of the symmetry axis only along these J orientations.

4.3.2.2 1D search over centroid locations

For each $\theta_e \in \mathcal{Y}_\theta$, we search for the centroid location of the best symmetry axis in the following manner. First, we rectify the expanded segment r_m along θ_e so that the axis becomes parallel to the image y axis, and we only need to search along the x axis to determine the centroid location. Next, we estimate via kernel density estimates, similar to eq. (4.5), a pdf $p_{d,\theta_e}(x)$, that captures the edge counts along the rectified x axis as

$$p_{d,\theta_e}(x) = \frac{1}{M\beta_x} \sum_{j=1}^{M} K_N \left(\frac{x - \xi_j}{\beta_x} \right) \tag{4.8}$$

where each $\xi_j \in \Xi_x = \{\xi_1, \xi_2, \ldots, \xi_M\}$ are the rectified x coordinates of the M edges in r_m, and β_x is the bandwidth of the normal kernel K_N which we set as

$\beta_x = X_{r_m}/100$. Given a test location, x_t, the symmetry distance score between the distributions at x_t: $\{p_{d,\theta_e}(x_t), p'_{d,\theta_e}(x_t)\}$, is again obtained via the EMD-L1 distance:

$$d_{EMD}(x_t) = EMD(p_{d,\theta_e}(x_t), p'_{d,\theta_e}(x_t)), \tag{4.9}$$

and their distributions are derived from $p_{d,\theta_e}(x)$ as:

$$\begin{cases} p_{d,\theta_e}(x_t) = p_{d,\theta_e}(i)|_{i=x_t-R(x_t)}^{i=x_t} \\ \\ p'_{d,\theta_e}(x_t) = p_{d,\theta_e}(i)|_{i=x_t+1}^{i=x_t+1+R(x_t)} \end{cases} \tag{4.10}$$

$R(x_t)$ defines the size of the area from which we consider edge points. The same parameter can be viewed as the *optimal* support for the symmetry axis expected within r_m. This means that we search x_t within the range of $[R(x_t), X_{r_m} - R(x_t)]$. In practice, $R(x_t)$ is a fixed value that is learned from training data, or if this is not available, it can be a function that varies the support adaptively over different x_t: for example one can set $R(x_t)$ as a normal distribution $N(\mu_\Xi, \sigma_\Xi^2)$ over Ξ_x so that we bias a larger support at locations with the densest edge counts. By repeating eq. (4.9) over all test locations, we obtain a 1D score function defined over x, $Y_{sym,\theta_e}(x)$, that codes the symmetrical distance d_{EMD} at each evaluated x (Fig. 4.3 (b)). The set of L local minima obtained from $Y_{sym,\theta_e}(x)$: $\mathcal{Y}_x = \{x_1, \ldots, x_L\}$ are therefore the top L centroid locations (after de-rotating the image) per θ_e: $\mathcal{Y}_{(x,y)} = \{(x,y)_1, \ldots, (x,y)_L\}$.

To summarize, at the end of the two 1D search procedures, we obtain for each of the J orientation minima, a set of top L centroid locations. This results in a combined set of $J \times L$ potential axes per r_m: $\mathcal{Y}_A = \{A_1, \ldots A_{J\times L}\}$.

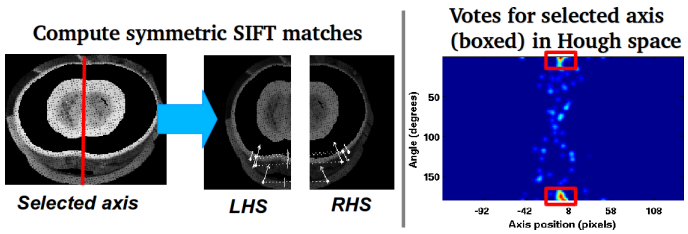


Figure 4.4: Using a robust Hough-voting technique to score an axis. (Left) Dilated and rectified LHS and RHS segments. (Right) Hough space with the scoring region boxed.

4.3.2.3 Scoring the symmetry axes

Given the set of top $J \times L$ potential symmetry axes for each r_m, in the final step we associate an appropriate symmetry score per $A_l \in \mathcal{Y}_A$. We use a robust Hough-based voting method derived from matching localized symmetric SIFT features of Loy and Eklundh [184] (Fig. 4.4). First, we rectify the edge image I_e by a_l so that a_l is now parallel to the y axis and is in the center of I_e. Next, we extract from the rectified edge image the relevant edges captured by the r_m (rectified according to a_l as well). This creates an edge map $I_e(r_m)$ that is a subset of all edges in I_e. We then dilate $I_e(r_m)$ by a factor $\delta_s = 20$ pixels so that a larger region surrounding the edges are used in detecting and matching symmetrical SIFT features. Each match between symmetrical features is a vote in the *linear* Hough space $H(\theta, x_c)$ (angle, x position of axis with respect to the image center) for a potential orientation and location of a symmetry axis. Unlike [184] that uses $H(\theta, x_c)$ directly to determine the best symmetry axes with the largest votes, our goal here is to obtain the same (normalized) voting score for the selected axis a_l. Doing this is extremely easy,

since $I_e(r_m)$ is rectified and centered with respect to a_l. The solution is the point located at $H(0,0)$ in the Hough space. In order to account for discretization effects in the Hough space, we take a small region of size $(\delta_H \times \delta_H)$ with $\delta_H = 5$ pixels surrounding $H(0,0)$ to obtain a mean normalized score per a_l: v_l. This procedure of scoring therefore affords us with a robust and symmetry sensitive score that is directly comparable with the scores used in the baseline [184] (§4.6.2.2).

Finally, for each r_m, we select the symmetry axis $A_m = \{(x_m, y_m), \theta_m, v_m\}$, that yields the largest symmetry score $v_m = \text{argmax}_v\{v_1, \ldots, v_{J\times L}\}$ from the $J \times L$ axes that were compared. Repeating the above procedure for each of the G segments in $\mathcal{R}_{sym}$ yields the final set of $T \leq G$[4] axes: $\mathcal{A} = \{A_1, \ldots, A_T\}$. The symmetry score, v_m will also be used in §4.5 to modulate symmetry prior for extracting the final symmetrical segments.

Further implementation details for the bilateral symmetry detector can be found in Appendix E, including its training and run-time evaluations. In the next section, we turn our attention to the detection of curved reflection symmetry using SRF.

4.4 Fast curved symmetry detection via SRF

Similar to the detection of bilateral symmetry, the input to our SRF-based curved symmetry detector is a RGB image I and the output is a set of n curved symmetry axes, $\mathcal{A}_c = \{A_{c1}, A_{c2}, \cdots, A_{cn}\}$. We describe the features used and the

[4] $T \leq G$ is due to an extra segment combination step that combines similar segments together.

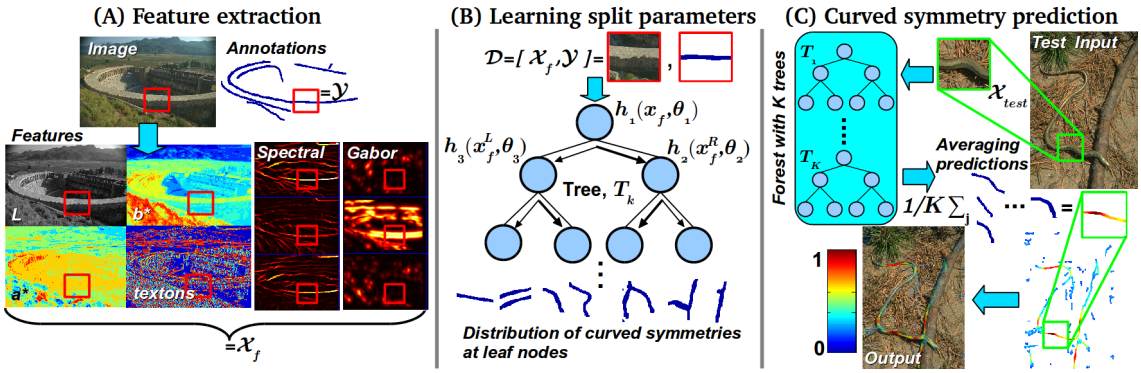


Figure 4.5: Training a SRF for curved symmetry detection. (A) Multiscale intensity, color, texture, spectral and oriented Gabors are used to compute a set of local symmetry responses, $\mathcal{X}_f$, by comparing histograms of patches. (B) By pairing patch-based features $x_f \in \mathcal{X}_f$ with their symmetry groundtruth annotations $\mathcal{Y}$, we determine the optimal split parameters θ associated with the split functions $h(x_f, \theta)$ that send features x_f either to the left or right child. The leaf nodes store a distribution of structured labels of symmetry axes. (C) During inference, a test patch is assigned to a leaf node within a tree that contains a prediction of the location and scale of the symmetry axis. Averaging the prediction over all K trees yields the final symmetry axes and their corresponding strengths (degree of symmetry).

training procedure next.

4.4.1 Patch-based symmetry features

In order to detect curved symmetries/centerlines in real images with clutter, a key requirement is the ability to efficiently extract robust features from the input image that are suggestive of symmetry. Our feature selection approach is motivated by two issues well-known in visual symmetry. First, similar to textures (which is a kind of translation symmetry), curved reflection symmetry is a function of image *scale*. Second, and related to the first is what features can one use to *define* symmetry in the image? In this work, we extract multiscale features based on intensity, color (from L^*, a^* and b^* channels), orientated Gabor edges and texture features [193] by comparing patches with different orientations (we use 8 discrete orientations) densely in the image (Fig. 4.5 (A)). The reason is that such features capture different forms of symmetry information that are complementary, e.g. edge based features can suggest symmetry at textureless regions. For efficiency, we adopt the integral image implementation of [278]. For each patch, we compare the empirical distribution of feature histograms using the robust EMD-L1 distance [178] where a small value suggests a region with strong symmetry. In addition to these local features, we compute symmetric *spectral features* proposed by [278]. These are similar to the intervening contour cue of [193] except that curved symmetry responses from histogram comparisons above are used to construct the affinity matrix prior to extracting the eigenvectors using normalized-cuts [245]. The output $\mathcal{X}_f$ is a set

local symmetry responses over multiple scales (we use 4 scales here) for each of the 6 feature channels considered.

4.4.2 Symmetry detection via SRF

In this work, we train a SRF in a similar fashion as border ownership (§2.3.2). We use patch-based symmetry responses of size $N \times N$, $x_f \in \mathcal{X}_f$ as features, and binary structured labels of groundtruth curved symmetries $\mathcal{Y} = \mathbf{1}^{N \times N}$ The goal of training the SRF is to learn, for the i^{th} internal (split) node, the optimal splitting parameters θ_i for each binary split function $h(x_f, \theta_i) \in \{0, 1\}$. If $h(\cdot) = 1$ we send x_f to the left child and to the right child otherwise (Fig. 4.5 (B)). $h(x_f, \theta_i)$ is an indicator function with $\theta_i = (d, \rho)$ and $h(x_f, \theta_i) = \mathbf{1}[x_f(d) < \rho]$, where d is the feature dimension of one of the input features described above. ρ is based on maximizing a standard information gain criterion M_i (eq. (2.4)) that splits the input data $\mathcal{D}_i \subset \mathcal{X}_f \times \mathcal{Y}$ at node i into $\mathcal{D}_i^L$ (left child) and $\mathcal{D}_i^R$ (right child) respectively. As was noted in [51], computing eq. (2.4) using structured $\mathcal{Y}$ is more feasible if one imposes an intermediate mapping function $\Pi : \mathcal{Y} \mapsto \mathcal{B}$ of structured labels onto the discrete labels $b \in \mathcal{B}$. The number and type of discrete labels, $|\mathcal{B}|$, is an empirical measure of the *diversity* of structured curved symmetries that we expect to encounter. To determine Π, we first apply an Expectation-Maximization (EM)-based clustering over DAISY [274] descriptors from randomly sampled symmetry patches from $\mathcal{Y}$. The final clusters obtained are then used to define $\mathcal{B}$. The process is repeated with the remaining data $\mathcal{D}^o, o \in \{L, R\}$ at both child nodes until M_i

Notation	Description	Value
-	Number of feature orientations	8 [0 to π]
-	Number of feature scales	4 ([0.1, 0.3, 0.5, 0.75] of image diagonal)
-	Number of feature channels	6 [L^*, a^*, b^*, textons, spectral, Gabor]
N	Patch size	16
-	Number of [positive/negative] training samples per dataset	$[10^5/10^5]$
$\|\mathcal{B}\|$	Size of structured labels	150
K	Number of trees	16
h_d	Maximum tree depth	64
-	Minimum value of M_i	10^{-10}
-	Minimum length of A_c (pixels)	5
-	Minimum symmetry response, $A_c(r)$ per pixel $r \in A_c$	0.01

Table 4.1: Parameters used in `SRFSym`.

is below a fixed threshold or a desired tree depth h_d is reached. The leaf nodes at each T_k store a distribution of curved symmetry labels encountered during training. Inference using SRF is straightforward (Fig. 4.5 (C)). We sample test patches densely over the image to obtain test features, $\mathcal{X}_{\text{test}}$, and pass them into the SRF to produce a structured prediction of the symmetry axes per decision tree T_k. Averaging these responses over all K trees produces the final curved symmetry predictions. We then convert these predictions (a continuous value) into a set of curved symmetry axes: $\mathcal{A}_c = \{A_{c1}, A_{c2}, \cdots, A_{cn}\}$, where each $A_c \in \mathcal{A}_c$ is defined as a contiguous single-pixel wide segment. Notably, these responses can be seen as an estimate on the symmetry strength of the test patch, denoted as $A_c(r)$ for a pixel $r \in A_c$, and we use it to modulate the amount of symmetry to enforce in the segmentation step, described in §4.5. The parameters used for training the SRF are summarized in Table 4.1.

4.5 Symmetry-constrained segmentation using graph-cuts

We embed symmetry constraints via a modified Markov Random Field (MRF) representation over the binary image *edge map* (Fig. 4.6 (top)), I_e, derived from gPb [6] (§4.3) or SE [51] where we retain responses >0.07 or >0.03 respectively. Using I_e is important here as it ensures that the segmentation results obtained are *not* influenced by color or intensity similarity but only by the detected symmetry axes, which is our goal. Since this approach works for both bilateral symmetries A (§4.3) and curved symmetries A_c (§4.4), for simplicity and clarity, we will denote both kinds of symmetry axes as A in our descriptions.

Each node in the MRF is a pixel in I_e, with links (graph-edges) between nodes denoting the local relationship between connected pixels. Pixels that are directly connected with one another form a local neighborhood or clique. In addition to the unary and pairwise terms over links in a standard 4-way neighborhood clique system, we add in a link at each node that connects, based on the detected set of curved symmetry axes, $\mathcal{A}$, their closest *symmetrical* neighbor. To do this, we first compute its distance transform, $D_{\mathcal{A}}$. Next, pixels that lie on the same iso-contours on opposite sides of each $A \in \mathcal{A}$ are linked (Fig. 4.6 (below)). This additional link, called the *cross-symmetry* term, creates a new 5-way neighborhood clique system that enforces both local (4-way) and global curved symmetry constraints within a single MRF model. This ensures both global symmetrical consistency while allowing for small local deformations in the final segmentation. In addition, since this term is computed locally with respect to A, our model handles multiple axes and branching

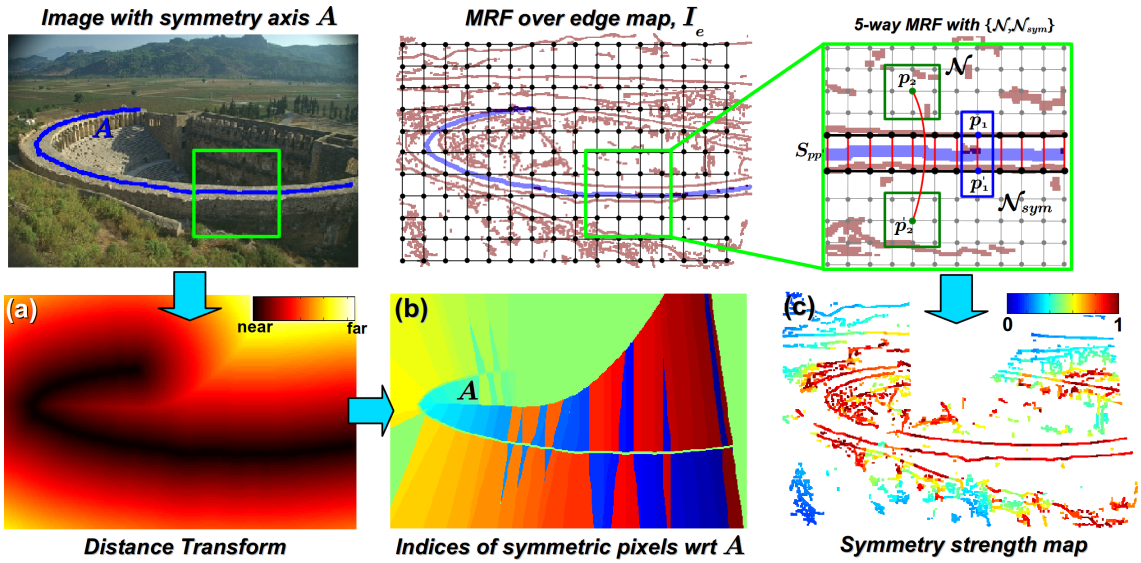


Figure 4.6: Symmetry-constrained segmentation. (Top) Constructing the 5-way MRF over I_e with cross-symmetry terms, $S_{pp'}$ given a (curved) symmetry axis, A. Every node in the MRF consists of $\mathcal{N}$ (4-way, green box) and $\mathcal{N}_{sym}$ (cross-symmetry, blue box) neighbors. We detail two symmetric neighbors $\{p_1, p'_1\}$ and $\{p_2, p'_2\}$ with their corresponding cross-symmetry terms as red links. Note that not all $S_{pp'}$ are shown for clarity. (Below) Computing the symmetry prior: (a) Given A, we compute its distance transform. (b) We then link the closest pixels along the same iso-contours on opposite sides of A to form symmetric pairs. (c) Visualization of the symmetry strength used in $es_{pp'}$, with red denoting stronger symmetries.

symmetries with no additional modifications. Finally, as the 5-way MRF model retains a local clique neighborhood system, the optimal labeling can be efficiently obtained using standard graph-cuts. We use the popular max-flow/min-cut toolbox of Kolmogorov and Zabih [139] in our implementation.

We detail the binary energy function E used here. Let $\mathcal{L} = \{0, 1\}$ be the labels of the background and the symmetrical region respectively. $\mathcal{P}$ is the set of all pixels in I_e, with $\{\mathcal{N}, \mathcal{N}_{sym}\}$ denoting the 5-way neighborhood clique system consisting of the 4-way pairwise neighbors (p, q) and cross-symmetry neighbors (p, p') respectively. The energy function is thus defined as:

$$E(f) = \sum_{p \in \mathcal{P}} U_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q) + \sum_{(p,p') \in \mathcal{N}_{sym}} S_{pp'}(f_p, f_{p'}) + \sum_{(p,q) \in \mathcal{N}} B_{pq}(f_p, f_q) \tag{4.11}$$

where $f_p \in \mathcal{L}$ is the label assigned to pixel $p \in \mathcal{P}$ and $f = \{f_p | p \in \mathcal{P}\}$ is the labeling of all the pixels in the image. The first two terms, $\{U_p, V_{pq}\}$ of eq. (4.11) are the standard unary and pairwise terms that encode the foreground prior and boundary information used in the majority of MRF-based segmentation approaches [28, 238]. For the unary term, instead of a foreground model derived from color or intensity information (which we do not have), we set pixels that overlap with any axis $\mathcal{A}$, $p_{\mathcal{A}}$, as foreground ($U_{p_{\mathcal{A}}}(0) = \infty$) and pixels along the image boundary, p_B, are set as background ($U_{p_B}(1) = \infty$). Similarly for V_{pq}, we replace image intensities used in [28] with their edge labels:

$$V_{pq}(f_p, f_q) = \begin{cases} \exp(-\frac{(I_e(p) - I_e(q))^2}{2\sigma^2}), \text{if } f_p \neq f_q \\ 0, \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{4.12}$$

so that the final segmentation aligns with I_e.

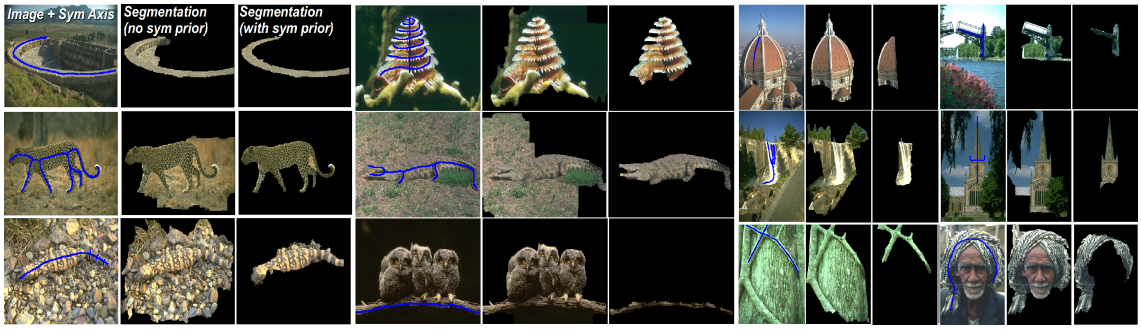


Figure 4.7: Example symmetry-constrained segmentations. Notice that we are able handle symmetry axes with multiple branches and produce more accurate segments with the symmetry prior term.

The third term is the *symmetry prior* term. This term sums up the cost for assigning different labels to symmetric neighbors (p, p'):

$$S_{pp'}(f_p, f_{p'}) = \begin{cases} es_{pp'}, & \text{if } f_p \neq f_{p'} \\ \beta, & \text{otherwise} \end{cases} \tag{4.13}$$

where $\beta < 1$ is a small positive value that provides a penalty when symmetrical neighbors are assigned similar labels. We set $\beta = 0.006$ for all results reported. $es_{pp'}$ is a measure of symmetry strength defined as:

$$es_{pp'} = 1 + \beta - \frac{1}{Z}\log(1 + \|D_{\mathcal{A}}(p) - D_{\mathcal{A}}(p')\| + \nu_{pp'}) \tag{4.14}$$

where $D_{\mathcal{A}}(p) \triangleq \min_{(p_a \in A)} \|p - p_a\|$ is the distance between pixel p and $p_a \in A$, its closest pixel along the symmetry axis obtained from the distance transform. $\nu_{pp'}$ is a symmetry score that depends on the type of symmetry predicted. For bilateral symmetry, $A = \{(x, y), \theta, v\}$ and we define $\nu_{pp'} = 1 - v$ based on the symmetry score v derived from §4.3.2.3. For curved symmetries, we define $\nu_{pp'} =$

$1 - (A(p_A) + A(p'_A))/2$ as the symmetry score predicted by the SRF, where we take the mean value of the two corresponding symmetry scores along A. Since $es_{pp'}$ is obtained by combining two estimates of symmetry, we tend to ameliorate the inherent noisy symmetric pixel correspondences caused by internal edges or textures in I_e. $Z = \max_{(p,p')\in P}(\log(1 + \|D_{\mathcal{A}}(p) - D_{\mathcal{A}}(p')\| + \nu_{pp'}))$ normalizes the second term in eq. (4.14) to $[0, 1]$ and as a result $es_{pp'}$ is in the range $[\beta, 1 + \beta]$. Since eq. (4.14) sets a large $es_{pp'}$ for pixels with different labels exhibiting *strong* symmetries, this encourages symmetrical pixels to have the same labels, and as a consequence, enforces symmetry in the final segmentation. Notably, as $es_{pp'}$ is derived from $D_{\mathcal{A}}$ and the SRF predicted symmetry scores, we are able to modulate the effect of this term, allowing for symmetrical and asymmetrical configurations to occur at appropriate locations. We show some example segmentation results in Fig. 4.7 comparing it to the case when no symmetry prior is used (a standard 4-way MRF).

