MODEL-BASED PERCEPTUAL GROUPING AND SHAPE ABSTRACTION

by

Pablo L. Sala

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Copyright © 2014 by Pablo L. Sala
Abstract

Model-Based Perceptual Grouping and Shape Abstraction

Pablo L. Sala
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2014

For many object classes, shape is the most generic feature for object categorization. However, when a strong shape prior, i.e., a target object, is not available, domain-independent, mid-level shape priors must play a critical role in not only grouping causally related features, but regularizing or abstracting them to yield higher-order shape features that support object categorization. In this thesis, we introduce novel approaches to grouping and abstracting image contours into a set of qualitative 2-D and 3-D parts. We do not assume any prior knowledge of the objects contained in the scene, but rather assume that they’re composed of parts drawn from a vocabulary of qualitative 2-D and 3-D model parts input to the system. We begin by proposing two methods, one contour-based and one region-based, that use 2-D part models to both group contours into regions (representing projections of object surfaces) and abstract the resulting contour groups to yield a set of 2-D idealized part models. The process of contour abstraction can be thought of as a form of “controlled hallucination” which yields many competing 2-D part hypotheses. To improve part hypothesis precision, we next explore two methods for hypothesis selection that exploit contextual information. In one approach, we exploit spatiotemporal coherence of part hypotheses in a dynamic environment, and formulate hypothesis selection in a graph-theoretic, probabilistic framework. In the second approach, we exploit the spatial context encoded in the aspects representing the projections of the surfaces of the 3-D volumetric parts comprising the vocabulary. Finally, we introduce a technique
that is able to recover the 3-D pose of a volumetric part from a collection of grouped 2-D parts hypotheses.
A June, Marquito, Bella, y Mamá.
Acknowledgements

I would like to start by thanking my supervisor, Sven Dickinson, for his continuous encouragement and support. Despite his often very busy schedule, he regularly made himself available for discussion and devoted countless hours to reading and correcting my drafts. His vision, guidance, and contagious enthusiasm have been invaluable in my growth as a researcher throughout these years.

I am also grateful to the members of my committee, Allan Jepson and Rich Zemel, as well as to my external examiner, Jianbo Shi, for thoroughly reading my thesis and for their insightful comments and suggestions.

Certainly, I cannot forget to mention my friends at the University of Toronto who enriched my lunchtimes with enjoyable conversations: Diego Macrini, Patricio Simari, Sebastian Sardiña, Andres Lagar-Cavillia, and Ariel Fuxman.

My especial thanks go to my family. To June, for her patience, love, and unconditional support over these years. And to Marquito and Isabella for the ongoing joy they bring to my life. I remember Mom with extreme gratitude for the many sacrifices she made to support me through college, and for her persistent encouragement during my years of graduate school. I am also thankful to my brothers, Mario and Jorge, and their families, who have supported me in different ways.

Finally, I am grateful to God, who has granted me the undeserved ability to accomplish this work.

Praise be to the name of God for ever and ever; wisdom and power are his.

He changes times and seasons; he sets up kings and deposes them. He gives wisdom to the wise and knowledge to the discriminating. He reveals deep and hidden things; he knows what lies in darkness, and light dwells with him.

Daniel 2:20-22
Contents

1 Introduction 1
  1.1 Thesis Overview ............................................. 5
    1.1.1 Summary of Contributions ................................. 11

2 Related Work 14
  2.1 Granularity of Shape Prior .................................... 19
    2.1.1 Weak models ............................................... 20
    2.1.2 Intermediate-Level Models ............................... 34
    2.1.3 Strong Models ........................................... 46
  2.2 Conclusions .................................................. 57