The final term B_{pq} is a "ballooning" term, introduced by Veksler [282] that encourages the final segmentation to expand, in opposite directions along both sides of A so that a reasonably-sized symmetric segment is obtained. Without this term, the final segmentation tends to be small, as symmetry strengths are usually the largest between the closest (p, p'). Assuming that pixel p is further away from pixel

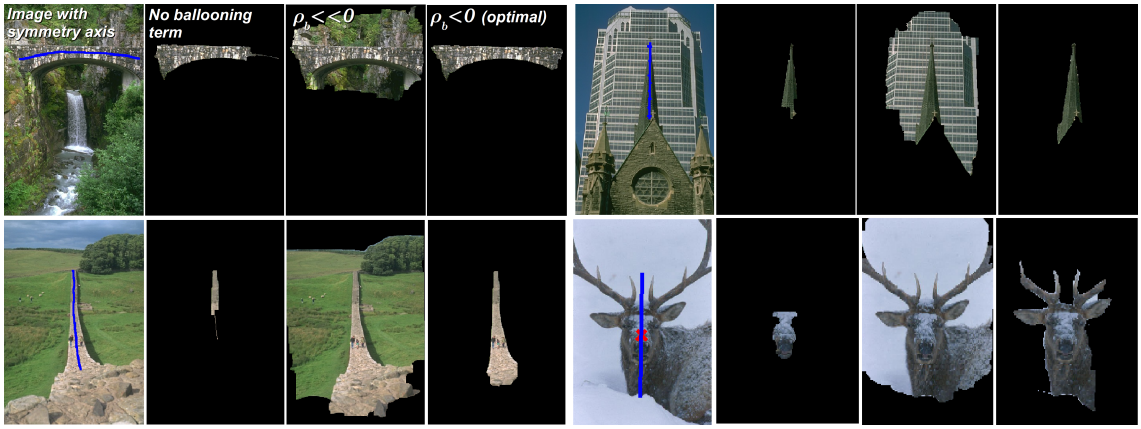


Figure 4.8: Effect of the ballooning term, B_{pq}. (L-R) Without B_{pq}, the final segmentation tends to be small. When $\rho_b \ll 0$, over-expansion occurs, resulting in a degenerate segmentation. Using an appropriate value for ρ_b produces an optimal segmentation.

q with respect to A, we have:

$$B_{pq}(f_p, f_q) = \begin{cases} 0, & \text{if } f_p = f_q \\ \infty, & \text{if } f_p = 1 \text{ and } f_q = 0 \\ \rho_b, & \text{if } f_p = 0 \text{ and } f_q = 1 \end{cases} \tag{4.15}$$

where ρ_b is a ballooning cost that we set to control the expansion of the final segmentation. Following [282], the value of ρ_b is usually set to a small negative value. However, when $\rho_b \ll 0$, over-expansion occurs resulting in a degenerate (and undesirable) symmetrical segmentation (Fig. 4.8). We use $\rho_b = -0.03$ in all experiments.

A note on the submodularity of all the pairwise terms in eq. (4.11). Clearly, V_{pq} and B_{pq} are submodular by construction. The symmetry prior term, $S_{pp'}$, is also submodular since by eq. (4.14), $es_{pp'} \geq \beta$ for all values of β. From [139], E can be minimized exactly via graph-cuts.

4.6 Experiments: Bilateral Symmetry Detection

4.6.1 Datasets, baseline and evaluation procedure

We use three datasets for the experiments. The first two, PSU 2011 and PSU 2013, are publicly available[5] while the UMD Symmetry dataset is new. Each dataset is separated into two different categories: 'singles' containing only one dominant symmetry and 'multiples' for images that contain multiple symmetric objects.

The PSU 2011 and PSU 2013 datasets consist of images taken under natural conditions and they contain reflection symmetries from a variety of different natural objects. For PSU 2011, we chose the 'real' subset (for real images) while ignoring synthetic images. For training, we used only the training set (35 for 'singles', 17 for 'multiples') from PSU 2013 to tune the parameters for the evaluation of *both* PSU 2011 and PSU 2013. For PSU 2011, we evaluated our results over the *training* subset (because it has more images): 79 (singles), 85 (multiples) while for PSU 2013, we used the testing subset: 40 (singles), 30 (multiples). Human annotated symmetry axes groundtruth are provided in both the testing and training subsets. The UMD dataset, which is the largest of its kind so far, consists of 107 (singles) and 123 (multiples) test images that are classified via several paid experts into four empirical categories of increasing symmetry *complexity*: (P) perfect, (Q) quasi (or approximate) symmetric, (C) corrupted with clutter and (N) not globally symmetric;

[5]PSU 2011: http://vision.cse.psu.edu/research/symmComp/index.shtml, PSU 2013: http://vision.cse.psu.edu/research/symComp13/index.shtml

but locally symmetric. See Appendix E.3 for details. An additional 70 images are used as a separate training subset. In addition to the hand annotated groundtruth symmetry axes, every axis is associated with an elliptical region that specifies the *extent* of the symmetry region that supports the symmetry axis. This allows more precise future evaluations that takes into account the estimated symmetry region as well.

As baseline, we use the state-of-the-art method of Loy-Eklundh [184]. We use the Matlab implementation available from the authors' website[6] with optimally tuned parameters per dataset to detect reflection (bilateral) symmetries. We tune four parameters: $\{t_s, t_a, t_r, t_m\}$, that control the scale, angular and radial matching distances and the number of matches admitted per symmetric SIFT feature. An offline search procedure is used to obtain the best results per dataset. The code, which has been optimized via C++ calls, runs very fast: 1.12 ± 0.70s for a 320×240 input image. We detail the parameter search procedure and optimal parameters, together with detailed runtimes for each dataset in Appendix E.

We adopt the same evaluation procedure used in the 2013 Symmetry Competition [180], where we compare the accuracy of detections via standard Precision-Recall (PR) curves. In order to determine if a detected axis $a_i = \{(x_i, y_i), \theta_i\}$ is correct with respect to a given groundtruth $GT = \{(x_{gt}, y_{gt}), \theta_{gt}\}$, we use the following three criteria: 1) the minimum angular difference between θ_i and θ_{gt} is less than $t_1 = 10$ degrees, 2) the shortest Euclidean distance from (x_i, y_i) to the GT axis is less than $t_2 = 0.2 \times \min\{l_{a_i}, l_{GT}\}$, where $l_{(\cdot)}$ is the length of the axis and

[6] http://www.nada.kth.se/~gareth/homepage/local_site/code.htm

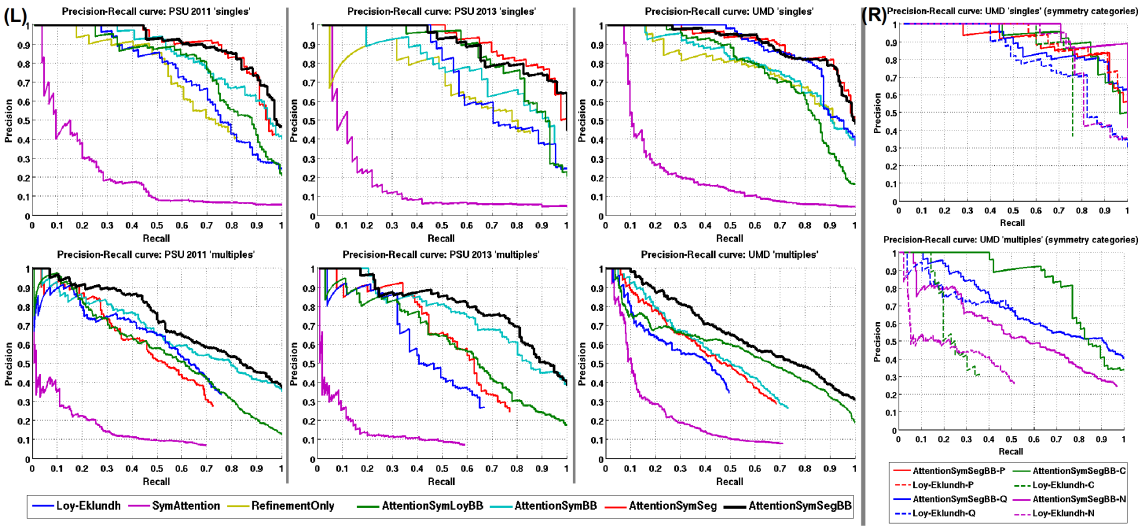


Figure 4.9: PR curves over the three datasets. (Left Panel) Columns (L-R): PSU 2011, PSU 2013, UMD Symmetry datasets. Rows: 'singles' (Top), 'multiples' (Bottom). (Right Panel) Results for different symmetry categories in the UMD Symmetry dataset. Average Precision (AP) scores are in Appendix E. See text for details.

3) the Euclidean distance between the centroids, (x_i, y_i) and (x_{gt}, y_{gt}) is less than $t_3 = 0.5 \times \min\{l_{a_i}, l_{GT}\}$. The first two criteria are from [180] while the third criterion was added to reject detections that are either too small (not the same scale) or not centered near the desired foreground/object[7].

4.6.2 Results

In this section, we report a series of detailed experimental evaluations of the proposed approach and compare its performance with the state of the art baseline Loy-Eklundh detector. In §4.6.2.1, we first evaluate the contribution of each

[7]Due to this additional criterion and the use of optimized baseline parameters, the PR curves reported here are not comparable to [180].

component of the approach and then compare variants of the approach with the baseline in §4.6.2.2. All the parameters (optimized with respect to the approach and dataset) are kept the same throughout all experiments and evaluated using the same evaluation procedure to produce comparable results.

4.6.2.1 Performance of individual stages

Before comparing the performance of the final approach with the baseline, an important question that needs to be answered is the contribution of each of the two stages of the approach: 1) the Symmetry Attention stage `[SymAttention]` and 2) the Symmetry Refinement stage `[RefinementOnly]`. We ran each of these stages individually[8] and compared their results when both stages are combined via fixation-based segments `[AttentionSymSeg]`. In addition, we investigated the effects of using only bounding boxes as simple segments within $\mathcal{R}_{sym}$ in the combined approach by not running the fixation-based segments at all in `[AttentionSymBB]`. The relevant PR curves evaluated over the three datasets, separated into 'singles' and 'multiples' categories are shown in Fig. 4.9 (left).

We highlight two important observations. First, we note that in all the datasets, the individual stages: `[SymAttention]` and `[RefinementOnly]` have consistently lower average precision (AP) compared to the case when both of them are combined in `[AttentionSymSeg]`. This is especially true for the symmetry at-

[8] As the symmetry refinement stage only returns the best symmetry axis, `[RefinementOnly]` was evaluated only on the 'singles' subset of each dataset. For other variants and 'multiples', the refinement stage is applied over segments obtained from the `[SymAttention]` stage.

tention stage which has the lowest AP over all datasets considered. This is not surprising since the attention map is meant to produce noisy putative symmetry axes. What the improvement in `[AttentionSymSeg]` shows is that our two-stage approach makes sense, where the refinement step helps remove a large number of false positives from the symmetry attention stage. Second, the contribution of using fixation-based segments in `[AttentionSymSeg]` compared to using only simple bounding boxes in `[AttentionSymBB]` is more pronounced in the 'singles' category. For 'multiples', we notice a consistently better performance for `[AttentionSymBB]` at larger recalls compared to `[AttentionSymSeg]`. A possible explanation for this behavior could be that images with multiple symmetries often occur in more complex/cluttered environments, causing erroneous regions to be segmented which reduces the accuracy of the returned final symmetry axes (Fig. 4.11 (a-f)). However, when one uses bounding boxes alone on 'singles' images, not enough symmetrical information is captured, due to the fact that most of the symmetries occur on objects that occupy a large part of the image. As both bounding boxes and fixation-based segments tend to provide complementary information, we therefore investigate an approach `[AttentionSymSegBB]` that *combines* a subset of the bounding boxes with the fixation-based segments in the next section.

4.6.2.2 Performance comparison with baseline

We compare quantitatively the performance of the full approach `[AttentionSymSegBB]` against the baseline detector of Loy and Eklundh `[Loy-Eklundh]` via PR curves eval-

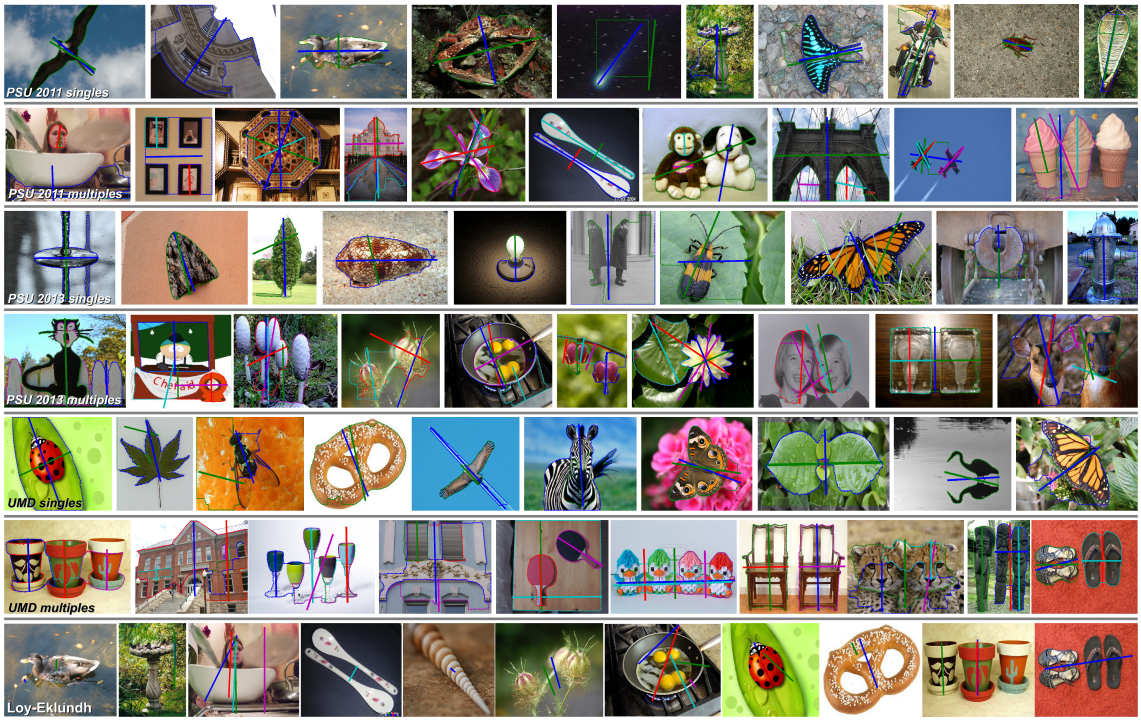


Figure 4.10: Example results (from fixation-based segments) and Loy-Eklundh [184] detections (last row). Ten images per dataset (rows). (From top row): PSU 2011 singles, PSU 2011 multiples, PSU 2013 singles, PSU 2013 multiples, UMD singles and UMD multiples. For 'singles', we show the top 2 detections while for 'multiples', the top 5 detections are shown: Symmetry axis (lines) and their support segments (dashes). Color encodes the relative ranking of the detections: blue, green, red, cyan and magenta (best – last).

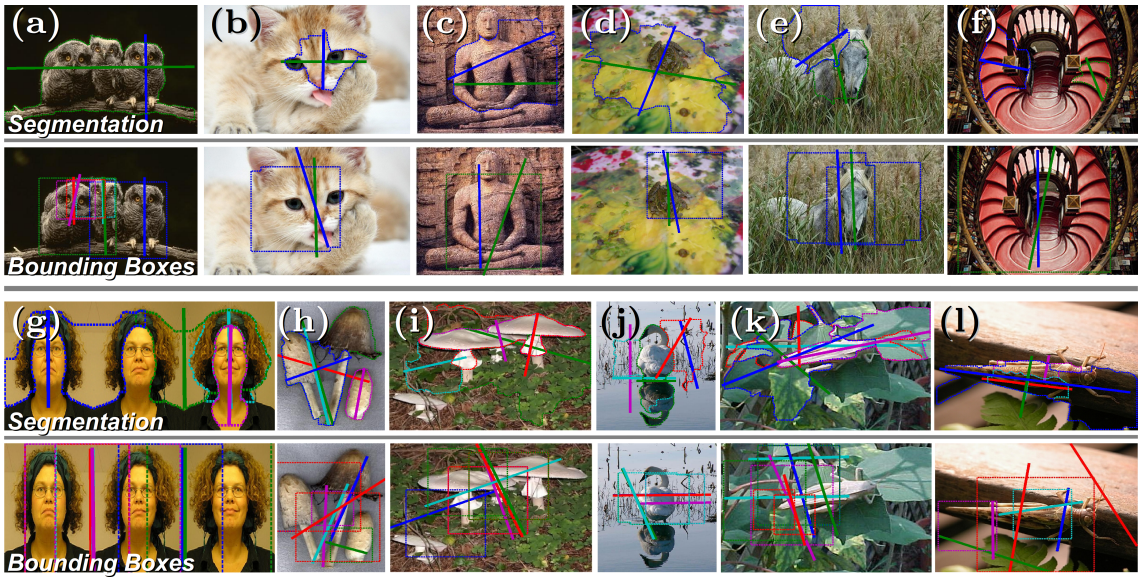


Figure 4.11: Complementary symmetry axes detected from fixation-based segments (top) and bounding boxes (below). In cases where segmentation is wrong or ambiguous (e.g. between objects, clutter), the predicted symmetries from bounding boxes are more accurate (a-f). In other cases, segmentation provides a more accurate region for prediction (g-l).

uated over the three datasets as shown in Fig. 4.9 (left). As a check on the contribution of the symmetry refinement step, we replaced the refinement step of the full approach with the method of Loy and Eklundh by running the detector only over the segments and bounding boxes in $\mathcal{R}_{sym}$, `[AttentionSymLoyBB]`.

From the results, we first note that the `[Loy-Eklundh]` baseline performs much better in the 'singles' categories compared to the 'multiples' categories, with the best AP for UMD singles (0.865) and the worst AP for UMD multiples (0.321). This indicates that matching keypoints in cluttered scenes typical of 'multiples' is more challenging compared to simpler scenes in 'singles'. Moving on to the comparisons, the full approach: `[AttentionSymSegBB]` has similar performance (precision) as the baseline at low recalls but quickly outperforms `[Loy-Eklundh]` at higher recalls: >0.5 ('singles'), >0.1 ('multiples'); and is among the top performing algorithm in terms of AP: ≥0.89 ('singles'), ≥0.66 ('multiples'). This consistent performance across both 'singles' and 'multiples' shows that by augmenting $\mathcal{R}_{sym}$ with bounding box segments, we capture enough complementary information for the refinement step to accurately localize the best symmetry axis (Fig. 4.11). Finally, as the PR curves of `[AttentionSymLoyBB]` are consistently lower in performance compared to the full approach, we can conclude that the symmetry refinement approach is able to localize the correct symmetry axis with much better precision within the segment, compared to using the local based symmetric features of Loy and Eklundh. The implications of these results are discussed in §4.6.3.

In addition, we compared the full approach against the baseline over different symmetry categories in the UMD Symmetry dataset (Fig. 4.9 (right)). We note

that [Loy-Eklundh] has the best AP in the 'P' (perfect symmetry) category for 'singles' (0.862) and in the 'Q' (quasi-symmetric) category for 'multiples' (0.439), while it has the worst APs in the 'C' (corrupted with clutter) category. By contrast, our approach achieves the best performance in the more challenging 'N' (not globally symmetric) and 'C' categories for 'singles' (0.967) and 'multiples' (0.820) respectively. This shows that our approach is able to better handle more complex cluttered situations compared to [Loy-Eklundh].

4.6.3 Discussion

We discuss two key insights provided by the experimental results presented in the preceding section and illustrate them with example outputs in Fig. 4.10.

4.6.3.1 Advantages of a two stage approach

An important hypothesis of this approach is the proposition that symmetry detection, should be performed in a two-stage manner that starts off with 1) an attentional-based mechanism to quickly determine potential symmetrical regions and 2) followed by a more expensive symmetry detection step applied at each region. The experimental results clearly demonstrate the advantage of this strategy, with the full approach [AttentionSymSegBB] significantly outperforming [Loy-Eklundh] in all the datasets at medium and high recalls. There are three main reasons.

First, we note that many of detections from [Loy-Eklundh] are relatively small and insignificant compared to the expected groundtruth (4.10 (last row)). This is

because the approach does not estimate the correct symmetry *scale*. Our approach, on the other hand, defines scale in terms of the segments or *objects*, which reduces the chance that insignificant symmetries are detected, thereby improving performance. Of course, as we have noted earlier, errors in segmentation will reduce the performance of the approach when relying on fixation-based segments alone (especially for 'multiples'). Integrating the segments with simple bounding box regions ameliorates the problem but using a more sophisticated segmentation mechanism that takes advantage of high-level (e.g. symmetry, object-hood) information may provide a better solution.

Second, combining symmetry attention with segmentation also improves the *precision* of the symmetry axes detected in the refinement stage. This is because by limiting the search of the symmetry axis to the approximate regions, irrelevant information (texture or edges) are removed, reducing false positives. This is shown clearly in the improved results of both the full approach and when Loy-Eklundh is used in the refinement stage `[AttentionSymLoyBB]` (except UMD singles).

Finally, using a two stage approach is more natural for detection of *multiple* symmetries. This is because at the first (attentional) stage, we have already detected putative symmetry locations in the image which we then independently verify in the second (refinement) stage. A single stage approach, however, often needs to apply a mechanism for *separating* the input data into various clusters or potential symmetries via ad-hoc similarity measures. In the Loy-Eklundh detector, the approach first checks for matching strengths and for scale consistency to detect different symmetry clusters, by using a pre-determined threshold. This approach

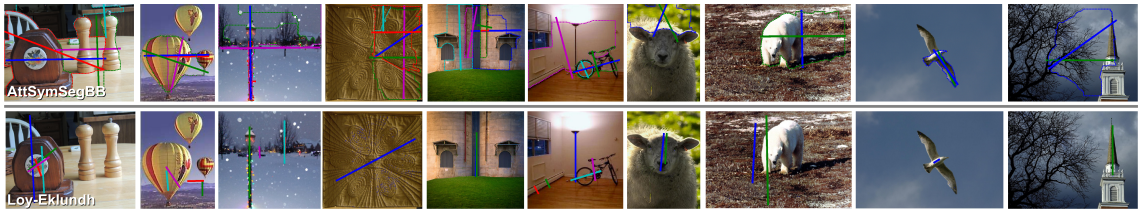


Figure 4.12: Example failure cases of the proposed approach (top row) compared to Loy-Eklundh (bottom row).

often removes many detections, especially for the 'multiples' dataset, resulting in reduced recall rates.

4.6.3.2 Local features versus statistics-based detection of symmetry

The experimental results also support the use of robust statistics to detect symmetry compared to using local image based features as in the baseline. This is demonstrated by comparing the PR curves of `[AttentionSymSegBB]` (our approach) and `[AttentionSymLoyBB]`. The latter is the same as the full approach except that the Loy-Eklundh detector was used instead of our statistical approach in the symmetry refinement step. Beyond a recall rate of 0.3 ('singles') and 0.05 ('multiples'), our proposed approach consistently outperforms Loy-Eklundh's detections even though the same segments from $\mathcal{R}_{sym}$ were used. The main reason for this difference in performance is that often many of the segments are devoid of texture that is required for good feature matching. Furthermore, although local SIFT based features are robust to occlusions, matching them across clutter in real situations often produces numerous mismatches and (as a consequence) incorrect detections. This also explains why `[Loy-Eklundh]` has the worst performance in the 'C' subset of the UMD

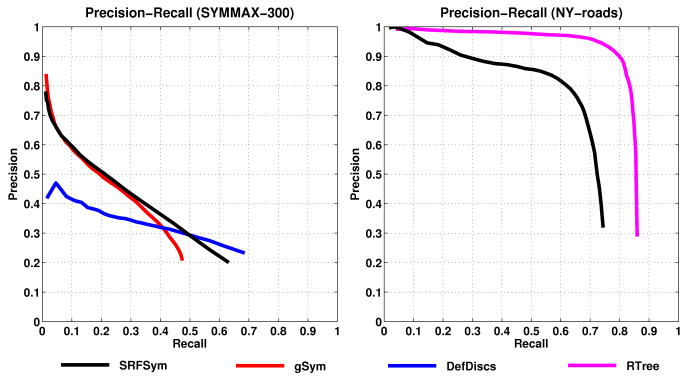
Symmetry dataset.

Although using edge-based statistics for detecting symmetries is clearly advantageous in such situations, there are certain limitations as well (Fig. 4.12). Since we only used edge-based features, errors in the edge detection/segmentation, or noise from background may result in errors in localizing the symmetry axis. Also, and more importantly, edge statistics alone may not provide sufficient discriminative information to support two competing symmetries, especially when the scale (sample size) is small. One possible solution that we will explore in future work is to incorporate more discriminative features, such as the symmetric SIFT features of the baseline, into the statistical comparison framework.

4.7 Experiments: Curved Symmetry Detection

4.7.1 Datasets, baselines and evaluation procedures

We use the SYMMAX-300 dataset (200 train/ 87 test) introduced by [278], that contains curved symmetry annotations from the BSDS-300 dataset [193]. Specifically, automatically generated medial axes are presented to human annotators who select which axes best supports the groundtruth segments. We follow the same evaluation procedure as [193], where instead of boundaries, the groundtruths are human annotated curved symmetries and we report the Precision-Recall (P-R) curves and the ODS, OIS and AP metrics of [7] using the same evaluation parameters suggested by [278] where symmetry pixels close enough to the groundtruth (<0.01% of the image diagonal) are considered correctly matched. As baselines, we compare



Method	SYMMAX-300	NY-roads
Our approach, SRFSym	**0.38**,**0.42**,**0.27**	0.70,0.70,0.63
gSym [278]	0.36,0.40,0.22	-
DefDiscs [159]	0.37,0.41,0.23	-
RTree [249]	-	**0.85**,**0.85**,**0.83**

Figure 4.13: Curved symmetry prediction accuracy. (Top) Precision-recall curves: SYMMAX-300 (left) and NY-roads (right). (Below) [ODS, OIS, AP] scores [7] in each cell. Best viewed in color.

our SRF-based approach (SRFSym) with two state-of-the-art curved symmetry detectors: 1) global-symmetry (gSym) of Tsogkas and Kokkinos [278] and 2) symmetric deformable-discs (DefDiscs) of Lee et al. [159]. Finally, the same evaluation parameters suggested by [278] are used to compare the symmetry prediction accuracies of all three approaches. We also compared our approach to the recent regression-tree based (RTree) centerline prediction method of Sironi et al. [249] over published results in their "Aerial" dataset of 14 satellite images (7 train/ 7 test) of road networks from New York state (NY-roads). Following [249], we consider symmetry pixels within 2 pixels of the groundtruth axes as correct to obtain the P-R, ODS, OIS and AP metrics.

4.7.2 Results and discussion

Fig. 4.13 summarizes the the evaluations performed as described in the previous section. We briefly discuss these results and their implications. We show

example results of curved symmetry detections with their corresponding segmentations in Fig. 4.17 in §4.9.

4.7.2.1 Curved symmetry accuracy over SYMMAX-300

Our method, `SRFSym`, returns the most accurate curved symmetry predictions compared to `gSym` and `DefDiscs` in all accuracy metrics [ODS,OIS,AP] (Fig. 4.13 (top-left)). Notably, we see that beyond a recall value of 0.2, `SRFSym` outperforms `gSym` consistently at higher recalls. This shows that `SRFSym`'s curved symmetry predictions are more accurate across a larger range of symmetry scores. This is likely due to: 1) our complementary set of features (`gSym` does not use edges) which works better at non-textured regions and 2) the structured predictions over multiple scales smooth out wrong predictions across multiple decision trees.

4.7.2.2 Curved symmetry accuracy over NY-roads

In this dataset, `SRFSym` is unable to match the (almost) perfect curved symmetry predictions of `RTree` (Fig. 4.13 (top-right)), even with reasonably high predictions (>0.8) for most recalls. The reason for the drop in precision at high recalls is that `SRFSym` responds to other symmetric regions (besides roads) that are not in the groundtruth. This shows that for this particular task and modality, the regression formulation proposed in [249] makes sense compared to our approach which uses more general features for detecting symmetry. Modifying our approach to take advantage of features derived from the sparse convolutional filters of `RTree` may

also improve our performance further. Finally, it is also important to note that although `SRFSym` is comparatively less precise, its inference is extremely fast compared to `RTree`: seconds compared to the minutes/hours reported in [249].

4.8 Experiments: Bilateral Symmetry-Constrained Segmentation

4.8.1 Datasets, baselines and evaluation procedure

We use three datasets for experimental evaluation of extracting symmetric regions exhibiting bilateral symmetry. The first dataset is the PSU 2013 dataset[9], introduced in §4.6.1, consisting of 20 training and 40 testing images augmented it with human labeled segmentation ground-truth of the symmetrical regions. The second and third datasets come from the work of Sun and Bhanu [264] UCSD symmetry segmentation dataset. These datasets consist of selected images from two publicly available datasets: a) Berkeley Segmentation Dataset (BSDS) [6] (15 images) and b) 93 images from Caltech-101 object categories [67]. Since both datasets only have segmentation ground-truths, ground-truth symmetry axes associated with the symmetrical regions were added in manually. In addition, we selected 10 images from BSDS and 50 images from Caltech-101 (from the same categories), which we used as a separate training set.

Note that we only evaluate with bilateral symmetries predicted from the initial symmetry attention step, `[SymAttention]`. There are two reasons for this. First, as the full approach `[AttentionSymSegBB]` consists of a the refinement step applied over

[9] Available online at http://vision.cse.psu.edu/research/symComp13/index.shtml

segmented regions or bounding boxes of initial putative symmetry fixation points, including a symmetric segmentation *after* the refinement step does not make sense. Second, as other approaches do not apply a "segmentation-in-the-middle" phase as well, a proper and fair evaluation of the contribution of the symmetry-constrained segmentation can only be derived meaningfully from `[SymAttention]`.

We compare with two baselines. The first baseline is the standard 4-way MRF with no symmetry constraints, `GC`. The second baseline is the region-merging approach of Sun and Bhanu [264] that was discussed in §4.2.2. As the segmentation results using Sun and Bhanu are not available, we re-use the results reported in Table 10 of their paper.

A note on the performance metrics used. We capture the accuracy of the segmentation via the segmentation covering score C_{seg} used in the BSDS dataset [6] that measures the overlap of the regions R' in the final segmentation S' with the groundtruth S containing regions R by:

$$C_{seg}(S' \rightarrow S) = \frac{1}{|\mathcal{P}|} \sum_{R \in S} |R| \cdot \max_{R' \in S'} \mathcal{O}(R, R') \tag{4.16}$$

where $\mathcal{O}(R, R') = \frac{|R \cap R'|}{|R \cup R'|}$ measures the overlap of the two regions R, R'. Additionally, we used the unsupervised and supervised segmentation performance metrics $EVA\_SEG_{unsup}, EVA\_SEG_{sup}$ used in in the UCSD dataset for comparison. EVA_SEG_{unsup} is a simple measure of region contrast between segments computed using image color and texture features. The larger the contrast, the better "separated" are the regions, which leads to a larger EVA_SEG_{unsup} score. EVA_SEG_{sup} measures the overlap between the test and groundtruth regions sim-

Dataset	Method	C_{seg}	EVA_SEG_{unsup}	EVA_SEG_{sup}
PSU 2013	Our Method, `SymSegGC`	**0.72** (+0.08)	**0.88** (+0.02)	**0.69** (+0.06)
	`GC`	0.64	0.86	0.63
Caltech-101	Our Method	**0.69** (+0.07)	**0.85** (+0.02)	**0.67** (+0.04)
	`GC`	0.62	0.82	0.63
	Region-Merging [264]	–	0.83	–
BSDS	`SymSegGC`	**0.78** (+0.09)	**0.88** (+0.02)	**0.79** (+0.03)
	`GC`	0.69	0.86	0.66
	Region-Merging	–	–	0.76

Table 4.2: Performance comparison of mean segmentation accuracy between different approaches over the three datasets. Dashes (–) indicate missing results which were not reported in [264] or for C_{seg} not performed since final segmentation results were not made available. Improvements ($+x$) are with respect to the next closest result.

ilar to $\mathcal{O}(R, R')$ with additional penalties for over and under-segmentation.

4.8.2 Results and discussion

The first set of experiments evaluates the contribution of our symmetry-constrained segmentation approach, `SymSegGC`, in enforcing symmetry within the final segmentation. For this purpose, we used the human annotated symmetry axes as input and compare with `GC` (no symmetry) and the state-of-the-art symmetry embedded region growing approach of Sun and Bhanu [264]. Table 4.2 summarizes and compares the performance of various approaches over the three datasets.

From Table. 4.2, several key results are worth highlighting. Firstly, our ap-

Dataset	SymSegGC +	C_{seg}	EVA_SEG_{unsup}	EVA_SEG_{sup}
PSU 2013	[SymAttention]	**0.67** (+0.03)	**0.87** (+0.01)	**0.65** (+0.02)
	Loy-Eklundh [184]	0.63	0.86	0.62
Caltech-101	[SymAttention]	**0.68** (+0.02)	0.84	0.62
	Loy-Eklundh	0.63	0.84	0.62
BSDS	[SymAttention]	**0.68** (+0.03)	**0.87** (+0.01)	**0.55** (+0.01)
	Loy-Eklundh	0.64	0.87	0.51

Table 4.3: Performance comparison of mean segmentation accuracy with two different methods of automatic symmetry axis detection. Improvements ($+x$) are with respect to the next closest result.

proach achieves the overall best performance in terms of segmentation accuracy compared to the other two approaches over all of the performance metrics used. The most significant improvement occurs over the fixation-based baseline, highlighting the contribution of the symmetry prior in improving the final segmentation accuracy. Secondly, compared with the symmetry integrated region-merging approach, our approach performs significantly better using the C_{seg} metric and less significantly so using the EVA_SEG_{sup} and has almost the same performance for EVA_SEG_{unsup}. This is not surprising, since both C_{seg} and EVA_SEG_{sup} measure the overall segmentation accuracy with respect to the groundtruth symmetry target(s), while EVA_SEG_{unsup} evaluates segmentations based on simple color and texture contrast. Fig. 4.14 highlights the improvements shown via example final segmentations of the fixation-based approach compared with the proposed approach.

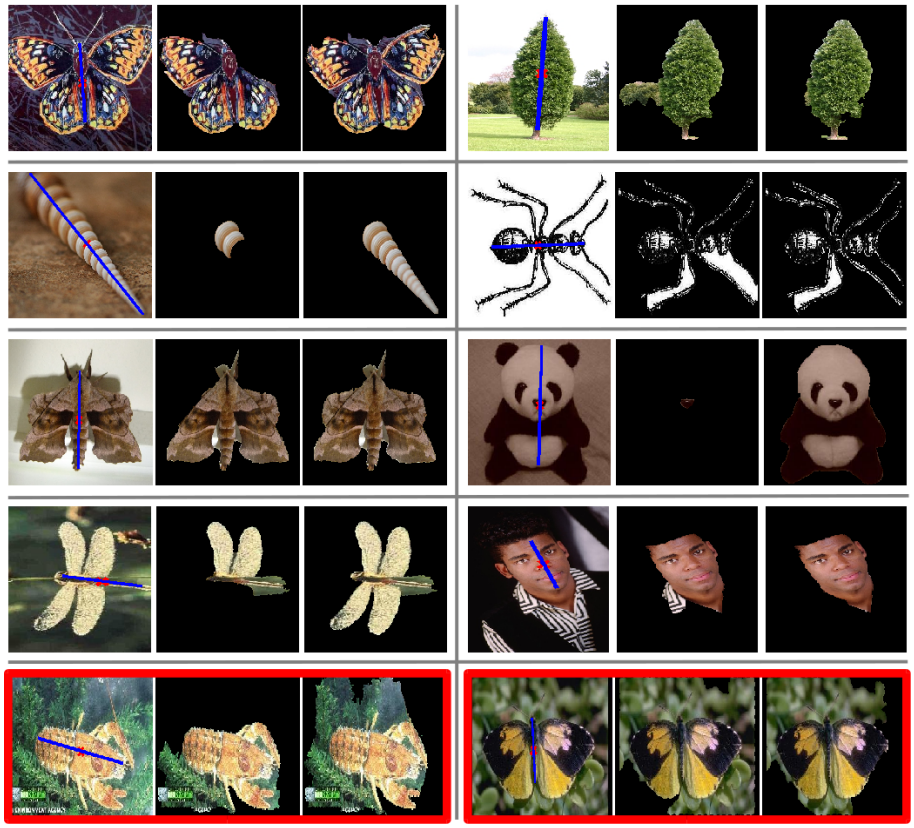


Figure 4.14: Example final segmentation results, two results per row: (L-R) Input image + Symmetry axis, `GC`, Our method `SymSegGC`. The last row with red boxes indicates typical failure cases: (L) Weak symmetry causes segmentation to leak into background, (R) Noisy background confuses the symmetry strength measure, with only partial improvements.

In the second set of experiments, we evaluate the complete approach by integrating the results of putative symmetry axes detections derived from the symmetry attention map D_{sym} as described in §4.3.1.1. We compared the final segmentation results using detections of Loy-Eklundh [184]. In both cases, we used the symmetry axis with the highest response (strongest bilateral symmetry). Table. 4.3 compares the performance evaluation of the two approaches over the three datasets used.

As expected, the performance of the complete approach takes a hit when noisy putative symmetry axes are used. However, we see that in most cases, the performance of the complete approach does not differ too much from `GC` of Table 4.2. This is due to the modulation of the symmetry prior term (eq. (4.14)) for situations where the bilateral symmetry is weak or completely wrong. Another interesting observation is that integrating the symmetry attention axes tends to give slightly better final segmentation accuracies compared to using the detections of Loy-Eklundh. A possible explanation could be that the images used in the three datasets contain symmetrical regions without texture (especially PSU 2013 and Caltech-101) which may result in numerous wrong (but strong) symmetry axes to be detected using Loy-Eklundh. Fig. 4.15 shows some qualitative results when we integrate the detected symmetry axes using the two detection methods.

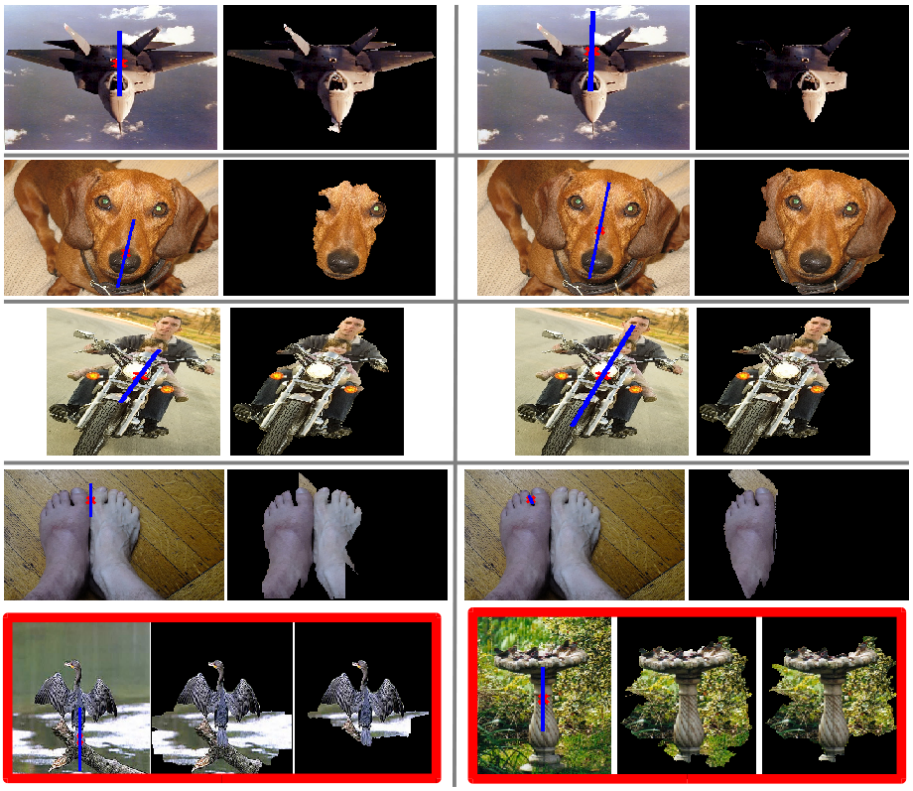
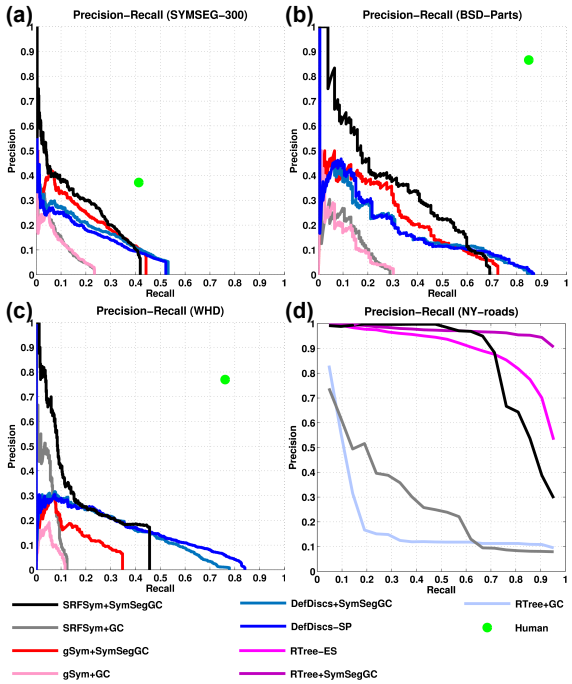


Figure 4.15: (Top) Example bilateral symmetry-constrained segmentations with automatically detected symmetry axes, two results per row showing `[SymAttention]` (left column) and Loy-Eklundh detections (right column). Last row with red boxes indicates typical failure cases where segmentation leakage still occurs.

4.9 Experiments: Curved Symmetry-Constrained Segmentation

4.9.1 Datasets, baselines and evaluation procedure

We use three datasets (train/test splits) for our main evaluation: 1) SYMSEG-300 (200/87), 2) BSD-Parts (0/36) and 3) Weizmann Horses, WHD (20/61) (both from [159]). SYMSEG-300 is an extension of SYMMAX-300 where we extract symmetric segments based on the original BSDS-300 groundtruth segments. BSD-Parts and WHD were introduced by [159] as one dataset for evaluating the `DefDiscs` superpixel grouping approach. For comparisons, we applied our symmetry-constrained segmentation approach (`SymSegGC`) using symmetry axes predictions from: 1) `SRFSym` (our approach), 2) `gSym` and 3) `DefDiscs`. As an additional demonstration of the contribution of the symmetry priors, $\{S_{pp'}, B_{pq}\}$, we evaluated `SRFSym` and `gSym` *without* these two priors, effectively reducing the segmentation to the standard MRF-based approach (`GC`) that was used in §4.8. We also compared the grouped superpixels segments obtained from `DefDiscs` (`DefDiscs-SP`) as an additional baseline. Following [159], we consider a segment as correct when its standard Intersect-over-Union (IoU) score with respect to the groundtruth exceeds 0.4 over all three datasets and report the resulting P-R curves and Average Precision (AP) metrics for each method. We also compared `SymSegGC` with the estimated centerline scales predicted by `RTree` (`RTree-ES`) over the NY-roads dataset where we used symmetry axes predictions from `SRFSym` with/without symmetry priors and similarly for `RTree`. The same evaluation procedure of [249] that applies an exclusion zone of 0.4% of the groundtruth



Methods		Datasets			
Sym Detector	Segmentation	SYMSEG-300	BSD-Parts	WHD	NY-roads
SRFSym	SymSegGC	**0.13**	**0.27**	**0.15**	0.86
	GC	0.04	0.05	0.04	0.28
gSym [278]	SymSegGC	0.11	0.18	0.06	-
	GC	0.03	0.05	0.01	-
DefDiscs [159]	-SP [159]	0.09	0.17	0.14	-
	SymSegGC	0.10	0.16	0.13	-
RTree [249]	-ES [249]	-	-	-	0.90
	SymSegGC	-	-	-	**0.97**
	GC	-	-	-	0.19

Figure 4.16: Curved symmetry-constrained segmentation accuracy. (Left) Precision-recall curves: (a) SYMSEG-300, (b) BSD-Parts, (c) WHD and (d) NY-roads. (Right) Corresponding Average precision (AP) scores per cell.

radius was used to generate comparable results.

4.9.2 Results and discussion

Fig. 4.16 summarizes the the performance evaluation of curved symmetry-constrained segmentation as described above. Some example results of curved symmetry detection and segmentation of corresponding symmetric regions using our approach compared to DefDiscs-SP [159] are shown in Fig. 2.8 and we briefly discuss these results and their implications.

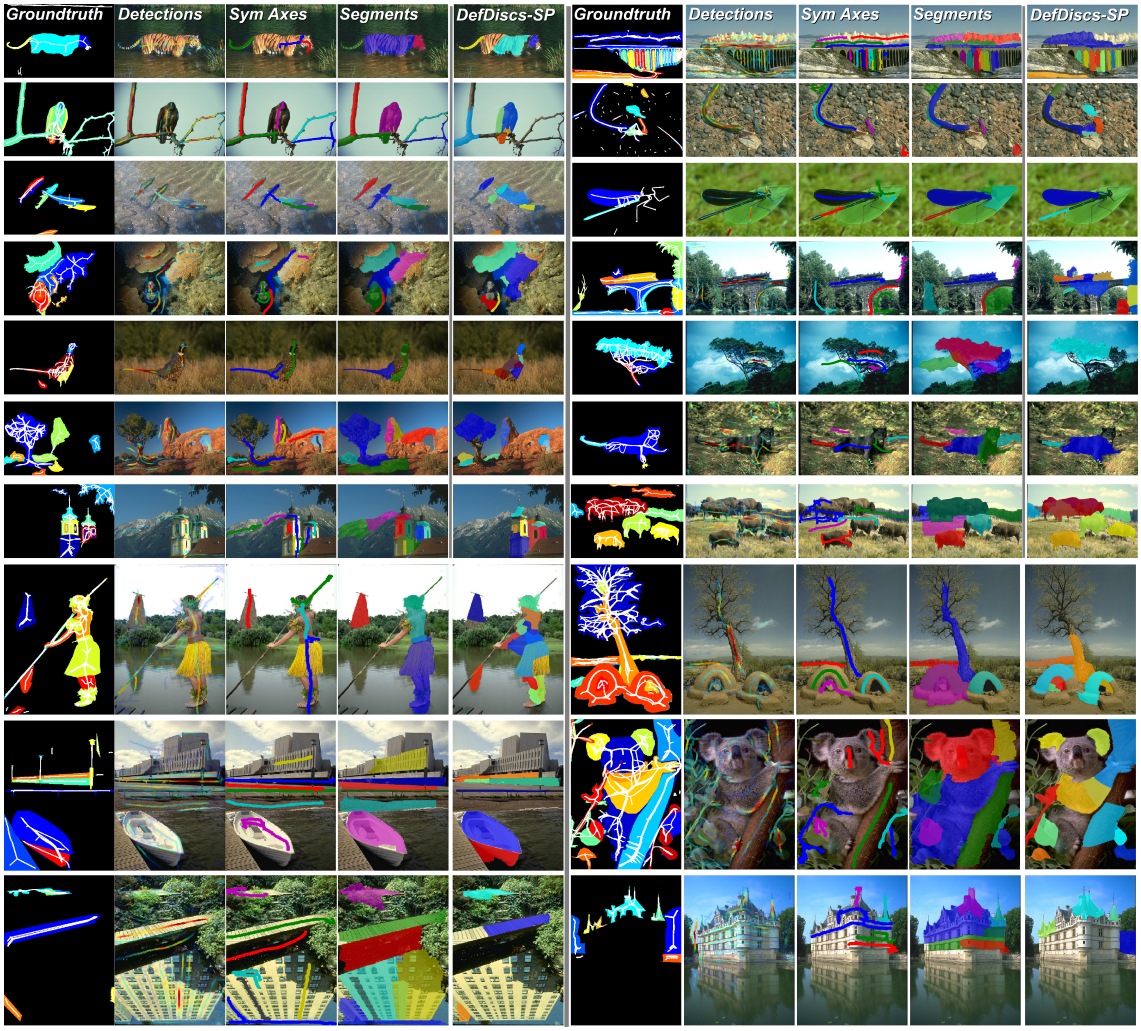


Figure 4.17: Example curved symmetry detection and symmetrical segmentation results, 10 results per panel: (L-R): Groundtruth of symmetrical axes (white) and regions, curved symmetry detections using `SRFSym`, extracted symmetry axes, curved symmetry-constrained segments using `SymSegGC`, segmentation using `DefDiscs-SP` [159].

4.9.2.1 Symmetric segmentation accuracy over SYMSEG-300, BSD-Parts and WHD

The general observation is that our full approach (`SRFSym+SymSegGC`) reports the best overall AP compared to other approaches as shown in Fig. 4.16 (a-c). Removing the symmetry prior in all approaches decreases accuracy by a significant amount, highlighting its importance. The dataset that challenges `SRFSym+SymSegGC` the most is WHD, where the textureless and small regions of horses (e.g. legs, tails) are better captured by the superpixels computed in `DefDiscs-SP`. Nonetheless, our full approach is able to extract symmetric parts with better accuracy from WHD until 0.46 recall. An interesting observation is when we pair up symmetry axes predicted by `DefDiscs` with `SymSegGC`, the performance is at least on-par (or slightly better in SYMSEG-300) with `DefDiscs-SP`. This shows that our proposed symmetry-constrained segmentation approach not only makes sense but is flexible enough to work with other approaches. It also highlight the *complementary* nature of both approaches: while [159] is a local (and slower) approach that groups superpixels, our proposed approach presents a faster alternative that captures longer range branched symmetries.

4.9.2.2 Symmetric segmentation accuracy over NY-roads

Although our full approach (`SRFSym+SymSegGC`) does not outperform `RTree-ES` in terms of overall AP, our precision is still higher than `RTree-ES` up to a reasonably

high recall of 0.7 (Fig. 4.16 (d)). The rapid drop in precision after this recall is once again due to `SRFSym` responding to other symmetric regions in the image. Another interesting observation is the improved performance over `RTree` when we pair the centerlines of `RTree` with `SymSegGC`. This highlights the key advantage of enforcing a global symmetrical consistency which greatly improves the accuracy of the final segmentation.