3 An Edge-Based Approach to Part Hypothesis Generation 58
  3.1 Overview of the Approach .................................... 59
  3.2 Part Hypothesis Generation .................................... 61
    3.2.1 The Model Grid ........................................... 63
    3.2.2 Rotation Invariance ....................................... 63
    3.2.3 Scale and Translation Invariance ......................... 64
    3.2.4 Edge Map Resampling .................................... 65
    3.2.5 Model Screening .......................................... 66
    3.2.6 Model Indexing .......................................... 67
    3.2.7 Model Scoring .......................................... 68
3.3 Part Hypothesis Selection ........................................... 70
3.4 Demonstration ......................................................... 73
3.5 Conclusions .......................................................... 76

4 A Region-Based Approach to Part Hypothesis Generation 79
4.1 Overview of the Approach ........................................... 80
4.2 Finding Consistent Cycles ........................................... 84
   4.2.1 Path Initialization .............................................. 84
   4.2.2 Path Extension .................................................. 85
   4.2.3 Training the Classifiers ....................................... 86
4.3 Abstracting the Shape of a Consistent Cycle ..................... 88
4.4 Results .............................................................. 93
4.5 Comparison of the Edge-Based and Region-Based Part Hypothesis Gen-
eration Methods ....................................................... 100
4.6 Conclusions ........................................................ 102

5 Part Hypothesis Selection by Spatiotemporal Coherence Priors 104
5.1 Related Work ......................................................... 106
5.2 Overview of the Approach .......................................... 107
5.3 Temporal Coherence of Consistent Cycles ......................... 108
   5.3.1 Retrieving Consistent Cycle Trajectories .................... 109
   5.3.2 Probability Density Estimation ............................... 113
5.4 Results .............................................................. 114
5.5 Conclusions ........................................................ 118

6 Recovering Abstract Volumetric Parts 119
6.1 Abstract Volumetric Part Recovery Overview .................... 121
6.2 Graph Construction .................................................. 125
   6.2.1 Labels .......................................................... 126
6.2.2 Aspect Representation ......................................... 128
6.2.3 Face Nodes .................................................... 134
6.2.4 Relational Nodes ................................................. 136
6.3 Grouping Faces into Aspects ........................................ 142
6.3.1 Labeling Optimization and Parameter Learning .............. 146
6.4 From Labelings to 3-D Segmentations ............................... 147
6.5 Evaluation ......................................................... 151
6.6 Limitations ......................................................... 154
6.7 Conclusions ......................................................... 159

7 Conclusions .......................................................... 165
7.1 Review of Contributions ............................................. 165
7.2 Future Work ......................................................... 168

Appendix A Improvements to the Consistent Cycle Detection Algorithm 171
A.1 Image Segmentation ............................................... 172
A.2 Consistent Cycle Detection ....................................... 178
A.2.1 Search Algorithm .............................................. 179
A.2.2 Consistency Classification .................................... 182
A.2.3 RDF Training .................................................. 184
A.2.4 Path and Cycle Features ...................................... 185
A.3 Non-Maximum Suppression ....................................... 188
A.3.1 Approach ...................................................... 190
A.3.2 Model Features and Training ................................ 191
A.4 2-D Abstraction .................................................... 194
A.4.1 Training Models .............................................. 194
A.4.2 Abstract Model Classes ..................................... 194
A.4.3 False Positive Filtering ..................................... 196
Chapter 1

Introduction

Object categorization refers to the problem of determining the object category of previously unseen object exemplars. Although an apparently trivial task for humans, this is an extremely challenging computational problem, which remains largely unsolved. The difficulty of categorization stems from the large variety of different images that can be produced by member objects of the same category, due to changes in pose and illumination, variations in scale, occlusion, within-class variability, and object articulation. (See Figure 1.1.)

In Figure 1.2 (a), the image of a few object exemplars from a common category are shown. In spite of their variations in structural detail, color, texture, and viewpoint, humans effortlessly recognize all these objects as belonging to the same category, i.e., binoculars\(^1\). What specific image features common to all these objects are exploited by the human visual system in order to recognize all of them as of the same class? Certainly, appearance does not play an essential role in this process since there is a large variation in color, texture, and shading among these exemplars. Moreover, humans are able to quickly recognize the line drawing in Figure 1.2 (b) as another exemplar from this category, even though it is devoid of any color, texture, or surface markings.