4.10 Conclusions

We have presented a complete approach for detecting and segmenting symmetric structures from real images. Robust approaches for detecting bilateral and curved reflection symmetries were proposed and evaluated over different datasets, with state-of-the-art results compared to other approaches. A novel two-stage approach that uses putative symmetry attention points followed by a symmetry refinement step was proposed to accurately detect and localize bilateral symmetries. For curved reflection symmetries, we developed a fast SRF-based symmetry detector trained on multiscale patch-based symmetry features sensitive to local symmetry. For segmenting symmetric regions, symmetry constraints are embedded within a novel 5-way MRF via an additional pairwise cross-symmetry term that is appropriately modulated by the predicted symmetry scores. This allows the approach to produce accurate segmentations of approximate symmetrical structures that are common in real images. Results of experimental evaluations confirm that our segmentation approach is not only more accurate than existing state-of-the-art, but is

also flexible enough to improve existing segmentation approaches when paired with their symmetry detections.

We have shown here a practical implementation of detecting a specific Gestalt principle (symmetry) and applied it for a specific higher-level visual task (segmentation). This is an important step for solving the FGO problem (§1.1) as we are now able to extract symmetric segments, which are typically salient foreground objects, for further processing. In the next chapter, we demonstrate the detection of *functionalities* or affordances, a universal concept similar to symmetry via geometrical features, and apply it for the task of detecting different parts of common household tools.

Chapter 5: Object-Level Functional Category Detection

An important aspect of Gestalt or Mid-Level Vision is the *generalizability* of the proposed approaches. Unlike approaches that require large amounts of training data or separate training regimes for different situations (e.g. [15, 122, 307]), mid-level vision advocates the use of well-known invariants captured by Gestalt (e.g. symmetry, closure, proximity) that generalize well to numerous conditions and environments. Developing invariant representations is also a key aspect of vision according to Marr (1976) [192] and Gibson (1979) [83]. As noted by Richards [232], although Marr, Gibson and Gestalt offer different and sometimes conflicting views of visual perception, the study of the object's "value to the observer" [135] or its *affordance* [83] remains an open research problem that ties these three differing schools of thought together. Motivated from these views, we tackle the preeminent issue of *object recognition* in Computer Vision from a *functional* or affordance perspective in this chapter. Specifically, we detect affordances of *tool parts* by training a SRF-based classifier to associate *geometric* features with seven affordance categories: `{grasp,cut,scoop,contain,pound,support,wrap-grasp}` so as to produce *pixel accurate* predictions of the affordances given a RGB-Depth image in *real-time*. Extensive comparisons with other (slower) approaches using more complex features

show that our SRF-based method is able to provide highly accurate predictions within a fraction of the time needed[1].

5.1 Introduction

The ability to understand and perceive objects and tools beyond their simple labels (e.g. names) is a vital requirement for Computer Vision to function "in-the-wild" [112, 179, 287, 293, 309]. This capability enables generalization so that novel objects with similar shared attributes can be recognized and used, and is key to scaling up Computer Vision approaches [48]. The goal of this work is to establish a technique of generalizing object recognition based on their intended functionalities or affordances. Such a capability will enable Computer Vision approaches and mobile agents to: 1) recognize a larger variety of objects based on their functions, 2) suggest meaningful alternatives and 3) know the correct (and safe) method of manipulating such tools while working with humans.

The input is a RGB-Depth (RGB-D) image, and the output is a probabilistic "functional" map that shows pixel accuracy localizations of potential target functional regions (Fig. 5.1). Key to our approach is the use of view-invariant local geometric features: 1) Depth gradients, 2) Surface normals, 3) Principal curvatures [49], 4) Shape-index and Curvedness measures [134]. Unlike other approaches (§5.2) that use full 3D (metric measures) or require detailed mesh reconstruction, we show that it is possible to relate local geometry and shape primitives (§5.3.1)

[1] Joint work with Austin Myers and was published as [205]. Full results, code and videos are available online http://www.umiacs.umd.edu/~amyers/part_affordance/

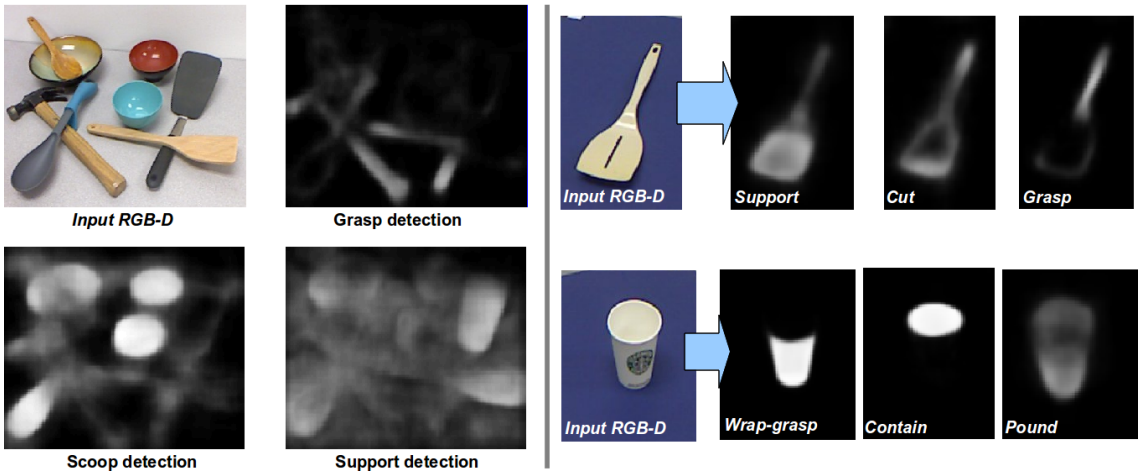


Figure 5.1: Predicting novel affordances in clutter (left) and in single objects (right). (Left) Detections of `grasp`, `scoop` and `support` in a cluttered scene. (Right) Novel affordances predicted for turner (spatula): `support,cut,grasp` (top) and mug: `wrap-grasp,contain,pound` (bottom). Notice that we are able to predict and localize reasonable locations for novel affordances, even in clutter and not just on well-defined object parts, but on the relevant regions of the object (e.g. the bottom of the mug affords pounding and the edge of the turner affords cutting). Brighter regions indicate higher probability.

to functionality as long as we impose that the features are view-invariant to some degree. This is achieved via a Structured Random Forests (SRF) [140] classifier trained for affordance prediction which we detail in §5.3.2. Experiments conducted over a new large RGB-D dataset, containing precise functionality annotations derived from multiple human annotators, showed that our SRF approach is able to achieve reasonable functionality detection in challenging test sequences containing novel objects with clutter and occlusions from different viewpoints (§5.4). In addition, we compare our approach with: 1) Superpixel Hierarchical Matching Pursuit (S-HMP) introduced by [204] and 2) a recent deep learning method termed the Sparse Autoencoder (SAE) that learns graspable features in common objects [163] and show in §5.4.4 that our approach is able to achieve comparable performance using simpler geometric view-invariant features.

5.2 Related Works

The study of affordance has a rich history in the computer vision and robotics communities. Early work sought a function-based approach to object recognition for 3D CAD models of objects like chairs [258]. More recently, many papers have focused on predicting grasping points for objects from 2D images [22,243,259]. [163] exploits a deep learning framework to learn graspable features from RGB-D images of complex objects and [126] detects tips of tools being held by a robot. From the computer vision community, [132] classify human hand actions in context of the objects being used, Grabner et al. [88] detect surfaces for sitting from 3D data.

Affordances might be considered a subset of object attributes, which have been shown to be powerful for object recognition tasks as well as transferring knowledge to new categories. Ferrari and Zisserman [73] learn color and 2D shape patterns to recognize the attributes in novel images. Parikh and Grauman [221] show that relative attributes can be used to rank images relative to one another, and Lampert et al. [152] and Yu et al. [304] show that attributes can be used to transfer knowledge to novel object categories. Using RGB-D data, [265] identify color, shape, material, and name attributes of objects selected via bounding boxes. [105] explored, using active manipulation of different objects, the influence of the shape, material and weight in predicting good pushable locations. [2] used a full 3D mesh model to learn so-called 0-ordered affordances that depend on object poses and relative geometry. Koppula et al. [143] view affordance of objects as a function of interactions, and jointly model both object interactions and activities via a Markov Random Field using 3D geometric relations ('on top', 'below' etc.) between the tracked human and object as features.

Recently, unsupervised feature learning approaches have been applied to problems with 3D information. [21] propose using hierarchical matching pursuit (HMP), and [251] propose using a convolutional recursive neural network to recognize objects from RGB-D images. For supervised methods, state-of-the-art performance using structured random forests [140] applied over RGB-D data for simultaneous object segmentation and recognition has been reported in [98].

5.3 Approach

In this section, we detail how we use train a SRF for affordance detection. Similar to border ownership prediction (§2.3), our SRF-based affordance detector is trained over small local feature patches that capture some form of affordance cues. In this work, we leverage on local measures of *geometry* and *shape* derived from RGB-D data and associate them with different affordance categories. In contrast to previous works that require accurate metric models [2] or predict attributes for segmented objects [265], we show that such local geometric and shape primitives are sufficient for pixel accurate functionality detection compared to those discovered via deep learning (which returns only a bounding box) [163], resulting in a more efficient and simple implementation that runs in *real-time* due to the fast inference inherent in SRF. We first introduce these geometric and shape primitives that we use as patch-based features for training the SRF.

5.3.1 Robust geometric and shape features

The key hypothesis of this work is that shape and geometry are physically grounded qualities which are deeply tied to the affordances of a tool part. When characterizing geometric qualities of a part, it is important that the features we compute are robust to variations, such as changes in viewpoint. At the same time, we would like to gain insight into the influence of basic geometric measures. Therefore, we leverage simple geometric features, such as surface normals and curvature, to learn the relationship between geometry and part affordance. In order to detect

affordances for a variety of tools in cluttered scenes with occlusions, we derive the following local geometric features from small $N \times N$ ($N = 16$) RGB-D input patches:

5.3.1.1 Depth features

We first apply smoothing and interpolation operators to reduce noise and missing depth values. Then, we remove the mean from the patch to gain robustness to absolute changes in depth. As features, we compute histograms over depth gradients (HoG-Depth). Similar to the 2D Histogram of Gradients (HoG) image descriptor [47], we compute gradients on the depth image and quantize them into four orientations to create a compact histogram feature.

5.3.1.2 Surface normals (SNorm)

We use the depth camera's intrinsic parameters to recover the 3D point cloud, from which we can estimate 3D surface normals. As with the depth, we remove the patch mean during feature learning, to make the representation more robust to changes in viewpoint.

5.3.1.3 Principle curvatures (PCurv)

The principle curvatures [49] are an extrinsic invariant of the local patch geometry, and are independent of viewpoint. The principal curvatures $(\kappa_1, \kappa_2), \kappa_1 \geqslant \kappa_2$ characterize how the surface bends in different directions.

5.3.1.4 Shape-index and curvedness (SI+CV)

The shape index (SI) and curvedness (CV) measures were introduced by Koenderink et al. [134] to characterize human perception of shape. These measures, which are derived from (κ_1, κ_2), are also viewpoint invariant and are defined as

$$SI = -\frac{2}{\pi}\arctan\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right), CV = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \tag{5.1}$$

SI and CV are continuous in the range $[-1, +1]$, where the shape index captures the type of local shape (elliptic, parabolic, etc.) and the curvedness its perceived strength.

5.3.2 SRF for affordance prediction

Different from previous approaches using SRFs for border ownership (§2.3) and curved symmetry detection (§4.4) introduced in the thesis, we impose here a novel structure that relates affordances to the local *patch* geometry and shape. This contrasts with the previous SRFs that predicts *pixel-wide* ownership or symmetries, resulting in a pixel-accurate prediction over *regions* given the test RGB-D image. To this end, we train a SRF that takes as input $\mathcal{X}$, features from local $N \times N$ patches described in §5.3.1 with pixel accurate annotations of the target affordance, $\mathcal{Y}$ (Fig. 5.2 (B)). The annotations impose the expected spatial structure of how the affordance should appear in the final prediction, which in this case are binary segments (c.f. annotations used for ownership in Fig. 2.5 and curved symmetry in Fig. 4.5 which are just a binary contour representation). For the j^{th} split (internal)

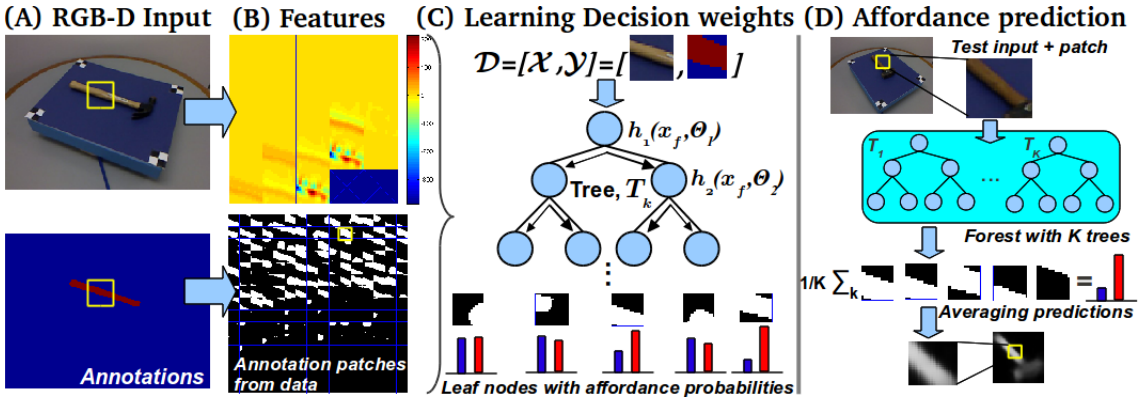


Figure 5.2: Affordance detection using SRF. (A) Input image with example patch highlighted. (B) Features extracted from each patch (top) and sampled annotation patches from data (below). (C) Training different patches, $\mathcal{X}$ with corresponding binary affordance annotations, $\mathcal{Y}$, learns the optimal θ_j at each split node. The leaf nodes store per pixel confidence scores for each $\mathcal{Y}$ encountered. (D) During inference, a test patch is assigned to a leaf node that contains affordance prediction. Averaging the predictions over the K trees produces an affordance confidence score per pixel.

node, we train a binary decision function $h(x, \theta_j) \in \{0, 1\}$ over random subsets, $x \in \mathcal{X}$, of the input features so that the parameters $\theta_j = (f, \rho)$ send $x(f)$ (where f is the feature dimension for each feature described in §5.3.1) to the left child when $h(\cdot) = 1$ if $[x(f) < \rho]$ and to the right child otherwise. Similar to other SRF-based approaches, the decision threshold, ρ, is obtained by maximizing a standard information gain criterion M_j over $\mathcal{D}_j \subset \mathcal{X} \times \mathcal{Y}$, the features and annotations using eq. (2.4) computed via an intermediate mapping $\Pi : \mathcal{Y} \mapsto \mathcal{L}$ of structured affordance labels into discrete labels $l \in \mathcal{L}$ following [51]. To determine Π, we first cluster via k-means random annotation patches that have the same affordance labels and select the largest $|\mathcal{L}|$ cluster centers. We repeat the training procedure until a maximum tree depth, d_t, is reached and we store at the leaf nodes per pixel confidence scores for each affordance annotation patch encountered during training. (Fig. 5.2 (C)). Each tree in the SRF therefore learns jointly, the 2D spatial structure *together* with the 2.5D features that describe the affordance within a patch. Inference using the trained SRF is extremely simple and fast. Given a forest of K trees and a testing patch with extracted features, the learned decision thresholds in each split node will send the patch to a leaf node that contains the predicted affordance labeling and confidence scores. We then average all K predictions for the final prediction (Fig. 5.2 (D)).

In our implementation, we train a SRF with $K = 8$ trees with a maximum training depth of $d_t = 64$. We use patches of size $N = 16$ and we set $|\mathcal{L}| = 10$ cluster centers for Π. Training over the entire affordance RGB-D dataset (§5.4.1) in parallel with an average of 5000 RGB-D images per split takes around 20 minutes on a 16

core Xeon 2.9GHz machine with 128GB of ram. Inference for a single RGB-D image of size (640×480) (height, width), takes an average of 0.1s which includes the time for feature extraction.

5.4 Experiments

5.4.1 Datasets

We use the RGB-D Part Affordance Dataset, introduced in [204], which focuses on everyday tools and the affordances of their parts. Each part or surface of the tool are hand annotated with multiply *ranked* affordances, ordered by their most likely (primary) functionality to their least likely functionality, for e.g. the inner surface of a bowl is labeled with `contain` as its primary affordance, followed by `scoop` and so on. Seven affordance categories are considered: `{grasp,cut,scoop,contain,pound,support,wrap-grasp}`. The dataset contains 107 diverse kitchen objects and tools with different appearances were captured over a turnstile (to get multiple poses) using a Kinect RGB-D camera. This results in 30,000 RGB-D images. Of these, more than 10,000 images have pixel-level ground truth affordance labels. In addition, we supplement the dataset with three sequences of around 1000 RGB-D frames, each collected by a mobile robot observing novel tools in clutter under changing viewpoints.

We also evaluate our approach to a more common, but related robotic task of determining where to grasp (a specific affordance). For this purpose, we used the recently introduced Cornell Grasping Dataset of Lenz et al. [163] to compare

against their deep-learning method and validate the effectiveness of our approach. The dataset contains 1035 RGB-D images of 280 graspable objects, where objects are captured from a small discrete number of viewpoints. Each image contains a single object, and is annotated with a set of rectangles indicating good or bad graspable locations.

5.4.2 Baselines

We compare our SRF-based affordance prediction method (SRF) with two state-of-the-art baselines: 1) Superpixels HMP (S-HMP) [204] and 2) a deep learning technique for detecting graspable regions from [163] termed the Sparse Autoencoder (SAE). S-HMP combines superpixels derived from the SLIC algorithm [1] with hierarchical sparse codes generated using Hierarchical Matching Pursuit (HMP) [21]. The sparse features generated by S-HMP are then passed into a SVM classifier that predicts the affordance category per pixel. SAE combines a cascade of two deep networks for detecting suitable grasping rectangles (location and orientations) given the input RGB-D image of an object. The first network is small runs fast to provide initial rectangles which are then passed for further evaluation using a deeper network with more complex features. To combine the features together, the authors explored several strategies and found that a two-stage structured regularization over features learned between the two networks yield the best performance, which is the variant that we compare with here.

5.4.3 Evaluation procedures

We use two evaluation metrics to provide different perspectives on the performance of our approach over the RGB-D Part Affordance dataset. The proposed approach output a probability map over the image for each affordance, which can be evaluated against ground truth labels to fairly compare their performance. First, we use the *Weighted F-Measure*, F_β^w, introduced recently by Margolin et al. [191] to evaluate saliency maps with continuous valued responses against binary valued ground-truths. F_β^w is an extension of the well-known F-measure F_β[2]:

$$F_\beta^w = (1+\beta^2)\frac{Pr^w.Rc^w}{\beta^2.Pr^w + Rc^w}, \text{with } \beta = 1 \tag{5.2}$$

where Pr^w and Rc^w are *weighted* versions of the standard precision $Pr = \frac{TP}{TP+FP}$and recall $Rc = \frac{TP}{TP+FN}$ measures. Here, TP, TN, FP, FN refer to true positives, true negatives, false positives and false negatives respectively. The key insight from [191] is to extend the standard precision and recall measures with weights derived by comparing the binary ground-truth and the continuous valued responses in order to reduce biases inherent in the standard measures. To do this, the authors proposed weights that measure the dependency of foreground pixels (pixels clustered together near the ground-truth are weighted higher), and assign lower weights to pixels far from the ground-truth.

Since the ground-truth in the RGB-D Affordance dataset provides rankings

[2]The F-measure with $\beta = 1$ is defined by the harmonic mean of the precision and recall values: $F_\beta = (1+\beta^2).\frac{Pr.Rc}{\beta^2.Pr+Rc}$ and is used as a measure of the accuracy of the Pr and Rc scores. β is a positive weight that gives preferences to either Rc ($\beta > 1$) or Pr ($\beta < 1$).

across multiple affordances, for a second measure we define a *rank* weighted F_β^w,

$$R_\beta^w = \sum_r w_r . F_\beta^w(r), \text{with} \sum_r w_r = 1 \tag{5.3}$$

that sums weighted $F_\beta^w(r)$ over their corresponding r ranked affordances. The ranked weights w_r are chosen so that the top ranked affordance is given the most weight, followed by the secondary affordance and so on. This allows us to capture if the detector is generalizing across multiple affordances appropriately. Note that when we impose $w_1 = 1$, (5.3) reduces to (5.2), where we consider only the top ranked affordance.

For the Cornell dataset, we follow the same evaluation procedure described in [163], where we averaged results from 5 random splits, and report both recognition accuracy, r_a and detection accuracy, d_a. For detection, we report the point-wise metric following [163] and [243], which considers the detection a success if it is within some distance from at least one ground-truth rectangle center. To obtain structured labels from this dataset for training the SRF, we estimated the ground-truth annotations of graspable regions by first applying a mask obtained over all graspable rectangles followed by a edge detection and hole filling operation (Fig. 5.3).

For fairness in comparison, we use the same training and evaluation parameters with the same input features in all experiments.

5.4.4 Results and discussion

We report results that demonstrate the performance of our approach using the proposed metrics described above: (F_β^w, R_β^w) for affordance detectors trained using

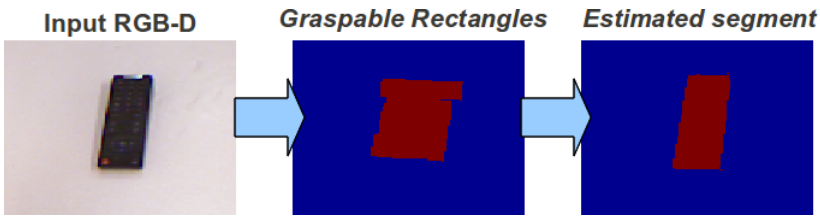


Figure 5.3: Estimating pixel accurate annotations from the Cornell Grasping Dataset. (Left) Input RGB image. (Middle) Overlay of several graspable rectangles. (Right) Edge detection and hole filling produces a pixel accurate segment.

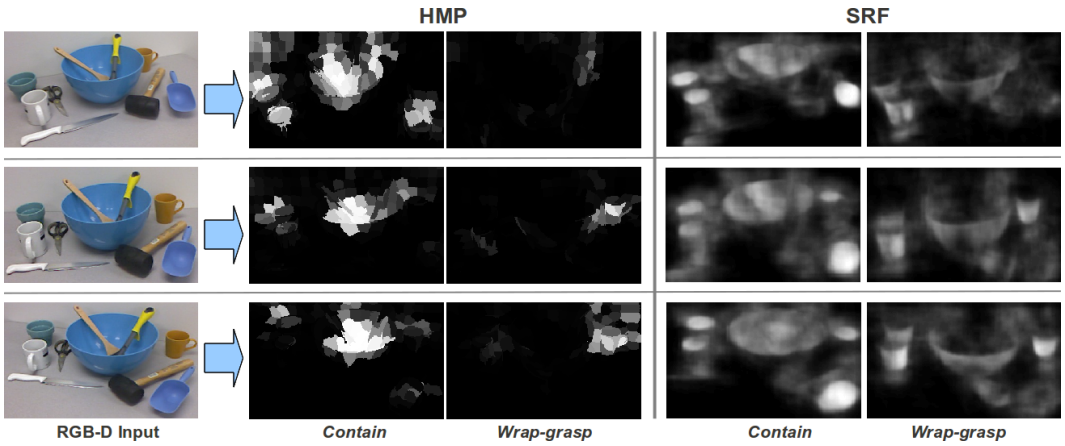


Figure 5.4: Results of affordance detection across three different input RGB-D frames (left) using S-HMP (middle) and SRF (right) over the cluttered sequence: two target affordances per method – `contain` (l) and `wrap-grasp` (r). Brighter means higher probability of the target affordance.

Affordance	Non-cluttered subset (single objects)		Cluttered subset (multiple objects)	
	S-HMP (F_β^w, R_β^w)	SRF (F_β^w, R_β^w)	S-HMP (F_β^w, R_β^w)	SRF (F_β^w, R_β^w)
`grasp`	0.367, 0.149	**0.423**, **0.173**	0.398, 0.268	**0.414**, **0.286**
`cut`	0.373, 0.043	**0.438**, **0.051**	0.350, 0.143	**0.490**, **0.200**
`scoop`	0.415, 0.046	**0.612**, **0.120**	0.348, 0.195	**0.634**, **0.377**
`contain`	**0.810**, 0.168	0.780, **0.170**	0.588, 0.334	0.588, **0.382**
`pound`	**0.643**, 0.035	0.606, **0.042**	0.474, 0.147	**0.553**, **0.172**
`support`	0.524, 0.030	**0.561**, **0.047**	0.451, 0.116	**0.485**, **0.171**
`wrap-grasp`	0.767, 0.102	**0.800**, **0.107**	0.394, 0.269	**0.504**, **0.342**
Mean	*0.557, 0.082*	***0.603, 0.101***	*0.429, 0.210*	***0.521, 0.275***

Table 5.1: Performance over the RGB-D Affordance Dataset. (Left) Non-cluttered subset and (Right) Cluttered subset.

Feature Sets	S-HMP F_β^w	SRF F_β^w
`Depth+SNorm+PCurv+[SI/CV]`†	0.557 (+0.018)	0.603 (+0.072)
`Depth+SNorm+PCurv`	0.562 (+0.023)	0.601 (+0.070)
`Depth+SNorm`	0.547 (+0.008)	0.599 (+0.068)
`Depth`	0.539	0.531

Table 5.2: Ablation experiments. $+x$ indicates the amount of change over `Depth`. †Since `SI` and `CV` are related measures, the best results using either one of them are reported. `SI` and `CV` yields the best results for HMP and SRF respectively.

Method	r_a %	d_a %
RF	85.3	62.5
SRF	93.5	87.0
SAE [163]	93.7	88.4
S-HMP	**95.2**	**92.0**

Table 5.3: Results on the Cornell Grasping Dataset.

S-HMP and SRF. We used the same train/test splits for both methods,and report averaged results over random splits of the RGB-D Affordance Dataset from [204]. Table. 5.1 summarizes the two detectors' performance over the seven affordance labels considered.

From the results, we can see that SRF consistently outperforms S-HMP in both evaluation metrics over both subsets of the RGB-D Affordance Dataset (Table 5.1). The difference is most significant in the cluttered subset where SRF outperforms S-HMP by 0.092 for F_β^w and 0.065 for R_β^w. This shows that the predictions from SRF are not only more accurate for the top ranked (primary) affordance, it also predicts more reasonable secondary affordances compared to S-HMP. This is further confirmed when we compare example outputs in Fig. 5.4. Not only are the SRF predictions better aligned to actual objects (unlike the superpixels of S-HMP), they also generalize to novel categories: e.g. the blue scoop is detected as `contain` by the SRF which is a reasonable secondary affordance while S-HMP has no detectable responses. Another outstanding aspect of SRF compared to S-HMP is that the predictions are performed per-pixel in *real-time* unlike S-HMP which takes minutes per image since the SVM classifier has to be run over all superpixels. In terms of feature ablations (Table 5.2), we see that the SRF improves most with the addition of `SNorm` while S-HMP benefits most with the addition of `PCurv` and both approaches saturate with `SI/CV`. A possible explanation for these differences is that the sparse-codes of S-HMP are already learning normal based representations from depth alone while principal curvatures are not as easily derived.

We compared all approaches with SAE over the Cornell grasping dataset using

the r_a and d_a metrics summarized in Table 5.3. In order to highlight the contribution of the structured constraints in the SRF, we trained a standard random forest (RF) with 20 trees over the annotated grasping rectangles in the dataset, using the *same* feature set of the SAE: `RGB` + `Depth` + `SNorms`. We note first that using a standard RF results only in mediocre performance. By adding the structured constraints and the proposed robust features, the SRF is able to achieve recognition and detection performances comparable to the deep learning based SAE. S-HMP outperforms the other approaches by a large margin, achieving state-of-the-art performance for this dataset. It is important to note, however, that the SRF provides very reasonable predictions of graspable locations with pixel-wise accuracy (Fig. 5.5), within a *fraction* of the time needed for inference using SAE (30s) or S-HMP (90s) vs. 0.1s in SRF.

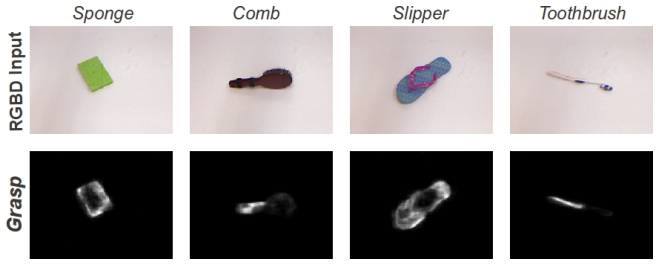


Figure 5.5: Grasping locations predicted by SRF. (Top) Input RGB-D images for four example objects. (Bottom) Predicted graspable locations. Notice the large difference in shape of the graspable regions. Brighter means higher probability.

5.5 Conclusions

We have described a fast, pixel-accurate affordance detector based on SRF in this chapter. Similar to other detectors based on SRFs explored in this thesis, our approach draws its success from the robust geometric features extracted from the RGB-D image. Compared to S-HMP and SAE, its strength lies in its generalizability to novel objects and affordance categories. As it predicts pixel-accurate affordances, the approach is robust to clutter, occlusions and viewpoint changes. Finally, as inference using SRF is extremely efficient, our approach is able to provide reasonable predictions within 0.1s which makes it more suitable for practical robotic applications compared to other (slower) but more accurate approaches (S-HMP and SAE).

The ability to detect affordances or functionalities, as was noted in the opening remarks of this chapter, is key to generalizing object recognition beyond its outward appearance. In terms of solving the FGO problem (§1.1), we move one step closer towards linking high-level knowledge as affordances themselves carry important *semantic* information of object attributes which can be found in natural language. We discuss the role of language with final remarks and insights in the concluding chapter of this thesis next.

Chapter 6: Closing the Semantic Gap using Language

In previous chapters, we have shown a gradual progression of using Gestalt to higher-level visual tasks. Starting from local cues and Gestalt operators, we have proposed an approach for border ownership; which aids contour-based recognition of object categories that share similar shapes (Chapters 2 and 3). From local cues, we detect bilateral and curved symmetries in images with clutter. By embedding a symmetry prior into a MRF-based representation of the image edges, we are able to segment symmetrical regions, linking symmetry (a key Gestalt) with the higher-level task of visual segmentation (Chapter 4). Finally, in Chapter 5, we demonstrate a fast and effective approach for detecting *affordances* of tool parts in real images with clutter and occlusions, which models Gibson's notion of "direct perception" [83] and agrees with Marr's viewpoint of invariant representations [192].

These approaches are important steps towards solving the figure-ground organization (FGO) problem, which we have argued in §1.1 to be important in bridging the so-called "semantic gap" between high-level (semantic) representations of the world with visual representations. The remaining question is how one can leverage from the approaches developed in the precedent chapters to actually link up with semantic or linguistic representations? In this concluding chapter, we present fu-

ture research directions that exploit structure from language in similar ways as we have done before to produce *simpler* linguistic representations that are *closer* to the mid-level visual features developed so far. In some sense, we are proposing a analog of Gestalt for language (though in the opposite direction), so that it is easier to find a common canonical feature space for linking these two modalities together.

6.1 Introduction

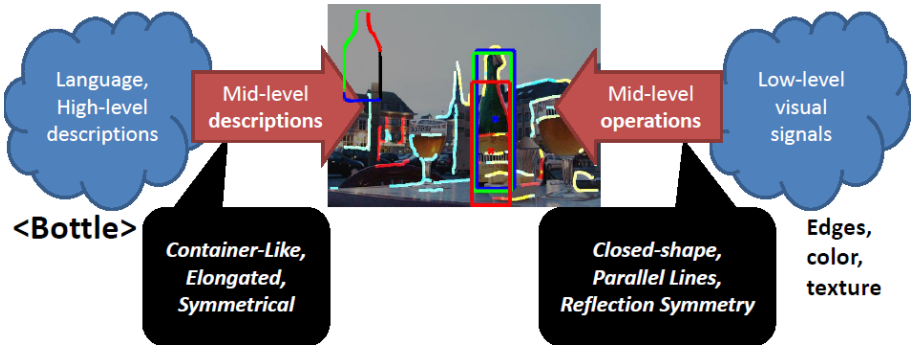


Figure 6.1: Illustrating how language and vision can be used together for e.g. detecting a bottle in the image. We produce *mid-level* representations on from both language (left) and vision (right) so that one can use text for describing the bottle's attributes which would then activate the correct visual operators for detection.

Throughout this thesis, we have advocated for solving the FGO problem as a key driver for various *visual* approaches presented in previous chapters. However, the FGO problem or the semantic gap in general, has two facets: Vision and Language. In this thesis, we have focused on the former and have largely ignored the latter, which we will address here as future work. The main idea is summarized in Fig. 6.1 where the goal is determine a common representation between these two

modalities so that a common task (e.g. object detection or scene understanding) can be enhanced. We have showed in this thesis that it is possible to use Gestalt in a principled manner for obtaining mid-level visual features, but an analog in language remains elusive. The key challenge, therefore, is finding such a representation from language and combining it with vision in a principled manner for solving multimodal tasks. We present some works from other communities in §6.2 and propose possible research directions in §6.3.

6.2 Related Works

In this section, we expand our review of related works from §1.3 to include works from the Natural Language Processing (NLP) and multimedia (MM) communities that solve similar problems but with a different (linguistic and multimodal) perspectives. We also present related works on *visual attributes*, which we show in §6.3.1 to be a good common representation between language and vision.

6.2.1 Vision and language from the NLP and MM communities

We have discussed the use of language by the Computer Vision community as a contextual tool for object recognition or scene understanding tasks in §1.3.2. Here we present relevant works from the NLP and MM communities that address similar issues.

In the NLP community, the most relevant works that explored processing raw linguistic data to obtain semantically meaningful information for integration with

vision is known as "semantic parsing". Among the most well-known is the Combinatory Categorical Grammar (CCG) of Steedman [260] which parses text into an a logical lambda expression that captures semantic meaning. A CCG is defined by a set of lexicon and a small set of combinatory rules. Each entry in the lexicon is a word-category pair that encodes semantic relationships. For e.g. the entry $Boston \vdash N : \lambda x.place(x)$ pairs the word "Boston" with a category that has syntactic type N (noun) and meaning $\lambda x.place(x)$. The output of the CCG parse of a sentence is a logical expression consisting of lambda expressions that encode the meaning and syntax of the sentence at the same time. Using CCG or other related semantic parsers [38, 125, 172], research in NLP have tried to integrate high-level knowledge to improve visual tasks and vice versa. This is known as the "grounding" problem in linguistics, where the goal is to embed real-world meaning so that ambiguity in both the linguistic and vision domains can be reduced. The key to do this is to train the semantic parser so that (in the case of CCG), the appropriate lexicon and mappings relevant to the task can be used. There are numerous works that vary in their learning approaches, representations and tasks. [29] uses Reinforcement Learning (RL) to map instructions to meaning for the task of mapping high-level instructions to automated system commands to be executed in a operating system's GUI. There have also been research into grounding instructions with the environment that the agent (robot) is in, with most work focused on robot navigation. [290] used a specialized MRF to encode transition probabilities on how the environment and its objects occur with observational probabilities derived from natural language directions. Inference on the MRF produced the most likely path

given noisy directional instructions from humans. [196] uses supervised learning on a statistical machine translation framework to parse natural language instructions to formal expressions, upon which a path-finding algorithm is applied to determine the optimal path to choose. Similarly, the work of [128] trains in a supervised manner a semantic parser for mapping instructions to navigation using automatically induced labels. Specifically, the authors learned a mapping between natural language instructions into a "meaning space" by determining the appropriate parameters for a Probabilistic Context Free Grammar (PCFG) when sentences are paired with some form of ambiguous description. [267] uses a graphical model representation to encode semantics (known as G^3) to learn a mapping between instructions and actions via demonstrations that are to be performed by a robot (e.g. move, load pellets, reverse etc.). The training data are natural language descriptions of the environments and instructions obtained from Amazon Mechanical Turks. [195] combines both visual information of objects on a table with linguistic descriptions to learn a mapping between attributes (color only) and objects. Finally, [145] combines language and visual perception together in a framework known as Logical Semantics with Perception so that scenes with objects and a linguistic query are parsed into logical expressions, and using a pre-defined environment (essentially a knowledge-base of how objects come together and interact), outputs the appropriate locations of objects that corresponds to the query.

There has also been significant development of techniques for fusing language and visual information together in the MM communities. By viewing this as a multimodal fusion problem, several approaches have been proposed along the realms of

context-based image retrieval, tag-based retrievals, image to image suggestions and image-to-text production etc. Many of the works are based on a technique known as Canonical Correlation Analysis (CCA) proposed in the 30s by Hotelling [110]. The idea is simple: CCA takes two features vectors from two different modalities (text and vision), and attempts to determine a set of basis vectors (for each modality) such that the correlations of the projections of the features in both modalities are mutually maximized. The basis vectors therefore define a new latent semantic space that is shared by both language and vision, simplifying the cross-modal retrieval tasks mentioned earlier. Recent works such as those of [95] have extended CCA to a kernelized version (Kernel CCA) to handle non-linear transformations and more complex cross-model relationship by first embedding a kernel to map the original features to higher dimensions. Works such as [116,129,197] have shown good results for action recognition, audio-video syncing/detection and object recognition. [85] recently introduced a 3-view approximate KCCA approach such that the kernels can be computed over extremely large datasets with very good cross-model retrieval results of tags and images. Finally, [5] proposed an extension to KCCA known as Deep CCA (DCCA) where the kernels for KCCA are first learned using deep neural networks before KCCA is applied.

6.2.2 Visual attributes

Attributes are intrinsic properties of entities that are *immutable* – that is, they do not change with time, space or location. Systems with such capabilities are

able to recognize new, unseen entities based simply from these shared properties, enabling them to scale up to a wide variety of categories. Because of its attractive properties, recognition based on shared attributes had been an area of active research in Computer Vision as it provides a strong invariance against changes in viewpoints and environmental conditions.

Among the earlier works in this direction is the work of [17] which introduced the idea of "geons": a set of 32 primitive 3D geometric shapes that were proposed to be crucial for human object recognition. Following similar lines, but from a more neurologically plausible viewpoint, is the work of [235] that proposed "HMAX", a hierarchical model of the human visual cortex for recognizing objects. The model begins with simple inputs from low-level cells (directed edge responses) that are then fed to more complex composite cells that pool responses over a larger visual field (longer edges). The work of [74] uses a similar hierarchical approach consisting of contour fragments but introduced a generative model that learns the appropriate combinations at different layers of the model from training data. Part-based models, introduced by [70] and extended further using HOG (Histogram of Gradient) features trained in a discriminative framework [69] has become a standard baseline for object recognition can also be viewed as an effort on this direction since the approach essentially recognizes shared parts (e.g. arms, torso, eyes, etc.).

Attributes in Computer Vision has also been widely used in several learning approaches, notably in the work of [152] where the authors proposed the notion of "zero-shot" and "one-shot" transfer learning where attributes learned from a dataset are used to classify a new object that has none or a single training exemplar. More re-

cently, attributes have been used in a discriminative approach for object recognition by training specific classifiers for particular hand-selected attributes of objects [64] and extended further for recognizing object categories [63]. Silberer et al. [248] directly used such attributes to learn a joint model of textual attributes and visual features for cross-modality tasks. A recent development was the introduction of "relative attributes" [221] where attributes are used in a ranked manner to learn a relative scale for comparison, which yields better results than simply using manually specified attributes for learning a binary decision boundary. Other authors have recently addressed the issue of discovering attributes from data instead of manually specified terms. [14] uses image captions and annotated labels to learn discriminative visual attributes from web based images. To ensure that meaningful attributes are obtained, [220] proposed an active learning approach with humans in the loop. Along similar lines but over a much larger dataset, [222] used the MIT-SUN dataset of scene categories with the help of Amazon Mechanical Turks to re-categorize and discover novel scene-level attributes that are shown to be more precise than the original categories found the dataset. There has also been significant effort in using linguistic resources to discover attributes and to induce semantically meaningful ontologies for comparison and classification. [240] used the ImageNet dataset to learn attributes that parallels the WordNet ontologies of object categories. [275] uses visual concepts obtained from the Internet to design a novel "classeme" descriptor for efficient object recognition. There has also been a recent parallel trend where instead of learning/discovering so-called "nameable" attributes – those with precise semantic meaning understandable by humans, to discovering "un-nameable" attributes – at-

tributes derived directly from low-level visual features. [302] used such un-nameable attributes for object categorization where the authors designed a category-attribute matrix that allows for object category separation and is at the same time suitable for learning. Along similar lines, the authors in [54] used a latent CRF model to learn an attribute classifier that discovers attributes that are both semantically meaningful and have visually detectable features. The initial model is further refined in a second training step with humans in the loop to determine meaningful attributes so that a name can be assigned.

6.3 Future Research Directions

We present here three future research directions that we believe will lead to a complete solution for the FGO problem. First, we present ideas for developing an appropriate mid-level representation of language via affordances, which we call *grounded affordances*. Second, we propose to use CCA, popular in the MM communities to associate mid-level linguistic features with mid-level visual representations developed in earlier chapters. Finally, we discuss the role of deep learning as a modern and viable alternative for multimodal association.

6.3.1 Language grounding of affordance-based attributes

The first task is to determine the appropriate level of representation in language so that it is even possible to learn a reliable mapping. This has remained an open research question (§6.2.1 and §1.3.2). The key challenge is finding a representa-

tion that is able to capture the complexity of language while associating them with their appropriate visual counterparts at the same time. As was noted in Chapter 1, this is a very difficult problem as it deals with the fundamental semantic gap between language and vision (Fig. 6.2 (top)).

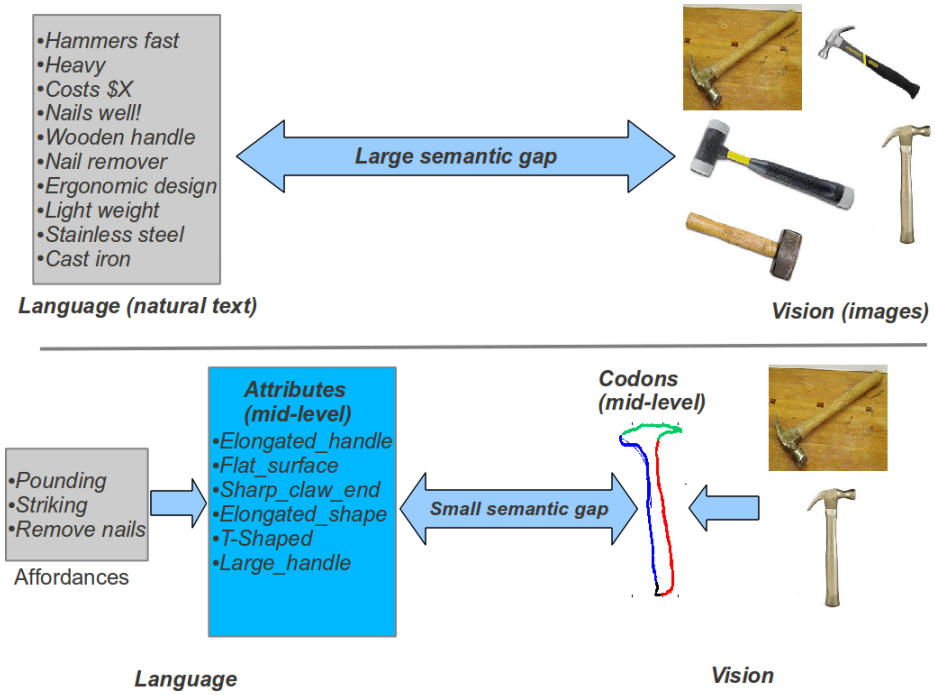


Figure 6.2: Why grounding attributes using affordances makes sense. (Top) Semantic gap between language (text) and low-level visual representation is large – one does not naturally describe a complex object in terms of their low-level visual representations or appearances. (Bottom) Grounding attributes using affordances provides a mid-level representation in language to bridge the gap with mid-level visual representations (here we show grouped contours or codons from Chapter 3).

Faced with this formidable challenge, it is clear that using language at the semantic level is not appropriate. We therefore draw our motivation on recent re-

search into attributes (§6.2.2) that is claimed to be some form of "mid-level" visual concept that has properties that is shared among objects within the same category. Additionally, we assume the objects of interest have some (a small number) associated functions that we want to categorize[1] such as tools (see Chapter 5). Our key hypothesis is that attributes grounded via affordances, or *grounded attributes*, can be easily associated in both the language and vision space (Fig. 6.2 (bottom)). This hypothesis is supported by the observation that when asked to describe objects in a scene as reported in [222], most subjects expressed the highest degree of confidence when they described the object via its functionalities or affordances. This indicates that when we "reduce" the complexity of textual description down to the level of affordances, the object or scene is still understandable by others.

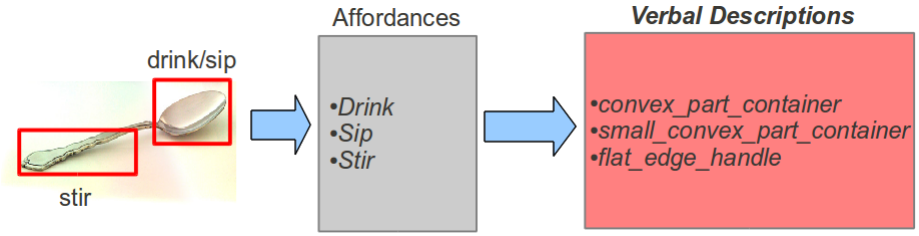


Figure 6.3: A typical tool (spoon), with its associated verbs and verbal descriptions. Note that verbs contain high-level semantic meanings while the verbal description is closer to the visual space.

To do this, we propose that the functionality/affordance of a particular target m be expressed as a set of *verbal descriptions*, $\mathcal{V}_m = \{v_1, v_2, v_3, \cdots, v_n\}$, in the

[1] Objects that do not serve a particular function are not well defined (e.g. a "pear"), and complex objects that have *numerous* uses (e.g. a "car") are not considered since it is their *parts* that affords the functions – "wheels", "doors" etc.

language space. Verbal descriptions are *not* the same as verbs but are related to verbs in a *consequential* relationship. To be more concrete, let us give an example of a typical tool, e.g. "spoon", with its associated verbs (affordances) and its verbal descriptions (Fig. 6.3). A typical list of associated verbs[2] would be `{drink, sip, stir}`, but such verbs themselves contain a significant amount of semantic content so that it is hard to find any visual association without further processing. However, looking at the verbal descriptions, we have `{convex_part_container, small_convex_part_container, flat_edge_with_handle}` which corresponds to a simplified description of the part of the spoon that affords a particular function/verb. Such descriptions can be seen as a form of *consequence* due to the associated verbs: e.g. Because a spoon has as a verbal description `convex_part_container`, it induces as a consequence the verb `drink`. Since the descriptions describe certain visual aspects of the target that contribute to the verb, it is therefore more suitable for learning a visual association with the language space than the actual verbs themselves. Another reason why we choose verbal descriptions is because they relate fundamentally to the innate qualities of the target, essentially defining its purpose and hence identifying it. This is in contrast with other kinds of attributes (whether visual or textual) such as for e.g. color, texture, which are shared across categories but do not fully define the target. For example, a "red" or "rough" attribute can correspond to many different objects, and is nearly not as useful as `{sharp_edges, has_cutting_edge}` verbal descriptions. Since such descriptions are grounded to the physical properties of the target, we call them *grounded attributes* in order to

[2]Verbs are the most direct representation of affordances in language

emphasize this fact.

An important challenge that one must address is determining $\mathcal{V}_m$. There are several ways to do this. Firstly, one can learn a (limited) set of possible grounded attributes from structured ontologies such as WordNet or from online dictionary definitions (e.g. Wiktionary[3]). This is done by extracting definitions of the noun or from parsing of example verses that demonstrate how the noun is used (and extracting the correct verbs). Such an approach, however, results in a very limited and noisy set of associated verbs. Take for example, the definition of "spoon" (from Wiktionary) is:

> *spoon (plural spoons):*
>
> *1. An implement for eating or serving; a scooped utensil whose long handle is straight, in contrast to a ladle.*
>
> *2. An implement for stirring food while being prepared; a wooden spoon.*
>
> *3. A measure that will fit into a spoon; a spoonful.*

One can see that the definition is written in a high level because it is meant to be understood by humans, so only a few key verbs can be found using this approach: `{eat, serve, stir, measure}`. Additional functional description of the object such as "straight handle", "scooped utensil" which are visually important could be used but requires additional processing since such definitions often imply another word (e.g "handle", "scooped", "wooden spoon") that may result in a circuitous definition or use inconsistent word descriptions such that it is hard to

[3] http://en.wiktionary.org/

associate them with say another spoon-like object. Another possible way to expand this verb association would be to mine for object-verb relationships from large data such as in [299]. However, this approach is problematic because unstructured text from copora like Gigawords [89] often do not contain a clear definition or description of the object used under a variety of conditions. The reason is because when we write about objects in text, they are almost always understood at the visual level rather than at the textual level. This means that it would be extremely hard to extract the verbs or verbal descriptions from large unstructured corpora.