\(^1\)In particular, they all belong to the subcategory of binoculars with cylindrical barrels and a single bridge.
Chapter 1. Introduction

Figure 1.1: Challenges of categorization: (a) changes in pose and illumination; (b) variations in scale; (c) occlusion; (d) within-class variability; and (e) object articulation.

Figure 1.2: Objects from the “binoculars” category: (a) a few exemplars from the category; (b) a line drawing of an exemplar of the category; (c) a volumetric abstraction of the category.

What about shape? These objects differ in their precise 3-D geometry as they all have different specific geometrical designs, i.e., no two exemplars share the exact same 3-D geometric model. However, we can argue that all the objects roughly have the 3-D
shape described by the model shown in Figure 1.2 (c), which depicts the composition of three volumetric parts, namely, two cylinders joined by a cubic prism. Such a 3-D model captures the prototypical shape of the category without paying attention to detail. So we see that it is not until the objects’ 3-D shapes are abstracted into their qualitative components that their commonality fully arises. In some instances, the same qualitative shape is shared by more than one category, e.g., an orange and a soccer ball, and in such cases, other visual cues, such as color and texture, may need to be employed to disambiguate the object’s category. Still, shape remains an important visual cue for categorization.

In the absence of object-specific knowledge, low-level image features need to be organized into higher-level structures before shape can be recovered. Humans have an uncanny ability to spontaneously organize ambiguous visual stimuli into coherent groups, even without knowing what they are looking at. This essential mechanism, that does not rely on any object-specific knowledge, is referred to as perceptual grouping. Very important visual tasks, such as scene understanding and object recognition, are facilitated by this ability. Effectively, perceptual grouping guides the parsing of a scene into its components, swiftly eliminating from consideration a combinatorial number of potential interpretations. As in the human case, computational perceptual grouping is no less important when dealing with unconstrained scenes. Testing for the presence of all possible objects under all possible image conditions is intractable. Perceptual grouping is necessary to deal with such complexity.

The problem of computational perceptual grouping received considerable attention before the advent of appearance-based recognition, when object models were typically shape-based and image features were typically contour-based [98, 19, 8, 97, 135, 40]. Moreover, while object databases were rather small, it was generally assumed that a

\footnote{Although there are some object categories that are not primarily defined by shape, shape is a defining factor for a very large number of object categories, including most man-made objects and natural organisms [11, 132, 133].}
linear search of a database, i.e., matching the image features against each model in succession and choosing the best-matching model, was an unacceptable strategy, for it did not scale to very large databases. In an effort to achieve sublinear scaling, much effort was devoted to the problem of object indexing, i.e., using a set of image features to query the database for candidate objects that might account for the image features. An effective query structure, or index, should be small enough to be reliably extracted, yet discriminative enough to aggressively prune the database down to a few promising candidates. Since image features were contour-based, perceptual grouping played a major role in grouping together contours that were unlikely to co-occur by chance. Moreover, grouping was based not on object-level prior knowledge, but rather on mid-level (object-independent) prior knowledge. Such grouping was essential, since local contour features were highly ambiguous, and without grouping them into more discriminative structures, effective indexing into large databases was problematic.

The object categorization community’s common reformulation of the recognition problem as a detection problem (in which the image is searched for a single target object) diminished the role of perceptual grouping. Even with the re-emergence of image contours as the basis for categorical models (e.g., [52]), the continuing focus on object detection means that the stronger shape prior offered by a detector subsumes the domain-independent shape priors that make up the non-accidental properties that define perceptual grouping. In other words, the process of domain-independent, bottom-up perceptual grouping to extract a meaningful indexing structure in order to select promising candidates is largely unnecessary, since in a detection task we know what, i.e., which candidate we’re looking for, and where we’re looking for it (inside the sliding window).