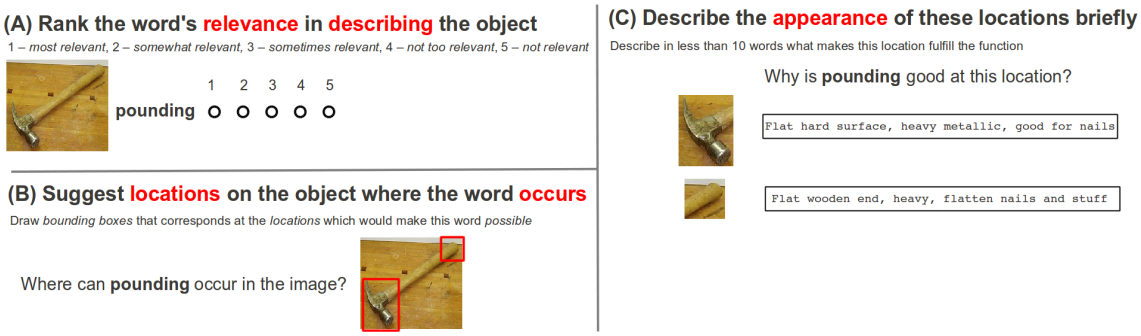


Figure 6.4: Interface used for soliciting verbal description responses from AMT turkers. (A) Ranking the proposed verb in terms of relevance. (B) Selecting locations via bounding boxes which corresponds to the verb. (C) Brief description of the location in terms of its appearance.

Clearly, since the data that we need to construct $\mathcal{V}_m$ is not readily available or easily processed in the wild, we propose to extract such information from humans directly via crowd-sourcing methods such as Amazon Mechanical Turk (AMT). Starting from a list of target objects/categories, we will use standard syntactic parsing methods (e.g. [224]) together with a dependency parser (e.g. [36]) to extract the

initial list of verbs associated with the object's affordance. We then use this initial list of verbs to design an interface that will enable us to solicit responses using AMT turkers as shown in Fig. 6.4. A typical image of the target is shown together with one of the selected verbs from the list. The task is simple and consists of two steps: 1) draw a bounding box indicating the part (if possible) of the object that affords the particular verb and 2) write down a brief (< 10 words) description of the part based on its *appearance*. Different images will be shown yielding a large number (at least 20) of the parts that are associated with the verb together with potential responses for their verbal descriptions. With these verbal descriptions, we will apply a clustering step to determine the most consistent and highest ranked descriptions for $\mathcal{V}_m$, together with a set of description image parts $\mathcal{I}_m$ that will be used for learning a mapping between language and vision space. How this can be done is described next.

6.3.2 Learning a canonical multimodal space

Given the set of visual descriptions $\mathcal{V}_m$ and image parts $\mathcal{I}_m$, we would like to formulate a learning mechanism that learns a mapping between $\mathcal{V}_m$ and $\mathcal{I}_m$ – essentially, a mapping between the components in the language space of $\mathcal{V}_m$ and the visual space of $\mathcal{I}_m$. Since the two spaces are very different, we propose using Canonical Correlation Analysis (CCA) or the kernelized version (KCCA) [95] to determine this mapping into a common latent space as shown in Fig. 6.5. Using CCA/KCCA makes sense because: 1) the features extracted from $\mathcal{V}_m$ and $\mathcal{I}_m$ are describing the

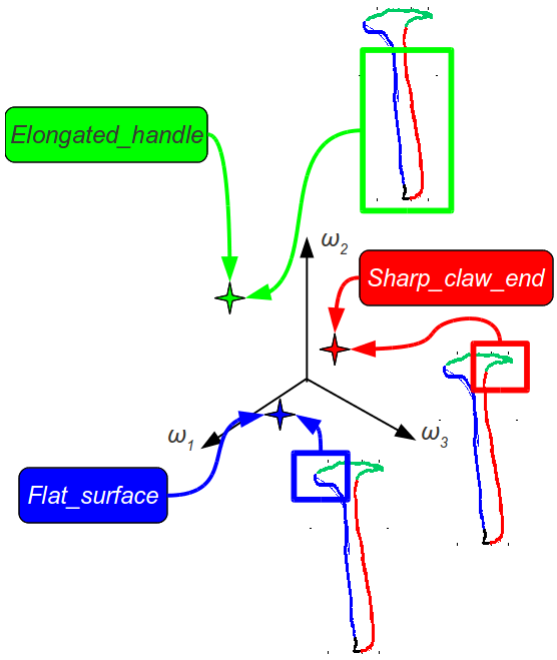


Figure 6.5: Illustrating CCA applied between verbal descriptions associated with image features to form a common latent space that generates the information in the two modalities. The verbal descriptions, within boxes of the same color, are paired with their visual representations (here we use codons described in Chapter 3) that are highlighted with the same color. The space here is represented by the first canonical variate: $w^1 =< \omega_1, \omega_2, \omega_3, \cdots >$ which maximizes the correlation of the two features.

same object parts, and therefore must be generated by a common (latent) source and 2) the two spaces are coming from two different views (modalities), and the problem of fusing language and vision can be seen a cross-modal problem that CCA/KCCA is designed to handle. We first describe CCA followed by KCCA. Without loss of generality, we will denote $(\mathbf{X}_1, \mathbf{X}_2)$ the data from vision and language respectively.

CCA [110] attempts to find linear projections $\mathbf{w}_1, \mathbf{w}_2$ of two random vectors $\mathbf{X}_1 \in \mathbb{R}^{n_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{n_2}$ such that their correlation is mutually maximized:

$$\begin{aligned}(\mathbf{w}_1^*, \mathbf{w}_2^*) &= \underset{\mathbf{w}_1, \mathbf{w}_2}{\text{argmax}}\, \texttt{corr}(\mathbf{w}_1^\top \mathbf{X}_1, \mathbf{w}_2^\top \mathbf{X}_2) \\ &= \underset{\mathbf{w}_1, \mathbf{w}_2}{\text{argmax}}\, \frac{\mathbf{w}_1^\top \Sigma_{12} \mathbf{w}_2}{\sqrt{\mathbf{w}_1^\top \Sigma_{11} \mathbf{w}_1 \mathbf{w}_2^\top \Sigma_{22} \mathbf{w}_2}} \end{aligned} \tag{6.1}$$

where $\Sigma_{11} \in \mathbb{R}^{n_1 \times n_1}, \Sigma_{22} \in \mathbb{R}^{n_2 \times n_2}$ are the covariances of $\mathbf{X}_1, \mathbf{X}_2$ and $\Sigma_{12} \in \mathbb{R}^{n_1 \times n_2}$ their cross-covariance. These covariances are derived from the total covariance matrix $\hat{\Sigma}$ defined as:

$$\hat{\Sigma} \triangleq \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} = \hat{\mathbb{E}} \left[\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix}^\top \right] \tag{6.2}$$

where $\hat{\mathbb{E}}$ is the empirical expectation defined for a function $f(\mathbf{a}, \mathbf{b})$:

$$\hat{\mathbb{E}}\left[f(\mathbf{a}, \mathbf{b})\right] = \frac{1}{m} \sum_{i=1}^{m} f(\mathbf{a}_i, \mathbf{b}_i) \tag{6.3}$$

An important observation is that the objective in (6.1) is invariant to changes in scales for $\mathbf{w}_1$ and $\mathbf{w}_2$, which means that the same results can be achieved by simply maximizing the numerator while constraining the projections in the denominator to be of unit variance:

$$(\mathbf{w}_1^*, \mathbf{w}_2^*) = \underset{\mathbf{w}_1^\top \Sigma_{11} \mathbf{w}_1 = 1, \mathbf{w}_2^\top \Sigma_{22} \mathbf{w}_2 = 1}{\text{argmax}} \mathbf{w}_1^\top \Sigma_{12} \mathbf{w}_2 \tag{6.4}$$

Eq. (6.4) has a closed form solution using Lagrange multipliers [95], which can be reduced into a pair of standard eigenproblem:

$$\begin{cases} \Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\mathbf{w}_1 = \lambda^2\mathbf{w}_1 \\ \\ \Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\mathbf{w}_2 = \lambda^2\mathbf{w}_2 \end{cases} \tag{6.5}$$

Solving eq. (6.5) yields the eigenvectors $(\mathbf{w}_1^*, \mathbf{w}_2^*)$ that defines the canonical bases for $(\mathbf{X}_1, \mathbf{X}_2)$ respectively. Choosing the eigenvectors (w_1^1, w_2^1) corresponding to the largest eigenvalue λ^2 yields the largest correlation between the first canonical variates: $(x_1', x_2') = (w_1^{1\top}\mathbf{X}_1, w_2^{1\top}\mathbf{X}_2)$ (and so on) in the new space.

KCCA extends CCA in a similar way. Instead on finding linear projections $(\mathbf{w}_1, \mathbf{w}_2)$ in CCA, KCCA finds pairs of *non-linear* projections by first projecting the data into a higher dimensional space using the so-called "kernel-trick". Denoting a sequence of data of length m from the two views as $\mathcal{X}_1 \in \mathbb{R}^{m\times n_1}, \mathcal{X}_2 \in \mathbb{R}^{m\times n_2}$ (each sequence is a row in the data matrices), we can rewrite the definition of their covariances in eq. (6.2) as $\Sigma_{11} = \mathcal{X}_1^\top\mathcal{X}_1, \Sigma_{12} = \mathcal{X}_1^\top\mathcal{X}_2$ (and similarly for Σ_{21}, Σ_{22}). The projections, $\mathbf{w}_1, \mathbf{w}_2$ can also be rewritten as the projection of the data matrices with respect to $\alpha_1 \in \mathbb{R}^m, \alpha_2 \in \mathbb{R}^m$ such that we have: $\mathbf{w}_1 = \mathcal{X}_1^\top\alpha_1$ and $\mathbf{w}_2 = \mathcal{X}_2^\top\alpha_2$. Substituting these into eq. (6.1) yields:

$$(\alpha_1^*, \alpha_2^*) = \underset{\alpha_1,\alpha_2}{\operatorname{argmax}} \frac{\alpha_1^\top\mathcal{X}_1\mathcal{X}_1^\top\mathcal{X}_2\mathcal{X}_2^\top\alpha_2}{\sqrt{\alpha_1^\top\mathcal{X}_1\mathcal{X}_1^\top\mathcal{X}_1\mathcal{X}_1^\top\alpha_1\alpha_2^\top\mathcal{X}_2\mathcal{X}_2^\top\mathcal{X}_2\mathcal{X}_2^\top\alpha_2}} \tag{6.6}$$

Denoting the kernel matrices for each view as $K_1 = \mathcal{X}_1\mathcal{X}_1^\top \in \mathbb{R}^{m\times m}$ and $K_2 = \mathcal{X}_2\mathcal{X}_2^\top \in \mathbb{R}^{m\times m}$, we derive from eq. (6.6) the dual form of eq. (6.1):

$$(\alpha_1^*, \alpha_2^*) = \underset{\alpha_1,\alpha_2}{\operatorname{argmax}} \frac{\alpha_1^\top K_1K_2\alpha_2}{\sqrt{\alpha_1^\top K_1^2\alpha_1\alpha_2^\top K_2^2\alpha_2}} \tag{6.7}$$

However, eq. (6.7) has problems in practice because without constraining the directions of the projections, one can easily find trivial projections in the higher dimensions. The solution is to regularize the projections using Partial Least Squares (PLS) so as to penalize the norms of the associated weights in the denominator:

$$(\alpha_1^*, \alpha_2^*) = \underset{\alpha_1,\alpha_2}{\operatorname{argmax}} \frac{\alpha_1^\top K_1 K_2 \alpha_2}{\sqrt{(\alpha_1^\top K_1^2 \alpha_1 + \kappa_1 \|\mathbf{w}_1\|^2).(\alpha_2^\top K_2^2 \alpha_2 + \kappa_2 \|\mathbf{w}_2\|^2)}} \tag{6.8}$$

with κ_1, κ_2 as the regularization coefficients.

Similar to the primal form of the objective in eq. (6.6), the objective in dual form is not affected by changes in scales for α_1 and α_2, which leads us to the simplified objective where we constrain the terms in the denominator to be unity:

$$(\alpha_1^*, \alpha_2^*) = \underset{\alpha_1^\top K_1^2 \alpha_1 + \kappa_1 \|\mathbf{w}_1\|^2 = 1, \alpha_2^\top K_2^2 \alpha_2 + \kappa_2 \|\mathbf{w}_2\|^2 = 1}{\operatorname{argmax}} \alpha_1^\top K_1 K_2 \alpha_2 \tag{6.9}$$

The solution to eq. (6.9) can be found using Lagrange multipliers which is again reduced into a pair of standard eigenproblem:

$$\begin{cases} (K_1 + \kappa_1 I)^{-1} K_2 (K_2 + \kappa_2 I)^{-1} K_1 \alpha_1 = \lambda^2 \alpha_1 \\ (K_2 + \kappa_2 I)^{-1} K_1 (K_1 + \kappa_1 I)^{-1} K_2 \alpha_2 = \lambda^2 \alpha_2 \end{cases} \tag{6.10}$$

Solving eq. (6.10) yields as the top eigenvectors (α_1^*, α_2^*) which maximizes the correlation of the projections in the new space.

Depending on the complexity of representations used in the text and vision domains, one can choose to use the linear CCA projections or non-linear KCCA projections. CCA is fast and easy to optimize but in practice does not handle data with large non-linearities between the two modalities, while KCCA depends on the assumption that the kernels K_1, K_2 can be inverted, which limits its applicability

when m gets large, although several fast approximations had been proposed [85,189, 281] to overcome this limitation.

Once the common latent space has been learned, it is straightforward to generate novel views in vision given input text descriptions (and vice-versa) of the target. The most naive way to do this (but the fastest) would be to use a nearest neighbor (NN) approach where transformed visual centers closest to the transformed language centers are returned directly. One can also apply Support Vector Machines (SVMs) to learn classifiers within this space as well.

There are several important issues that need to be addressed for this approach to work. Firstly is the representation of the language and vision data that will be used in CCA/KCCA. For visual data, we propose to directly use the mid-level representations: ownership, grouped contours (codons), symmetry and affordances developed in the thesis. For language, since we are working directly with verbal descriptions of grounded attributes $\mathcal{V}_m$, one simple way would be to first cluster all the attributes in $\mathcal{V}_m$ to remove redundant attributes (and reducing the dimensions) using techniques such as, k-means or probabilistic latent semantic analysis (pLSA) [108] or latent Dirichlet allocation (LDA) [18]. Using the reduced set of vocabulary of attributes, we can then represent each $v_k \in \mathcal{V}_m$ as a binary vector where 1 indicates the presence of the attributes and 0 otherwise. A linear kernel, $K_v(v_i, v_j) = v_i^\top v_j$ can then be use as a representation of the number of shared attributes for learning the mapping in CCA/KCCA. Other forms of representations are possible such as using a sub-sequence kernel [201] that counts the number of word sequence that are shared from a sentence that is generated (from Turker's response) to describe

the functionality of the object. Secondly, and related to the first is the issue of an appropriate distance measurement in the shared latent space. Most works surveyed in §6.2.1 had traditionally used standard Euclidean distance, but as was noted in [85], designing a new similarity measure that takes into account the magnitude of the eigenvalues in CCA produced better results than using Euclidean. We intend to experiment with several similarity measures in the latent space in the proposed work as well. Finally, a key issue if we are to use KCCA is the design of the vision and text kernels. There are numerous choices, such as a standard Radial Basis Functions (RBF) kernels or a d-degree polynomial kernels that can be used in both modalities. Motivated by the results reported in [5], we plan to experiment with kernels that tend to capture some form high-level semantics within the data, for example, by grouping longer contour fragments together (vision) and/or longer sequences of word/verbal descriptions that would capture long-range semantics.

6.3.3 Multimodal features from deep networks

A recent trend in Computer Vision and NLP is the learning of appropriate feature representations using so-called deep convolutional networks (CNN) [156]. Such deep networks have also been used successfully by a number of authors for learning cross-modality representations between language and vision e.g. [76, 123, 131, 208, 250, 252, 255]. The basic idea for earlier works [76, 208, 250, 255] is simple: train two different networks – one for vision and the other for language – and combine together their derived representations for solving joint language and visual tasks (e.g.

image-based retrieval using text and vice versa). Recently, [131] replaced the deep language models with a log-bilinear model for generating sentences describing the image (unlike simple tags used in prior works). Along similar lines, Karpathy et al. [123] demonstrated a deep bilinear model that links sentences to images that embeds fine-grained image patches corresponding to scene/object parts and their spatial relations, resulting in a more meaningful generated sentence. Socher et al. [252] combined language features derived from dependency parse trees with image features from a deep network to learn a joint multimodal space capable of retrieving images and generating sentences based on the dependency parse.

These works show that learning appropriate multimodal representations remains challenging due to the large semantic gap noted in §6.3.1 and in Chapter 1. A key issue with these approaches is that joint learning between vision and language occurs *after* features are derived from separate networks. This makes learning more difficult since the two modalities may contain biases that would not be recoverable in the final stage. To overcome this, we propose to learn features using deep networks in a more careful and supervised manner. Similar to the Deeply-supervised Nets of Lee et al. [157], where additional supervision of layers in the CNNs are shown to be useful for improving certain visual tasks, we believe a similar supervision, where mid-level linguistic attributes [248] can be paired our mid-level visual representations to guide the learning of a joint linguistic-visual feature in a principled manner. Along similar lines as the DCCA of Andrew et al. [5], we can then inject such jointly learned deep features into CCA to perform text or sentence based retrieval of images or to generate image descriptions.

6.4 Final Conclusions and Outlook

From the FGO problem, we have motivated the approaches presented in this thesis as a means to bring low-level visual signals, guided by Gestalt principles, to a richer and more meaningful mid-level representation. However, this is only one (the visual) side of the FGO problem which is the focus of this thesis. In this chapter, we discussed the linguistic side of the same problem, and proposed to use *grounded attributes* as a means to simplify linguistic representations for learning a joint canonical space via CCA that links vision and language together. We have also discussed how multimodal features could be derived from deep learning methods, where we propose to inject mid-level linguistic features within the network to guide the learning of visual features (and vice versa).

We believe the results and approaches presented in this thesis have important implications in related fields beyond Computer Vision. In terms of visual psychology, we have confirmed that their models of visual Gestalt make sense for tasks such as border ownership, contour-based recognition, detection of symmetries and segmentation of symmetrical parts; and extends to functionality detection as well. For robotics, our computational algorithms using SRFs and dynamic programming are fast and approach real-time performance, which should be useful for mobile intelligent agents working in real environments containing clutter, occlusions and lighting changes. Finally, for NLP, we have proposed research into mid-level representations of language which would be useful for learning joint visual-language models.

Appendix A: Generalizing the image torque to other patterns

Following the notations used in §2.3.1.3, we write down the following iso-contour functions for: 1) radial $f_r(x,y)$, 2) spiral $f_s(x,y)$ and 3) hyperbolic $f_h(x,y)$:

$$\begin{aligned} f_r(x,y) &= \text{atan}\left(\frac{y}{x}\right) \Rightarrow \nabla f_r(x,y) = \begin{pmatrix} \frac{y}{\sqrt{x^2+y^2}} \\ \frac{-x}{\sqrt{x^2+y^2}} \end{pmatrix} \\ f_s(x,y) &= x^2 - ay^2 \Rightarrow \nabla f_s(x,y) = \begin{pmatrix} x \\ -ay \end{pmatrix} \\ f_h(x,y) &= \ln(\sqrt{x^2+y^2}) - a\,\text{atan}\left(\frac{y}{x}\right) \Rightarrow \\ \nabla f_h(x,y) &= \frac{1}{x^2+y^2}\begin{pmatrix} ax-y \\ x+ay \end{pmatrix} \end{aligned} \tag{A.1}$$

which leads to the following expressions for the tangent vectors $g(x,y)$:

$$\begin{aligned} g_r(x,y) &= (x,y) \\ g_s(x,y) &= (ax-y, x+ay) \\ g_h(x,y) &= (ay,x) \end{aligned} \tag{A.2}$$

for some values of $a = \{\frac{1}{3}, 1, 3\}$. Substituting the corresponding $g(x,y)$ from eq. (A.2) in eq. (2.3) enables us to compute the alignment of the target pattern in the image.

Appendix B: Summary of Contour-Based Categorical Object Recognition Algorithm

Algorithm 1 summarizes the contour-based object recognition approach presented in Chapter 3. 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_I^m that contains the modulated torque per patch, $\tau_P^\omega, P \in I_e$ at every image point.

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 i^{th}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 eq. (3.2) 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 i^{th} 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

(sec. 3.3.2.2);

$\mathcal{O}_g \leftarrow \{O_g, O_g + 90, O_g + 180, O_g + 270\}$;

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 $V(a, b) = D_{sc}^{\tau}(\mathcal{R}'_{\{e\},\ldots,\{e+a\}}, \mathcal{M}_b)$ via eq. (3.10);

end

$E_{D_{sc}^{\tau}}(e, o_g) \leftarrow \min_{\mathcal{J},l} V$;

end

$E_{D_{sc}^{\tau}}(o_g) \leftarrow \min_e E_{D_{sc}^{\tau}}(e, o_g)$;

end

$E_{D_{sc}^{\tau}} \leftarrow \min_{o_g} E_{D_{sc}^{\tau}}(o_g)$;

for $e \leftarrow 1$ **to** d **do**

for $r_i \in \mathcal{R}'_e$ **do**
Convert $E_{D_{sc}^{\tau}}$ to weights $W_{D_{sc}^{\tau}}(r_i)$ via eq. (3.11);

Compute $\tau_{p_c r_i}^{\omega}$ via eq. (3.12);

end

end

for $(r, c) \leftarrow (1, 1)$ **to** (H, W) **do**
Compute $\tau_{P_s(r,c)}^{\omega}$ via eq. (3.13) for every patch $P_s(r, c)$ centered at (r, c) over I_e over all scales $s \in \mathcal{S}$;

$T_I^m(r, c) \leftarrow \max_{s \in \mathcal{S}} \tau_{P_s(r,c)}^{\omega}$;

end

end

Algorithm 1: Pseudocode for the proposed approach

Appendix C: Simulating Log-polar Coordinates in Cartesian Coordinates

We demonstrate here that the distance between two neighboring points (p, q) in Log-polar coordinates is equivalent to $1/r$ times their distance in Cartesian coordinates.

Proof. We consider a Log-polar system, where points have coordinates $(\rho, \theta) = (\ln r, \theta)$, with r the radius from the fixation center (pole) to the point. Let p and q be two neighboring boundary points. Their coordinates are (x_p, y_p) and (x_q, y_q) in a Cartesian system, and (ρ_p, θ_p) and (ρ_q, θ_q) in log-polar coordinate system, respectively. Let us denote,

$$dx = x_p - x_q,\ dy = y_p - y_q$$

$$d\rho = \rho_p - \rho_q,\ d\theta = \theta_p - \theta_q.$$

The distance between the points p and q in the log-polar coordinate system amounts to

$$D_{p,q} = \sqrt{d\rho^2 + d\theta^2} \tag{C.1}$$

The relationship between log polar and Cartesian coordinates is:

$$\rho = \ln \sqrt{x^2 + y^2},\ \theta = \arctan(\frac{y}{x})$$

Thus, using the transformation:

$$\begin{pmatrix} d\rho \\ d\theta \end{pmatrix} = \begin{vmatrix} \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} \end{vmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} \tag{C.2}$$

we obtain that

$$D_{p,q} = \sqrt{(d\rho^2 + d\theta^2)} = \frac{1}{r}\sqrt{(dx^2 + dy^2)} \tag{C.3}$$

where $\sqrt{x^2 + y^2}$ is the distance of the two points. Since in our implementation we use a 4-way MRF neighborhood, it is always 1.

Invoking this result to weigh the binary pairwise term, V_{pq}, in the energy minimization, we obtain that we have to weigh V_{pq} by $\frac{1}{r}$.

Finally, let us note, if instead we used a simple polar transform with coordinates (r, θ), the transformation from (dx, dy) to $(dr, d\theta)$ would be: $dr = \frac{x}{r}dx + \frac{y}{r}dy$ and $d\theta = \frac{-y}{r^2}dx + \frac{x}{r^2}dy$ following a similar derivation. □

Appendix D: Separability of orientation from translation components

We provide a geometric argument to demonstrate the separability of the orientation θ_{l_c} from its centroid location (x_{l_c}, y_{l_c}) in the computation of the symmetry axis l_c.

Proof. Without loss of generality, we assume that l_c separates two lines l_1, l_2 as illustrated in Fig. D.1. We parameterize lines in the Hough space by the coordinates r and θ, which gives us: $r_{l_{\{1,2,c\}}}$ (centroids) and $\theta_{\{1,2,c\}}$ (orientations). Since l_c bisects $\angle APC$ between l_1 and l_2, forming a pair of congruent triangles ($\triangle DPE \equiv \triangle EPF$). From this observation, we can write down the first relationship between θ_{l_c} and $\theta_{l_1}, \theta_{l_2}$:

$$\begin{aligned}\angle BOE &= \angle BPC \text{ as } (\triangle BOE \sim \triangle BPC) \\ &= \angle AOD \text{ as } (\triangle AOD \sim \triangle APC) \\ &= \theta_{l_2} - \theta_{l_c}\end{aligned}$$

Similarly, we have $\angle DOG = \theta_{l_c} - \theta_{l_1}$. Now since $\angle DOG \equiv \angle BOE$, because l_c bisects $\angle AOG$, we have

$$\begin{aligned}\theta_{l_c} - \theta_{l_1} &= \theta_{l_2} - \theta_{l_c} \\ \theta_{l_c} &= \frac{\theta_{l_1} + \theta_{l_2}}{2}\end{aligned} \tag{D.1}$$

Eq. (D.1) shows that θ_{l_c}, the orientation of the symmetry axis l_c, is fully defined by the orientation of l_1, l_2 and is independent of its centroid r_{l_c}. For this reason the orientation of the symmetry axis can be recovered independently of its position. On the other hand, we show next that r_{l_c} is a function of θ_{l_c}.

As $r_{l_1} = |OG|$, $r_{l_2} = |OC|$ and $r_{l_c} = |OE|$, this gives us

$$r_{l_2} = |OF| \cos(\angle AOD)$$

$$r_{l_1} = |OD| \cos(\angle DOG) = |OD| \cos(\angle AOD)$$

Since $(\triangle DPE \equiv \triangle EPF) \Rightarrow |DE| = |EF|$, we have

$$|OF| = |OD| + 2|DE|$$

$$|OF| = |OD| + 2(|OE| - |OD|) = 2|OE| - |OD|$$

$$|OE| = \frac{|OF| + |OD|}{2}$$

Replacing in the above the distance with the centroid of the lines and substituting for the angle from eq. (D.1) yields

$$\begin{aligned} r_{l_c} &= \frac{r_{l_2} + r_{l_1}}{2\cos(\angle AOD)} = \frac{r_{l_2} + r_{l_1}}{2\cos(\theta_{l_2} - \theta_{l_c})} \\ &= \frac{r_{l_2} + r_{l_1}}{2\cos\left(\frac{\theta_{l_2} - \theta_{l_1}}{2}\right)} \end{aligned} \tag{D.2}$$

which completes the proof. □

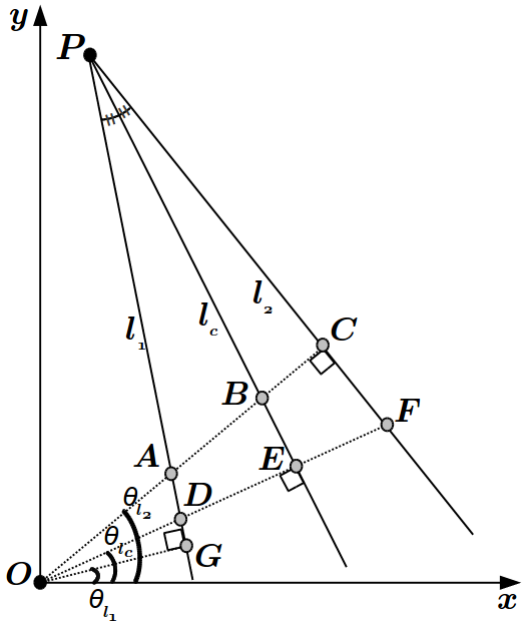


Figure D.1: Illustrating the separability conditions for the symmetry axis l_c of two lines l_1, l_2. See text for details.

Appendix E: Bilateral symmetry detector: supplementary information

E.1 Implementation Details

Our approach has a worst case run time given by the number of segments, G, and the number of final potential axes, $J \times L$, considered per segment. This results in a run time complexity of $O(G \times J \times L)$. J and L are usually small, with typical values ranging from 5 to 8, while G can range from 5 to 50 depending on the complexity/size of the image. Reducing/limiting G will therefore decrease the computational time per test image. There are two possible ways. The most direct method is to simply select the top G fixation points in $\mathcal{F}_{sym}$. However, this approach may miss weak symmetries depending on G, and it is hard to determine a reasonable value of G beforehand. Instead, we choose a slightly different approach based on the observation that many of the segments $r_m \in \mathcal{R}_{sym}$ are actually very similar, with large amounts of overlap between them. The implication is that searching over all G segments is likely to return similar results. To avoid this, we apply a simple filtering step that checks the region overlap between adjacent segments by computing the standard intersect over union scores: $SO_{r_m} = \frac{r_m \cap r_b}{r_m \cup r_b},\ \forall r_b \in \mathcal{R}_{sym}$. We then retain the T remaining segments with $SO_{r_m} > t_o$, where t_o is an overlap threshold that can be set to a reasonably (large) value (e.g. 0.75 to 0.9) or learned from training data.

This results in a reduced set of filtered segments $|\mathcal{R}_{sym}| = T \leq G$ and decreases the running time of the approach in practice. Typical values of T range from 5 to 10. As we have shown in the experimental results (sec. 4.6.2), the best performance is achieved when we add simple bounding box regions to $\mathcal{R}_{sym}$. For all the results reported here, we limit the addition to the top 5 fixation points, increasing the size of $\mathcal{R}_{sym}$ slightly: between 10 to 15. The current implementation runs over Matlab. The mean running times for a 320×240 image using a dual Intel Xeon 2.9GHz CPU with 128GB of ram is 8.79 ± 1.43s, where 2.52 ± 0.07s are used for generating the putative fixation points and 6.27 ± 1.36s for the refinement step.

Two key parameters are learned, for each dataset, via a separate offline training procedure: 1) $R(x_t)$, the optimal support for the symmetry axis within each segment r_m (sec. 4.3.2.2) and 2) t_o, the overlap threshold between segments in $\mathcal{R}_{sym}$ described earlier in this section. For each parameter, we search over a predefined range while holding the other parameter fixed at their default value. For $R(x_t)$, we search over 10 equally spaced discrete factors of the segment's width, X_{r_m}, ranging from $[0.1, 1.0]$ with default at 0.5. For t_o, we search between $[0.5, 0.95]$, with a default overlapping factor of 0.75. The optimal parameter values that yield the best overall performance in the training set of images are then used during evaluation over the test set. All parameter values used and their running times are listed in the sections that follow.

Method subset	Notation	Description
Edge Detection, gPb [6]	$t_{pb} = 0.07$	Edge detection threshold
Symmetry attention	$S = 4$	Number of Gabor scales searched
	$O = 16$	Number of orientations searched
Fixation-based segmentation	$\sigma = 3$	Edge contrast pairwise penalty standard deviation
	t_o	Segment overlap clustering threshold
Symmetry refinement	$\delta_e = 10$	Dilation factor per segment
	$t_\theta = 50\%$	Modal orientation selection threshold
	$\epsilon_\theta = \pi/180$	Orientation search resolution
	$R(x_t)$	Optimal centroid search width (factor of the segment width X_{r_m})
Symmetry axis scoring	$\delta_s = 20$	Dilation factor over edges of a segment, $I_e(r_m)$
	$\delta_h = 5$	Search size of scoring region in Hough space

Dataset	Type	Parameters $t_o, R(x_t)$
PSU 2011	singles	$0.75, 0.6$
	multiples	$0.75, 0.2$
PSU 2013	singles	$0.75, 0.6$
	multiples	$0.75, 0.8$
UMD Symmetry	singles	$0.9, 1.0$
	multiples	$0.7, 0.8$

Table E.1: (Left) Description of parameters. Those with values are the same over all datasets. (Right) Optimal parameters used per dataset.

E.2 Description of parameters and their values

E.2.1 Full approach `[AttentionSymSegBB]`

A brief description of the parameters used and their default values are summarized in Table E.1 (left). The values of the two learned optimal parameters: $t_o, R(x_t)$, used by the full approach `[AttentionSymSegBB]` are listed in Table E.1 (right) for each of the datasets considered. For $R(x_t)$, the value provided is a factor applied to X_{r_m}, the width of the segment. Note that other variants: `[SymAttention, RefinementOnly, AttentionSymBB]` and `[AttentionSymSeg]` are ablations of the full approach using the same parameters listed here as well.

Notation (default value, [search range])	Description
$t_s = 0.2, [0.1, 1.0]$	Scale ratio threshold for matching features of different scales
$t_a = 3, [1, 10]$	Angular threshold (degrees) for matching features with different orientations
$t_r = 3, [1, 10]$	Radial distance threshold (pixels) between matching features
$t_m = 1, [1, 10]$	Number of matches admitted per feature

Dataset	Type	Parameters t_s, t_a, t_r, t_m
PSU 2011	singles	$0.2, 3, 10, 1$
	multiples	$0.6, 8, 10, 3$
PSU 2013	singles	$0.2, 5, 3, 1$
	multiples	$0.4, 5, 10, 10$
UMD Symmetry	singles	$0.2, 3, 3, 9$
	multiples	$0.7, 8, 8, 2$

Table E.2: (Left) Description of parameters with their default values and parameter search ranges. (Right) Optimal parameters used per dataset.

E.2.2 Baseline [Loy-Eklundh]

We modify the original code[1] so that four key parameters: $\{t_s, t_a, t_r, t_m\}$ described in Table E.2 (left) are tuned from their default values via an offline parameter search procedure: 1) We search one parameter at a time, while holding the remaining three parameters fixed to their default values. 2) For each parameter, we search over ten discrete values listed in Table E.2 (left). The best parameter is selected that yields the highest Average Precision (AP) score over the training subset per dataset. 3) We then compute the AP scores when the best parameters obtained in step 2) are combined together and select the final parameter combination with the top AP scores. The final optimal parameters used per dataset are summarized in Table E.2 (right).

[1] http://www.nada.kth.se/~gareth/homepage/local_site/code.htm

E.3 Symmetry complexity coding in the UMD Symmetry dataset

We denote images via a systematic file naming procedure summarized in Table E.3 that encodes via 10 characters the attributes of the symmetry, including the *complexity* of the symmetry obtained from a set of 3 paid experts (taking the majority of their votes, with the authors as the tie-breaker).

POSITION	MNEMONIC	DESCRIPTION
1 (symmetry type)	2 or 3	2: planar 2D symmetry
		3: non-planar 3D symmetry
2 (symmetry number)	S or M	S: Single symmetry
		M: Multiple symmetries
3 (image type)	N or S	N: Natural image
		S: Synthetic image
4 (symmetry complexity)	P, Q, C or N	P: Perfect
		Q: Quasi (or approximate) symmetric
		C: Corrupted with clutter
		N: Not globally symmetric; but locally symmetric
5-10		Unique file number

Table E.3: Symmetry coding nomenclature.

E.4 Average Precision (AP) scores

Table E.4 lists the AP scores for the baseline `[Loy-Eklundh]` approach and all variants of the proposed approach explored in the experiments over the three datasets. For the symmetry complexity categories in the UMD symmetry dataset,

we report the AP scores of the baseline and the full approach `[AttentionSymSegBB]` in Table E.5.

Dataset	Type	[Loy-E]	[SymAtt]	[R-Only]	[AttSymLoyBB]	[AttSymBB]	[AttSymSeg]	[AttSymSegBB]
PSU 2011	singles	0.736	0.233	0.659	0.774	0.837	0.847	**0.889**
	multiples	0.525	0.183	-	0.576	0.673	0.493	**0.727**
PSU 2013	singles	0.741	0.209	0.631	0.832	0.738	**0.899**	0.890
	multiples	0.458	0.157	-	0.611	0.761	0.559	**0.795**
UMD Symmetry	singles	0.865	0.239	0.743	0.765	0.792	**0.898**	0.891
	multiples	0.321	0.219	-	0.569	0.471	0.432	**0.667**

Table E.4: Summary of AP scores comparing the baseline and all variants of the approach. Abbreviations used: `[Loy-E]`=`[Loy-Eklundh]`, `[R-Only`=`RefinementOnly`, `[Att*]`=`[Attention*]`. '-' indicates no experiments for this variant was performed. Best performance per dataset (row) is highlighted in bold.

E.5 Running times per dataset

We summarize the run time performance of the full approach `[AttentionSymSegBB]` and the baseline `[Loy-Eklundh]` by the mean, max, min and standard deviation of running times per test image over all three datasets in Table E.6 and Table E.7, respectively. For `[AttentionSymSegBB]`, we provide separate timings for the symmetry attention and refinement stages separately in addition to the total processing time. For the refinement stage, we used a parallelized implementation where the symmetry axes for different segments in $\mathcal{R}_{sym}$ are computed in parallel. We used the optimal parameters described in sec. E.2 of each approach. For fairness, we timed the results using the same hardware setup: Dual Intel Xeon 2.9GHz CPU + 128GB ram and repeated the timings three times per approach and took the fastest results.

Dataset Type	Complexity	[Loy-Eklundh]	[AttentionSymSegBB]
UMD Symmetry – singles	P	**0.862**	0.893
	Q	0.802	0.890
	C	0.718	0.905
	N	0.860	**0.967**
UMD Symmetry – multiples	Q	**0.439**	0.711
	C	0.268	**0.820**
	N	0.287	0.540

Table E.5: Summary of AP scores comparing the baseline and the full approach over different symmetry complexity categories in the UMD Symmetry dataset. Best performance for each approach (column) per dataset type is highlighted in bold.

Finally, all input images are resized to 320×240 and we exclude all I/O processing (reading in data, resizing and displaying) in the timings.

Dataset	Type	Runtimes (seconds): mean, max, min, std-dev		
		Sym Attention	Sym Refinement	Total
PSU 2011	singles	2.52, 2.79, 2.37, 0.06	5.84, 10.34, 3.18, 1.40	8.36, 13.13, 5.55, 1.47
	multiples	2.51, 3.15, 2.38, 0.09	6.53, 10.00, 3.34, 1.36	9.04, 13.15, 5.73, 1.45
PSU 2013	singles	2.53, 2.78, 2.43, 0.06	6.21, 9.43, 3.46, 1.31	8.75, 12.21, 5.89, 1.37
	multiples	2.53, 2.84, 2.39, 0.08	6.76, 10.43, 4.33, 1.40	9.29, 13.27, 6.72, 1.47
UMD Symmetry	singles	2.50, 2.85, 2.34, 0.06	6.13, 11.12, 3.34, 1.38	8.63, 13.98, 5.68, 1.44
	multiples	2.53, 2.86, 2.37, 0.07	6.15, 10.43, 3.50, 1.33	8.68, 13.29, 5.88, 1.40
Overall running times		2.52, 2.88, 2.38, 0.07	6.27, 10.29, 3.53, 1.36	8.79, 13.17, 5.91, 1.43

Table E.6: Running times for [AttentionSymSegBB] (full approach).

Dataset	Type	Runtimes (seconds): mean, max, min, std-dev
PSU 2011	singles	1.13, 3.94, 0.42, 0.69
	multiples	1.25, 7.07, 0.46, 0.91
PSU 2013	singles	1.19, 3.64, 0.45, 0.80
	multiples	1.27, 4.02, 0.52, 0.81
UMD Symmetry	singles	0.83, 2.47, 0.41, 0.45
	multiples	1.04, 3.55, 0.44, 0.57
Overall running times		1.12, 4.11, 0.45, 0.70

Table E.7: Running times for `[Loy-Eklundh]` (baseline).

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 181

[2] A. Aldoma, F. Tombari, and M. Vincze. Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1732–1739, 2012. 174, 175

[3] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012. 1, 20

[4] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, 32(1):19–34, 2013. 12

[5] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, 2013. 194, 209, 210

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 17, 37, 38, 65, 119, 137, 157, 158, 222

[7] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2294–2301, 2009. 154, 155

[8] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 328–335. IEEE, 2014. 2, 8

[9] F. Barranco, C. L. Teo, C. Fermüller, and Y. Aloimonos. Contour detection and characterization for asynchronous event sensors. In *Proc. Int'l Conf. on Computer Vision*, Accepted, 2015. 44, 46

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, Apr. 2002. 57, 74, 83

[11] G. Ben-Yosef and O. Ben-Shahar. A tangent bundle theory for visual curve completion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(7):1263–1280, 2012. 7

[12] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi. Event-based visual flow. *IEEE Trans. on Neural Networks and Learning Systems*, 25(2):407–417, 2014. 47

[13] T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth. Who's in the picture? In *Advances in Neural Information Processing Systems*, 2004. 12

[14] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *Proc. European Conference on Computer Vision*, pages 663–676. Springer, 2010. 196

[15] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4380–4389, June 2015. 65, 170

[16] G. Bertasius, J. Shi, and L. Torresani. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. *arXiv preprint arXiv:1504.06201*, 2015. 65

[17] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 195

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003. 208

[19] H. Blum. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT press Cambridge, 1967. 113

[20] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1729–1736. IEEE, 2011. 66, 67

[21] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. *Int'l Symp. on Experimental Robotics*, 2012. 67, 104, 105, 174, 181

[22] J. Bohg and D. Kragic. Grasping familiar objects using shape context. *Int. Conf. on Advanced Robotics*, pages 1–6, 2009. 173

[23] A. Bosch, A. Zisserman, and X. Muoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(4):712–727, 2008. 9

[24] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proc. European Conf. on Computer Vision*, pages 168–181. Springer, 2010. 65

[25] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2559–2566. IEEE, 2010. 4

[26] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004. 9

[27] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 15, 119

[28] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. *Proc. Int'l Conf. on Computer Vision*, 1:105–112, 2001. 125, 139

[29] S. Branavan, L. S. Zettlemoyer, and R. Barzilay. Reading between the lines: Learning to map high-level instructions to commands. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1268–1277. Association for Computational Linguistics, 2010. 192

[30] A. S. Bregman. Asking the 'what for' question in auditory perception. *Perceptual organization*, pages 99–118, 1981. 5, 6

[31] X. Bresson, P. Vandergheynst, and J. Thiran. A priori information in image segmentation: energy functional based on shape statistical model and image information. In *Proc. Int'l Conf. on Image Processing*, volume 3, pages III–425. IEEE, 2003. 7

[32] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, 31(4):532–540, 1983. 8

[33] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *Proc. Int'l Conf. on Computer Vision*, pages 1–8. IEEE, 2007. 10

[34] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3286–3293, 2014. 17, 20

[35] M. Chertok and Y. Keller. Spectral symmetry analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(7):1227–1238, 2010. 114

[36] J. D. Choi and M. Palmer. Robust constituent-to-dependency conversion for english. In *Proceedings of the 9th International Workshop on Treebanks and Linguistic Theories*, pages 55–66, Tartu, Estonia, 2010. 202

[37] P. M. Claessens and J. Wagemans. A bayesian framework for cue integration in multistable grouping: Proximity, collinearity, and orientation priors in zigzag lattices. *Journal of Vision*, 8(7):33, 2008. 6

[38] J. Clarke, D. Goldwasser, M.-W. Chang, and D. Roth. Driving semantic parsing from the world's response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 18–27. Association for Computational Linguistics, 2010. 192

[39] R. W. Conners and C. T. Ng. Developing a quantitative model of human preattentive vision. *IEEE Trans. on Systems, Man and Cybernetics*, 19(6):1384–1407, 1989. 113

[40] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 7

[41] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 65

[42] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1124–1131 vol. 2, June 2005. 8

[43] E. Craft, H. Schütze, E. Niebur, and R. von der Heydt. A neural model of figure-ground organization. *J. Neurophysiology*, 97(6):4310–4326, 2007. 3, 6, 17, 42

[44] D. Cremers, F. R. Schmidt, and F. Barthel. Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2008. 8

[45] A. Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, February 2013. 28

[46] F. C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Computer Graphics*, 18(3):207–212, 1984. 73, 122

[47] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 886–893, 2005. 22, 176

[48] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proc. European Conf. on Computer Vision*, pages 71–84. Springer, 2010. 171

[49] M. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976. 171, 176

[50] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014. 22

[51] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2015. 21, 28, 30, 37, 38, 60, 135, 137, 179

[52] N. Dorfman, D. Harari, and S. Ullman. Learning to perceive coherent objects. In *CogSci*, pages 394–399, 2013. 22, 24

[53] J. Driver and G. C. Baylis. Preserved figure-ground segregation and symmetry perception in visual neglect. *Nature*, 360(6399):73–75, 1992. 112

[54] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3481. IEEE, 2012. 197

[55] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu. A mid-level representation framework for semantic sports video analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 33–44. ACM, 2003. 4

[56] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. European Conference on Computer Vision*, volume 2353, pages 97–112. Springer, 2002. 11

[57] T. Egner, J. M. Monti, E. H. Trittschuh, C. A. Wieneke, J. Hirsch, and M.-M. Mesulam. Neural integration of top-down spatial and feature-based information in visual search. *The Journal of neuroscience*, 28(24):6141–6151, 2008. 3

[58] M. Eimer, M. Kiss, and S. Nicholas. What top-down task sets do for us: An erp study on the benefits of advance preparation in visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 37(6):1758, 2011. 3

[59] J. H. Elder, A. Krupnik, and L. A. Johnston. Contour grouping with prior models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):661–674, 2003. 7

[60] J. H. Elder and S. W. Zucker. Computing contour closure. In *Proc. European Conf. on Computer Vision*, pages 399–412. Springer, 1996. 7

[61] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(2):222–234, 2014. 20

[62] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int'l J. of Computer Vision*, 88(2):303–338, 2010. 2

[63] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2352–2359. IEEE, 2010. 196

[64] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009. 196

[65] A. Farhadi, S. M. M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. In *Proc. European Conference on Computer Vision*, pages 15–29. Springer, 2010. 13

[66] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 4

[67] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. 157

[68] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 524–531. IEEE, 2005. 9

[69] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 195

[70] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 195

[71] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303, 2010. 102

[72] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *Proc. European Conf. on Computer Vision*, pages 14–28. Springer, 2006. 64

[73] V. Ferrari and A. Zisserman. Learning visual attributes. In *Advances in Neural Information Processing Systems*, 2007. 174

[74] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 195

[75] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: Evidence for a local "association field". *Vision research*, 33(2):173–193, 1993. 6

[76] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013. 209

[77] H. Fu, X. Cao, Z. Tu, and D. Lin. Symmetry constraint for foreground extraction. *IEEE Trans. on Systems, Man and Cybernetics*, 44(5):644–654, 2014. 116, 120

[78] J. L. Gallant, C. E. Connor, S. Rakshit, J. W. Lewis, and D. C. Van Essen. Neural responses to polar, hyperbolic, and cartesian gratings in area v4 of the macaque monkey. *J. Neurophysiology*, 76(4):2718–2739, 1996. 26

[79] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2, 1986. 7

[80] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006. 30

[81] T. Ghose and S. E. Palmer. Extremal edges versus other principles of figure-ground organization. *Journal of Vision*, 10(8):3, 2010. 22, 24

[82] J. J. Gibson. *The theory of affordances*. Hilldale, USA, 1977. 15

[83] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. 170, 189

[84] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012. 11

[85] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2013. 194, 208, 209

[86] G. González, F. Aguet, F. Fleuret, M. Unser, and P. Fua. Steerable features for statistical 3d dendrite detection. In *Proc. Int'l Conf. on Medical Image Computing and Computer Assisted Intervention*, pages 625–632. 2009. 115, 117

[87] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 68

[88] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1529–1536, 2011. 173

[89] D. Graff. English gigaword. In *Linguistic Data Consortium, Philadelphia, PA*, 2003. 202

[90] A. S. Greenberg, M. Esterman, D. Wilson, J. T. Serences, and S. Yantis. Control of spatial and feature-based attention in frontoparietal cortex. *The Journal of Neuroscience*, 30(43):14330–14339, 2010. 3

[91] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proc. European Conference on Computer Vision*, pages 16–29. Springer, 2008. 11

[92] A. Gupta, S. Satkin, A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1961–1968, June 2011. 11

[93] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 564–571, 2013. 2, 9, 17, 33

[94] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. 2013. 66

[95] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004. 194, 203, 206

[96] J. S. Hare, P. H. Lewis, P. G. Enser, and C. J. Sandom. Mind the gap: another look at the problem of the semantic gap in image retrieval. In *Electronic Imaging 2006*, pages 607309–607309. International Society for Optics and Photonics, 2006. 3

[97] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Proc. Int'l Conf. on Computer Vision*, pages 991–998, Nov 2011. 65

[98] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. European Conf. on Computer Vision*, pages 297–312, 2014. 174

[99] G. Hatfield and W. Epstein. The status of the minimum principle in the theoretical analysis of visual perception. *Psychological Bulletin*, 97(2):155, 1985. 5

[100] X. He, R. S. Zemel, and M. Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages II–695. IEEE. 8

[101] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2807–2814. IEEE, 2012. 11

[102] F. Heitger, L. Rosenthaler, R. Von Der Heydt, E. Peterhans, and O. Kübler. Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision research*, 32(5):963–981, 1992. 6, 45