There are signs that the categorization community is not only returning to the more categorical feature of object shape, but to the more general problem of recognition from a large database. In turn, the need to group together contour features into powerful indexing structures may stimulate interest in domain-independent perceptual grouping
Yet a simple return to classical grouping techniques is insufficient, for while non-accidentally related contours in an image may be grouped, there is still a semantic gap between the resulting contour groups and the shape structures that comprise a categorical shape model. Whereas simple groups of contour features may have been sufficient for indexing into a database of shape exemplars, today’s interest in categorization will require not only the grouping of causally related contour features, but their shape abstraction\textsuperscript{3} to yield categorical models, i.e., higher-order shape features that are invariant to within-class variation. This raises two important challenges: 1) how to perceptually group related contours; and 2) how to perceptually abstract the groups into high-order shape features that are more generic and less specific.

\section{Thesis Overview}

In this thesis, we develop novel approaches to grouping and abstracting a set of qualitative (2-D and 3-D) parts from 2-D images. We assume no object-level prior knowledge and, like the perceptual grouping community, assume only a mid-level shape prior. However, our shape prior is stronger than classical Gestalt features such as symmetry, parallelism, collinearity, etc. Specifically, our mid-level shape prior takes the form of a fixed, small user-defined vocabulary of simple 3-D part models and their associated vocabulary of 2-D projections \cite{38}. The parts can be combined in various relative sizes, orientations, and attachments to yield a potentially infinite number of object configurations. Such parts can provide enough knowledge to drive the grouping and abstraction processes without assuming knowledge of a target object. Since different object domains may demand different vocabularies of parts, it’s essential that the framework be independent of the part vocabulary; therefore, the vocabulary is an input to our framework. While the proposed

\textsuperscript{3}There are various forms of abstraction, e.g., \textit{functional abstraction}, where object categories are represented by knowledge about object function \cite{162}, and \textit{geometric abstraction}, where categories are represented by their coarse prototypical shape \cite{11}. In this thesis, we focus on geometric abstraction.
methods do not depend on a particular vocabulary, they assume that objects can be decomposed into simple parts. These approaches are potentially well suited for vocabularies where parts are simple and have high symmetry (e.g., superquadrics, generalized cylinders, geons). Diverging from classical techniques, we don’t assume one-to-one correspondence between extracted image contour features and part model contour features. Instead, we draw on the ability of a detector at the \textit{part} level, instead of at the \textit{object} level, to guide contour abstraction, and we introduce novel mechanisms to generate 2-D abstract part hypotheses given a collection of image contours as well as 3-D part hypotheses from groups of 2-D abstract parts. Thus, the resulting framework takes as input a 2-D qualitative part vocabulary, representing the 2-D aspects of a 3-D part vocabulary, and an image, and outputs a set of qualitative parts drawn from the vocabulary that are supported by image evidence.

Figure 1.3(a) shows images of two object exemplars that belong to the same class (bowl), while Figure 1.3(b) shows their extracted contours; note that corresponding
contour-based features are seldom in one-to-one correspondence. In Figure 1.3(c), we show sample instances from a simple vocabulary of 2-D shapes that will be used to group and abstract the contours in Figure 1.3(b). In Figure 1.3(d), we overlay the abstract shapes recovered by our algorithm (from Chapter 4). If we examine carefully the image contours in both images, we observe that due to within-class variation or noise, there are few corresponding contours between the two parts. As noise and within-class variation increase, methods that rely on one-to-one feature correspondences among specific contour-based features may fail. As in the previous case of the binoculars, only by examining the abstract shapes defined by these contours does commonality between the two exemplars emerge.

We approach this problem in two stages, as shown in Figure 1.4. Through an initial part hypothesis generation process, a set of plausible 2-D part hypotheses is generated from the input image. This is a very local process, in which hypotheses are independently generated based on support from image evidence. For this reason, this process’ output is a large set of redundant and even contradicting hypotheses, each independently supported by local image evidence. Effectively, it is infeasible to determine the relevance for recognition of a feature group or part hypothesis derived primarily from local image information, as hypothesis relevance cannot be measured in terms of the saliency or goodness of the features that were grouped. Consider, for example, an image of a zebra, where the most salient image edges are those induced by the animal’s stripes. However, it is the edges in the zebra’s occluding contour the ones that are most relevant to determine its shape.

A subsequent part hypothesis selection step is therefore needed to determine the subset of parts that better accounts for the scene. This is a more global step, which takes into account the image context. Such a selection process aims to identify good parts, i.e., parts that are well supported by image evidence, which also interact consistently with other good parts in the scene, and whose 2-D shape is consistent with the projection of a 3-D
part. Additionally, if instead of a single image, a video is available as input, the shape of a good part is also expected to remain stable under small changes in viewpoint.

In particular, this thesis begins by presenting two approaches, one edge-based and one region-based, to group image contours into abstract part hypotheses using a small vocabulary of simple 2-D part models. Such a vocabulary, which is provided by the user as an input to the problem, is formed by a collection of intermediate-level shape models, and does not assume any knowledge of the objects present in the scene. In our implementation, we used models whose shapes approximate the 2-D projections of faces from simple volumes (i.e., cubic prisms, cylinders, tapered-cylinders, ellipsoids, and bent ellipsoids) at multiple scales, orientations, and viewpoints. The results obtained from these two approaches show how a small vocabulary of intermediate-level shape models is able to drive a perceptual grouping process to generate powerful abstractions without the need to assume any knowledge of the objects present in the scene.

The first of these two approaches (Figure 1.5) takes as input a hierarchy of edge maps computed at different scales from a single image. Using a novel shape indexing structure generated from the user-defined model vocabulary, a set of abstract 2-D parts is hypothesized from contour information in the edge maps. Under the assumption that surfaces that are causally related tend to fit together in the world, we present a novel surface compatibility prior that, although very generic, is able to effectively prune false positives from the set of generated 2-D part hypotheses. We present results showing how much pruning we are able to accomplish using such a generic surface compatibility prior.
The second approach takes as input an image region oversegmentation of a 2-D image. (Figure 1.6 (b) shows a region segmentation of image (a).) By means of a novel model-based region grouping approach, 2-D part hypotheses (d) are generated for each group of adjacent regions (c) whose merged shape is consistent with one or more models in the input vocabulary. The shape of the hypothesized contours is abstracted by means of a novel approach using a new active shape model (ASM) fitting scheme applied to a single ASM trained over all deformations and poses of the vocabulary models. A simple approach to hypothesis selection is followed, in which contour hypotheses are ranked by the fitting error between their actual shape and the ideal shape of their abstraction. The obtained results show that this approach is able to recover the abstract shape of ground-truth object surfaces (e), despite irregularities in the actual surface contours and minor undersegmentation.

This thesis also presents an approach to 2-D part hypothesis selection based on spa-
tiotemporal coherence priors when a video sequence, in which there is relative motion between the camera and objects in a scene, is available. (Figure 1.7 (a) shows two frames of a video sequence input to the approach.) Under the assumption that the shape of an actual surface does not change dramatically from frame to frame, the spatiotemporal coherence of a perceptual group is exploited to prune false positives. (The top recovered 2-D parts and their rankings for the frames in (a) are shown in (b), and (c) shows the ground-truth parts.) The achieved results show that a great improvement can be achieved in the precision and recall of the generation of ground-truth hypotheses by simply considering spatiotemporal shape constancy, without assuming any object prior.

Finally, the thesis introduces a novel approach to perform 2-D part hypothesis selection and 3-D shape abstraction from prior knowledge of abstract volumetric parts and their projections. (Figure 1.8 shows example of the final 3-D abstractions (c) of various images (a) and their region segmentations (b).) This problem is formulated as a graph labeling task, in which a novel shape descriptor for efficient indexing of groups of closed contours into 2-D projections of a volume’s surfaces is employed. Unlike the other hypothesis selection approaches previously introduced in this thesis, whose output simply
consists of a set of 2-D parts with no explicit grouping of those parts into higher-level abstractions of the scene, 2-D part hypotheses are simultaneously selected and grouped into 3-D parts, recovering their volumetric parameters and pose. The results show how the presented approach is able to recover abstract 3-D shape in spite of the irregularities in the original image contours, and perspective deformations, like foreshortening.