[103] F. Heitger, R. von der Heydt, E. Peterhans, L. Rosenthaler, and O. Kübler. Simulation of neural contour mechanisms: representing anomalous contours. *Image and Vision Computing*, 16(6):407–421, 1998. 5

[104] R. D. Henkel. Segmentation in scale space. In *Computer Analysis of Images and Patterns*, pages 41–48. Springer, 1995. 8

[105] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick. Learning contact locations for pushing and orienting unknown objects. In *Proc. IEEE Int'l Conf. on Humanoid Robots*, 2013. 174

[106] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, 2012. 65, 95, 96, 97

[107] T. K. Ho. Random decision forests. In *Proc. IEEE Int'l Conf. on Document Analysis and Recognition*, volume 1, pages 278–282, 1995. 28

[108] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999. 208

[109] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *Int'l J. of Computer Vision*, 91(3):328–346, 2011. 1, 20, 21

[110] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. 194, 205

[111] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012. 65, 88, 95, 96, 97

[112] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 171

[113] Q. Huang, B. Dam, D. Steele, J. Ashley, and W. Niblack. Foreground/background segmentation of color images by integration of multiple cues. In *Proc. Int'l Conf. on Image Processing*, volume 1, pages 246–249 vol.1, Oct 1995. 2

[114] X. Huang and L. Zhang. Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. *Int'l J. Remote Sensing*, 30(8):1977–1987, 2009. 115

[115] P. S. Huggins, H. F. Chen, P. N. Belhumeur, and S. W. Zucker. Finding folds: On the appearance and identification of occlusion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 718–725, 2001. 24

[116] H. Izadinia, I. Saleemi, and M. Shah. Multimodal analysis for identification and segmentation of moving-sounding objects. *IEEE transactions on multimedia*, 15(2):378–390, 2013. 194

[117] J. F. Jehee, V. A. Lamme, and P. R. Roelfsema. Boundary assignment in a recurrent network architecture. *Vision research*, 47(9):1153–1165, 2007. 6

[118] H. Jiang and S. Yu. Linear solution to scale and rotation invariant object matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2474–2481. IEEE, 2009. 66

[119] L. Jie, B. Caputo, and V. Ferrari. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *Advances in Neural Information Processing Systems*. NIPS, December 2009. 12

[120] G. Kanizsa. Subjective contours. *Scientific American*, 234(4):48–52, 1976. 5

[121] G. Kanizsa and W. Gerbino. Convexity and symmetry in figure-ground organization. *Vision and artifact*, pages 25–32, 1976. 18

[122] A. Karpathy and L. Fei-Fei. Real time detection and segmentation of reflectionally symmetric objects in digital images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2015. 13, 170

[123] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems*, pages 1889–1897. 2014. 209, 210

[124] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l J. of Computer Vision*, 1(4):321–331, 1988. 7

[125] R. J. Kate and R. J. Mooney. Learning language semantics from ambiguous supervision. In *Proc. National Conference on Artificial Intelligence*, volume 7, pages 895–900, 2007. 192

[126] C. C. Kemp and A. Edsinger. Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *Proc. Intl. Conf. on Development and Learning*, 2006. 173

[127] R. Kennedy, J. Gallier, and J. Shi. Contour cut: identifying salient contours in images by solving a hermitian eigenvalue problem. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2065–2072. IEEE, 2011. 61

[128] J. Kim and R. J. Mooney. Unsupervised pcfg induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 433–444. Association for Computational Linguistics, 2012. 193

[129] T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 194

[130] R. Kimchi, M. Behrmann, and C. R. Olson. *Perceptual organization in vision: Behavioral and neural perspectives*. Psychology Press, 2003. 5

[131] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the International Conference on Machine Learning*, pages 595–603. JMLR Workshop and Conference Proceedings, 2014. 209, 210

[132] H. Kjellström, J. Romero, and D. Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90, 2011. 173

[133] J. J. Koenderink and A. J. Van Doorn. The singularities of the visual mapping. *Biological cybernetics*, 24(1):51–59, 1976. 19

[134] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8):557–564, 1992. 171, 177

[135] K. Koffka. *Principles of Gestalt Psychology*. Harcourt, Brace, 1935. 4, 170

[136] N. Kogo, C. Strecha, L. Van Gool, and J. Wagemans. Surface construction by a 2-d differentiation–integration process: A neurocomputational model for perceived border ownership, depth, and lightness in kanizsa figures. *Psychological review*, 117(2):406, 2010. 6

[137] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009. 8

[138] I. Kokkinos, R. Deriche, P. Maragos, and O. Faugeras. A biologically motivated and computationally tractable model of low and mid-level vision tasks. In *Proc. European Conf. on Computer Vision*, pages 506–517. Springer, 2004. 4

[139] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. 139, 142

[140] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *Proc. Int'l Conf. on Computer Vision*, pages 2190–2197, 2011. 14, 28, 111, 173, 174

[141] P. Kontschieder, H. Riemenschneider, M. Donoser, and H. Bischof. Discriminative learning of contour fragments for object detection. In *Proceedings of the British Machine Vision Conference*, pages 4.1–4.12. BMVA Press, 2011. 64

[142] G. Kootstra, N. Bergstrom, and D. Kragic. Using symmetry to select fixation points for segmentation. *Proc. Int'l Conf. on Pattern Recognition*, pages 3894–3897, 2010. 114, 116, 120

[143] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *Int'l J. of Robotics Research*, 32(8):951–970, 2013. 174

[144] P. Kovesi. Symmetry and asymmetry from local phase. In *Tenth Australian Joint Conf. on Artificial Intelligence*, volume 190, 1997. 115

[145] J. Krishnamurthy and T. Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. In *Transactions of the Association for Computational Linguistics*, 2013. 193

[146] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(8):1847–1871, 2013. 14, 17, 45

[147] M. Kubovy and J. Wagemans. Grouping by proximity and multistability in dot lattices: A quantitative gestalt theory. *Psychological Science*, 6(4):225–234, 1995. 6

[148] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In

Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 1601–1608. IEEE, 2011. 12

[149] M. Kumar, P. Ton, and A. Zisserman. Obj cut. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 18–25 vol. 1, June 2005. 10

[150] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *Proc. European Conf. on Computer Vision*, pages 239–253. Springer, 2010. 10

[151] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. 7

[152] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 174, 195

[153] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000. 8

[154] L. J. Latecki, C. Lu, M. Sobel, and X. Bai. Multiscale random fields with application to contour grouping. In *Advances in Neural Information Processing Systems*, pages 913–920. 2009. 8

[155] M. W. Law and A. C. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *Proc. European Conf. on Computer Vision*, pages 368–382. 2008. 115, 117

[156] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 209

[157] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *International Conference on Artificial Intelligence and Statistics,AISTATS*, 2015. 210

[158] S. Lee and Y. Liu. Curved glide-reflection symmetry detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(2):266–278, Feb 2012. 114

[159] T. S. H. Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *Proc. Int'l Conf. on Computer Vision*, pages 1753–1760, 2013. 112, 116, 118, 119, 155, 164, 165, 166, 167

[160] Y. J. Lee and K. Grauman. Object-graphs for context-aware visual category discovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 12

[161] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 2, page 7, 2004. 64, 67

[162] I. Leichter and M. Lindenbaum. Boundary ownership by lifting to 2.1 d. In *Proc. Int'l Conf. on Computer Vision*, pages 9–16, 2009. 17, 18, 20, 21, 22, 32, 33, 36, 50

[163] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *Int'l J. of Robotics Research*, 2014. 173, 175, 180, 181, 183, 185

[164] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 64

[165] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. European Conf. on Computer Vision*, pages 581–594. Springer, 2006. 10

[166] A. Levinshtein, S. Dickinson, and C. Sminchisescu. Multiscale symmetric part detection and grouping. *Proc. Int'l Conf. on Computer Vision*, pages 2162–2169, 2009. 116, 119

[167] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Optimal Contour Closure. *International Journal of Computer Vision*, 2012. 7

[168] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2036–2043. IEEE, 2009. 10

[169] W. H. Li and L. Kleeman. Real time object tracking using reflectional symmetry and motion. *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 2798–2803, 2006. 114

[170] W. H. Li, A. M. Zhang, and L. Kleeman. Real time detection and segmentation of reflectionally symmetric objects in digital images. *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 4867–4873, 2006. 116

[171] W. Lian and L. Zhang. Rotation invariant non-rigid shape matching in cluttered scenes. In *Proc. European Conference on Computer Vision*, pages 506–518. Springer, 2010. 66

[172] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446, 2013. 192

[173] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 times; 128 120 db 15 Îijs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, Feb 2008. 14, 39, 44

[174] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3158–3165, 2013. 24, 60

[175] J. Lin. Divergence measures based on the shannon entropy. *IEEE Trans. on Information Theory*, 37(1):145–151, 1991. 128

[176] T. Lindeberg. *Scale-space theory in computer vision*. Springer Science & Business Media, 1993. 8

[177] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *Int'l J. of Computer Vision*, 30(2):117–156, 1998. 113

[178] H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007. 118, 128, 134

[179] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1996–2003, June 2009. 171

[180] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu. Symmetry detection from real world images competition 2013: Summary and results. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 200–205, June 2013. 144, 145

[181] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. Van Gool. *Computational symmetry in computer vision and computer graphics*, volume 5. 2009. 113

[182] P. Locher and C. Nodine. The perceptual value of symmetry. *Computers & mathematics with applications*, 17(4):475–484, 1989. 113

[183] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision*, 60(2):91–110, 2004. 114

[184] G. Loy and J.-O. Eklundh. Detecting symmetry and symmetric constellations of features. *Proc. European Conf. on Computer Vision*, pages 508–521, 2006. 111, 114, 118, 126, 131, 132, 144, 148, 160, 162

[185] C. Lu, L. J. Latecki, N. Adluru, X. Yang, and H. Ling. Shape guided contour grouping with particle filters. In *Proc. International Conference on Computer Vision*, pages 2288–2295. IEEE, 2009. 65

[186] Y. Lu, L. Zhang, Q. Tian, and W.-Y. Ma. What are the high-level concepts with small semantic gaps? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3

[187] T. Ma and L. J. Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1441–1448. IEEE, 2011. 63, 86, 99, 100, 102

[188] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 64

[189] S. Maji and A. C. Berg. Max-margin additive classifiers for detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 40–47. IEEE, 2009. 208

[190] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1045. IEEE, 2009. 65, 99, 100, 102

[191] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 248–255, 2014. 182

[192] D. Marr. Early processing of visual information. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 275(942):483–519, 1976. 170, 189

[193] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004. 2, 17, 18, 20, 32, 50, 60, 89, 134, 154

[194] A. Martinez, L. Anllo-Vento, M. I. Sereno, L. R. Frank, R. B. Buxton, D. Dubowitz, E. C. Wong, H. Hinrichs, H. J. Heinze, and S. A. Hillyard. Involvement of striate and extrastriate visual cortical areas in spatial attention. *Nature neuroscience*, 2(4):364–369, 1999. 3

[195] C. Matuszek, N. Fitzgerald, L. Zettlemoyer, L. Bo, and D. Fox. A joint model of language and perception for grounded attribute learning. In *Proceedings of the International Conference on Machine Learning*, pages 1671–1678, 2012. 193

[196] C. Matuszek, D. Fox, and K. Koscher. Following directions using statistical machine translation. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, pages 251–258. IEEE Press, 2010. 193

[197] H. Meng, D. R. Hardoon, J. Shawe-Taylor, and S. Szedmak. Generic object recognition by combining distinct features in machine learning. In *Electronic Imaging 2005*, pages 90–98. International Society for Optics and Photonics, 2005. 194

[198] Y. Ming, H. Li, and X. He. Connected contours: A new contour completion model that respects the closure effect. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 829–836. IEEE, 2012. 61

[199] A. Mishra, Y. Aloimonos, and C. Fermüller. Active segmentation for robotics. *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 3133–3139, 2009. 40, 42, 116, 120, 125

[200] R. Mohan and R. Nevatia. Perceptual organization for scene segmentation and description. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(6):616–635, 1992. 6

[201] R. J. Mooney and R. C. Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2005. 208

[202] V. Movahedi and J. Elder. Combining local and global cues for closed contour extraction. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013. 7

[203] K. Murphy, A. Torralba, W. Freeman, et al. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in neural information processing systems*, 16:1499–1506, 2003. 9

[204] A. Myers, A. Kanazawa, C. Fermüller, and Y. Aloimonos. Affordance of object parts from geometric features. In *Proc. of Robotics: Science and Systems RGB-D Workshop*, 2014. 173, 180, 181, 186

[205] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2015. 171

[206] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. European Conf. on Computer Vision*, pages 746–760, 2012. 17, 18, 32

[207] D. Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive psychology*, 9(3):353–383, 1977. 5

[208] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the International Conference on Machine Learning*, pages 689–696, 2011. 209

[209] M. Nishigaki, C. Fermüller, and D. DeMenthon. The image torque operator: A new tool for mid-level vision. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 502–509, 2012. 4, 14, 20, 26, 39, 41, 57, 70, 73, 123

[210] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int'l J. of Computer Vision*, 42(3):145–175, 2001. 9

[211] B. Ommer and J. Malik. Multi-scale object detection by clustering lines. In *Proc. International Conference on Computer Vision*, pages 484–491. IEEE, 2009. 65

[212] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proc. European Conference on Computer Vision*, pages 575–588. Springer, 2006. 64

[213] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. 7

[214] D. Osorio. Symmetry detection by categorization of spatial phase, a model. *Proc. of the Royal Society of London. Series B: Biological Sciences*, 263(1366):105–110, 1996. 115

[215] S. E. Palmer. Hierarchical structure in perceptual representation. *Cognitive psychology*, 9(4):441–474, 1977. 5

[216] S. E. Palmer. *Vision Science: Photons to phenomenology*, volume 1. MIT press Cambridge, MA, 1999. 22

[217] S. E. Palmer and J. L. Brooks. Edge-region grouping in figure-ground organization and depth perception. *J. Exp Psychol: Hum Percep Perform*, 34(6):1353–1371, 2008. 25

[218] S. E. Palmer and T. Ghose. Extremal edge: A powerful cue to depth perception and figure-ground organization. *Psychological Science*, 19(1):77–83, 2008. 18, 22

[219] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):823–839, 1989. 6

[220] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1681–1688. IEEE, 2011. 196

[221] D. Parikh and K. Grauman. Relative attributes. *Proc. Int'l Conf. on Computer Vision*, pages 503–510, 2011. 174, 196

[222] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2012. 9, 196, 199

[223] E. Peterhans and R. von der Heydt. Mechanisms of contour perception in monkey visual cortex. ii. contours bridging gaps. *The Journal of neuroscience*, 9(5):1749–1763, 1989. 5

[224] S. Petrov, L. Barrett, R. Thibaux, and D. Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL/COLING)*, pages 433–440, 2006. 202

[225] J. R. Pomerantz and M. Kubovy. Theoretical approaches to perceptual organization: Simplicity and likelihood principles. *Organization*, 36:3, 1986. 5

[226] F. T. Qiu, T. Sugihara, and R. von der Heydt. Figure-ground mechanisms provide structure for selective attention. *Nature neuroscience*, 10(11):1492–1499, 2007. 6

[227] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009. 103

[228] S. Ramenahalli, S. Mihalas, and E. Niebur. Extremal edges: Evidence in natural images. In *Conf. on Information Sciences and Systems*, pages 1–5, 2011. 19, 24, 35

[229] S. Ravishankar, A. Jain, and A. Mittal. Multi-stage contour based detection of deformable objects. In *Proc. European Conference on Computer Vision*, pages 483–496. Springer, 2008. 64

[230] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: the generalized symmetry transform. *Int'l J. of Computer Vision*, 14(2):119–130, 1995. 114, 126

[231] X. Ren, C. C. Fowlkes, and J. Malik. Figure/ground assignment in natural images. In *Proc. European Conf. on Computer Vision*, pages 614–627. 2006. 17, 18, 20, 21, 22, 24, 32, 33, 36

[232] W. Richards. Marr, gibson, and gestalt: a challenge. *Perception-London*, 41(9):1024, 2012. 170

[233] W. Richards and D. D. Hoffman. Codon constraints on closed 2d shapes. *Computer Vision, Graphics, and Image Processing*, 31(3):265–281, 1985. 68

[234] H. Riemenschneider, M. Donoser, and H. Bischof. Using partial edge contour matches for efficient object category localization. In *Proc. European Conference on Computer Vision*, pages 29–42. Springer, 2010. 63, 64, 86, 102

[235] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999. 195

[236] T. Riklin-Raviv, N. Kiryati, and N. Sochen. Segmentation by level sets and symmetry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1015–1022, 2006. 116, 119

[237] L. Roberts. *Machine perception of 3-d solids*. PhD thesis, MIT, 1965. 20

[238] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 23(3):309–314, 2004. 17, 139

[239] M. Rousson and N. Paragios. Shape priors for level set representations. In *Proc. European Conf. on Computer Vision*, pages 78–92. Springer, 2002. 8

[240] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *Trends and Topics in Computer Vision*, pages 1–14. Springer, 2012. 196

[241] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1848–1853, 2009. 66

[242] E. Sali and S. Ullman. Combining class-specific fragments for object classification. In *Proceedings of the British Machine Conference, pages*, pages 21–1. 10

[243] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *Int'l J. of Robotics Research*, 27(2):157–173, 2008. 173, 183

[244] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 20

[245] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 134

[246] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, July 2008. 8, 64

[247] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):300–312, 2007. 9

[248] C. Silberer, V. Ferrari, and M. Lapata. *Models of Semantic Representation with Visual Attributes*, pages 572–582. Association for Computational Linguistics, 2013. 196, 210

[249] A. Sironi, V. Lepetit, and P. Fua. Multiscale centerline detection by learning a scale-space distance transform. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2697–2704, 2014. 112, 113, 115, 117, 118, 155, 156, 157, 164, 165

[250] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, pages 935–943. 2013. 209

[251] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *Advances in Neural Information Processing Systems*, 2012. 174

[252] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. 209, 210

[253] R. Socher, C. C. Lin, A. Ng, and C. Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 129–136, 2011. 13

[254] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1673–1680. IEEE, 2010. 64, 99, 100, 102

[255] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012. 209

[256] J. S. Stahl and S. Wang. Edge grouping combining boundary and region information. *IEEE Trans. on Image Processing*, 16(10):2590–2606, 2007. 7

[257] J. S. Stahl and S. Wang. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(3):395–411, 2008. 6, 7

[258] L. Stark and K. Bowyer. Function-based generic recognition for multiple object categories. *CVGIP: Image Understanding*, 59(1):1–21, 1994. 173

[259] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele. Functional object class detection based on learned affordance cues. In *Computer Vision Systems*, pages 435–444. Springer, 2008. 173

[260] M. Steedman. *The syntactic process*. The MIT press, 2001. 192

[261] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *Int'l J. of Computer Vision*, 82(3):325–357, 2009. 20, 21

[262] T. Sugihara, F. T. Qiu, and R. von der Heydt. The speed of context integration in the visual cortex. *Journal of neurophysiology*, 106(1):374–385, 2011. 6, 16

[263] C. Sun and D. Si. Fast reflectional symmetry detection using orientation histograms. *Real-Time Imaging*, 5(1):63–74, 1999. 126

[264] Y. Sun and B. Bhanu. Reflection symmetry-integrated image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(9):1827–1841, 2012. 116, 119, 157, 158, 159

[265] Y. Sun, L. Bo, and D. Fox. Attribute based object identification. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 2096–2103, 2013. 174, 175

[266] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Proc. Asian Conf. on Computer Vision*, pages 525–538. Springer, 2013. 66

[267] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proc. National Conference on Artificial Intelligence*, 2011. 193

[268] C. L. Teo, C. Fermüller, and Y. Aloimonos. Fast 2D border ownership assignment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 5117–5125, June 2015. 16, 45

[269] C. L. Teo, C. Fermüller, and Y. Aloimonos. A gestaltist approach to contour-based object recognition: Combining bottom-up and top-down cues. *The International Journal of Robotics Research*, 34(4-5):627–652, 2015. 57, 123

[270] C. L. Teo, C. Fermüller, and Y. Aloimonos. Detection and segmentation of 2D curved reflection symmetric structures. In *Proc. Int'l Conf. on Computer Vision*, Accepted, 2015. 112

[271] C. L. Teo, A. Myers, C. Fermüller, and Y. Aloimonos. Embedding high-level information into low level vision: Efficient object search in clutter. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 126–132. IEEE, 2013. 57, 66, 103, 104, 105

[272] C. L. Teo, H. Yi, and C. Fermüller. Object-centric bilateral symmetry detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, submitted. 111

[273] A. Thayananthan, B. Stenger, P. H. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–127. IEEE, 2003. 83

[274] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010. 135

[275] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *Proc. European Conference on Computer Vision*, pages 776–789. Springer, 2010. 196

[276] A. Toshev, B. Taskar, and K. Daniilidis. Object detection via boundary structure segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 950–957. IEEE, 2010. 64, 67

[277] A. Tsai, A. Yezzi Jr, W. Wells III, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. Model-based curve evolution technique for image segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages I–463. IEEE, 2001. 7

[278] S. Tsogkas and I. Kokkinos. Learning-based symmetry detection in natural images. *Proc. European Conf. on Computer Vision*, pages 41–54, 2012. 112, 114, 118, 134, 154, 155, 165

[279] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *Int'l J. of Computer Vision*, 63(2):113–140, 2005. 12

[280] C. W. Tyler. *Human symmetry perception and its computational analysis.* Lawrence Erlbaum Associates Publishers, 2002. 113

[281] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012. 208

[282] O. Veksler. Star shape prior for graph-cut image segmentation. In *Proc. European Conf. on Computer Vision*, pages 454–467. Springer, 2008. 141, 142

[283] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int'l J. of Computer Vision*, 50(3):271–293, 2002. 17

[284] R. Von der Heydt, E. Peterhans, and G. Baumgartner. Illusory contours and cortical neuron responses. *Science*, 224(4654):1260–1262, 1984. 5, 45

[285] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin*, 138(6):1172, 2012. 4

[286] C. Wang, Y. Li, W. Ito, K. Shimura, and K. Abe. A machine learning approach to extract spinal column centerline from three-dimensional ct data. In *SPIE Medical Imaging*, pages 72594T–72594T, 2009. 115

[287] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. European Conf. on Computer Vision*, pages 591–604. Springer-Verlag, 2010. 171

[288] X. Wang, X. Bai, T. Ma, W. Liu, and L. J. Latecki. Fan shape model for object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–158. IEEE, 2012. 64, 99, 100, 102

[289] X.-J. Wang, L. Zhang, M. Liu, Y. Li, and W.-Y. Ma. Arista-image search to annotation on billions of web photos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2987–2994. IEEE, 2010. 3

[290] Y. Wei, E. Brunskill, T. Kollar, and N. Roy. Where to go: Interpreting natural directions using global inference. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 3761–3767. IEEE, 2009. 192

[291] L. R. Williams and D. W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997. 6, 7

[292] J. Wu and J. M. Rehg. Centrist: A visual descriptor for scene categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501, 2011. 9

[293] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. 171

[294] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3485–3492, June 2010. 1

[295] J. Xiao, J. Hays, B. C. Russell, G. Patterson, K. A. Ehinger, A. Torralba, and A. Oliva. Basic level scene understanding: categories, attributes and structures. *Frontiers in psychology*, 4, 2013. 10

[296] Y. Xu, Y. Quan, Z. Zhang, H. Ji, C. Fermüller, M. Nishigaki, and D. Dementhon. Contour-based recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3402–3409, 2012. 20

[297] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1794–1801. IEEE, 2009. 9

[298] S. Yang and Y. Wang. Rotation invariant shape contexts based on feature-space fourier transformation. In *International Conference on Image and Graphics*, pages 575–579. IEEE, 2007. 66

[299] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 444–454. Association for Computational Linguistics, 2011. 202

[300] B. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485 –1508, aug. 2010. 12

[301] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 702–709. IEEE, 2012. 2, 10

[302] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 771–778. IEEE, 2013. 197

[303] S. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *IEEE Computer Vision and Pattern Recognition Workshops*, pages 1–7. IEEE, 2008. 11

[304] X. Yu and Y. Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *Proc. European Conf. on Computer Vision*, pages 127–140, 2010. 174

[305] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(12):1154–1166, 1995. 115

[306] N. R. Zhang and R. von der Heydt. Analysis of the context integration mechanisms underlying figure–ground organization in the visual cortex. *The Journal of Neuroscience*, 30(19):6482–6496, 2010. 3, 6, 42

[307] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014. 9, 170

[308] H. Zhou, H. S. Friedman, and R. Von Der Heydt. Coding of border ownership in monkey visual cortex. *The Journal of Neuroscience*, 20(17):6594–6611, 2000. 6, 16

[309] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2879–2886, June 2012. 171

[310] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proc. European Conf. on Computer Vision*, pages 391–405. 2014. 21