1.1.1 Summary of Contributions

This thesis explores model-based perceptual grouping and abstraction, thereby making the following contributions:

- Chapter 3, published in [145], introduces a new model-based approach for 2-D part grouping and abstraction from a single 2-D image, taking as input a hierarchy of edge maps.
  - We introduce a novel translation-, rotation-, and scale-invariant shape indexing structure for the grouping and abstraction of image contours.
Unlike most previous methods, that are restricted to silhouettes or scenes of textureless objects, the proposed approach works on images of objects containing some structural detail.

In contrast with many previous approaches, which assume a one-to-one correspondence between image and model contours, the presented method effectively abstracts groups of contours in the scene into qualitative parts.

Our approach is not limited to a particular model vocabulary, but is able to accommodate any vocabulary of simple (i.e., exhibiting a high degree of symmetry) 2-D parts.

Chapter 4, published in [146], introduces a new model-based approach for 2-D part grouping and abstraction from a single 2-D image, taking as input an image region segmentation.

We introduce a novel approach for finding cycles corresponding to 2-D part hypothesis contours, which manages complexity through a set of learned classifiers that prune cycles whose shapes are not consistent with input models.

The presented approach works on images of scenes containing lightly textured objects.

Our method effectively abstracts groups of contours in the scene into qualitative parts.

The proposed approach can accommodate any vocabulary of simple (i.e., exhibiting a high degree of symmetry) 2-D parts, which is an input to the system.

Chapter 5, published in [147], introduces a method to solve the problem of 2-D part hypothesis selection from a video sequence of a scene using spatiotemporal shape coherence priors.
Chapter 1. Introduction

- We introduce a novel probabilistic, graph-theoretic formulation of the problem of 2-D part hypothesis selection from video.
- The proposed method is able to effectively select 2-D part hypotheses that correspond to actual object parts across frames.

- Chapter 6 (yet unpublished) introduces an approach for 2-D part hypothesis grouping and 3-D abstraction from a single image.
- We introduce a novel graph-labeling formulation of the problem of 2-D part hypothesis selection that exploits aspect-based relational constraints.
- We introduce a novel indexing mechanism to recover 3-D volumetric abstractions from groups of 2-D abstractions.
- Our method is able to recover the abstraction of 3-D parts from real images of real scenes containing lightly textured objects and some structural detail.
- The proposed approach is not restricted to a particular vocabulary of 3-D models, but can accommodate any vocabulary of simple (i.e., exhibiting a high degree of symmetry) 3-D parts.

This thesis is organized as follows: Chapter 2 presents related work, Chapters 3 and 4 describe the edge-based and region-based 2-D part abstraction approaches, respectively. Chapter 5 describes the 2-D part hypothesis selection using spatiotemporal shape coherence priors. Chapter 6 discusses a method to simultaneously perform 2-D part hypothesis grouping and 3-D abstraction, and Chapter 7 summarizes the conclusions from our work.
Chapter 2

Related Work

Two quite distinct problems are referred to by the term object recognition: exemplar-based recognition, mainly interested in detecting the presence of particular object exemplars for which their exact shape and/or appearance is known (e.g., [97, 98, 169, 74, 27]), and generic object recognition, aiming to identify the category of previously unseen object exemplars in an image (e.g., [127, 38, 51, 47, 53, 49, 171, 55]). This survey focuses on the set of approaches that have this last goal in mind, focusing on shape-based approaches. Shape representation forms the backbone of shape-based object categorization, and is often explored in the context of a recognition task. In turn, shape representations have to be recovered from images, requiring image segmentation, perceptual grouping and, in some cases, shape abstraction. Therefore, in the literature review that follows, we begin with a brief discussion/review of the recognition problem to help set the stage for our our more focused review on the more relevant (to this thesis) related work on image segmentation, perceptual grouping, and shape abstraction.

In its early days, the recognition community approached the generic object recognition problem assuming no knowledge of what objects or object categories were present in the image. Such uninformed recognition, which attempted to recognize objects based on their coarse, prototypical shape was typically undertaken by recovering qualitative
shape descriptions with the intent of indexing into a database of known object models. Although object databases could be large, objects were composed of a small number of volumetric, category-independent 3-D parts drawn from a small vocabulary. Part-based representations were employed in order to deal with clutter, occlusion, within-class deformations, and articulation. Moreover, part models were object-centered (as opposed to view-based projections of the volumes), ensuring a reduced model representation.

In the 1980s and early 1990s, generic object recognition enjoyed some success in images of controlled scenes, restricted to simple objects devoid of markings or texture and under controlled illumination (e.g., [127, 39, 38, 37, 129, 130, 87]). The difficulty of recovering qualitative 3-D shape from a single 2-D image of a real scene stemmed from the lack of effective mechanisms to organize low-level image features into higher-level shapes. In the human visual system, the organization of low-level image features into higher-order structures without a priori knowledge of what is in the scene (even before shape can be recovered) is realized by a powerful perceptual mechanism known as perceptual grouping. Although a well-defined set of perceptual grouping laws was identified by the Gestalt psychologists in the early 20th century, a computational model did not emerge until Lowe’s work in the mid-80s [97]. Perceptual grouping alone, however, is insufficient to recover higher-level shape since there is the need for an abstraction mechanism to bridge the gap between image contours and abstract model contours. The lack of such a mechanism restricted the success of these approaches to simple scenes in which there was a one-to-one mapping between image and model contours.

In the early 1990s, the vision community started to back away from generic object recognition toward exemplar-based recognition strategies where knowledge of the object’s appearance is assumed. This paradigm shift was the result of the success of appearance-based approaches [173, 117] which, for the first time, worked on images of real objects and which had no need for object segmentation or perceptual grouping. Although the recognition power of these methods was limited (i.e., they were only able to recognize
specific object exemplars previously seen at training time) and despite some serious initial limitations (e.g., difficulty in dealing with occlusion, background clutter, changes in illumination, rotation, and scaling), some of which were eventually overcome, they enjoyed enormous popularity. However, given the global nature of the templates, invariance to scale and viewpoint could not be achieved.

In the late 90s and early 2000s, appearance-based approaches evolved from these global, view-based appearance models to rigid configurations of invariant local features, (e.g., [100, 21]). Interest in the problem of generic object recognition, lately referred to as object categorization, returned due, in part, to the emergence of domains where exact object appearance and/or geometry is unknown (e.g., content-based image retrieval, non-rigid object recognition, robot interaction in unexplored environments) and, in part, to the emergence of machine learning tools that offer the potential to learn restricted category detectors from local image features.

Categorization was now, however, reformulated as an object detection problem, in which an object model for each category (learned from training images) was employed to segment instances of the category. This new approach is not unlike the classical formulation of model-based target recognition, in which knowledge of a target model is used to constrain segmentation and grouping. Under this new trend, research was focused on the development of effective category detectors, aimed at detecting the presence of a particular object category while also allowing for some level of intra-class variation. This evolution was also fueled by advances in the field of machine learning, which yielded the development of such category detectors in the form of powerful templates learned from examples with various degrees of generalization and based on configurations of category-specific distinctive local image features. A categorization framework was thus reduced to successively applying each category detector on an input image.

Success in category detection was initially restricted to certain object categories where members shared a set of distinctive local features, as is the case for cars, faces, motor-
cycles, planes, horses, and pianos (e.g., [51, 47, 142, 95, 22]). Recently, more flexible
detectors have successfully been proposed for object classes with substantial intra-class,
viewpoint, and appearance variation (e.g., [54, 88, 52, 53, 49, 125]). Although having
enjoyed success on a variety of object categories in real imagery, this detection-based
approach to categorization does not scale up to larger object databases when the con-
tents of the image are unknown. Biederman [11] estimated that there are about 30,000
object categories recognizable by a human adult. Lack of viewpoint invariance is another
important drawback for these approaches since they are all view-based and their models
generally account for entire objects, therefore requiring the storage of a different model
for each topologically distinct viewpoint (i.e., aspect) of each object.

This becomes an intractable problem, for Plantinga and Dyer [134] showed that even
for a polyhedral object with \( n \) vertices, there are \( O(n^9) \) different aspects under a per-
spective viewing model. This problem is accentuated even more in the presence of part
articulation. In conclusion, a successive matching of each model from a large database to
an image is an infeasible approach; the number of potential candidate objects needs to
be narrowed down first, before each candidate can be tested. Recently, there have been
some attempts to overcome the scalability issues of detection-based categorization (e.g.,
[171, 57, 55, 187, 84]) where composite models made up from sets of shared parts among
multiple categories are learned from image data and a hierarchy of parts relates simple
contour fragments and their progressive combinations into contour models of different
categories.

Although the state of the art is continually advancing, it still considerably lags hu-
man performance. An important question to ask is whether models of distinctive local
features can be employed for more complex recognition tasks, and whether they have the
potential to attain a performance comparable to that of humans. Actually, there seems
to be evidence that the usefulness of distinctive local features for object detection di-
minishes exponentially as the number of object categories increases [35, 63, 58, 166], i.e.,
as the confusion between a larger number of classes or from similarly-looking categories increases.

This is not surprising since, in general, recognition approaches based on distinctive local features make the limiting assumption that among the members of a category, there exists a one-to-one correspondences among such local features. While true for very restricted categories, such as cars, faces, motorcycles, planes, horses, etc., this restriction does not hold for general categories, where a single correspondence among such local features may not exist for two members of the same category. For object categories defined by their shape, within-class shape similarity exists not at the level of local 2-D features (abstractions of small image patches centered at interest points) or even local 3-D features (corners, spin images, etc.), but rather at the level of more abstract, compositional shape representations. Hence, qualitative shape recovery seems to be an important step toward a general solution to the object categorization problem, and success in this field may be limited without the ability to extract qualitative shape.

As mentioned earlier, prior to the emergence of appearance-based categorization, the extraction of volumetric, domain-independent part-based descriptions received considerable support in both the computer and human vision research communities. Unfortunately, in the computer vision community, the machinery necessary for abstracting volumetric parts from real images of real objects did not exist, and both communities moved toward image-based descriptions, reducing – even eliminating – the representational gap between image and model features [82].

The nature of prior knowledge of the constituent shapes making up the objects in an image plays a critical role in the abstraction of such shapes. Such prior knowledge can range from low-level affinity rules, stating how basic image features (e.g., edges or pixels) can be grouped according to their homogeneity, all the way to a precise and detailed description of the entire object’s shape, including all intermediate possibilities.
2.1 Granularity of Shape Prior

The nature of the prior knowledge of shape employed in the shape recovery process can be classified in terms of a property that ranges over the Cartesian product of two orthogonal model dimensions. The first of these is *model scope* (i.e., models of entire objects *vs.* models of parts). The other dimension is the *model specificity* at which the object’s or part’s geometry is described by the model, and hence, the amount of deformation that the model is allowed to undergo and still be considered an adequate representation of an object or part’s shape.

We define a taxonomy of shape priors in an attempt to present a classification of the discussed methods by their level of abstraction. In this study, we divide this measure into four major segments: weak models, intermediate-level models, strong models, and very strong models. Aware that any partition of the spectrum into meaningful segments is arguable, as the division lines tend to be fuzzy, the presented division offers an intuitive taxonomy that allows the coarse classification of the discussed approaches into four distinctive levels of abstraction.

Our proposed partition of the spectrum is shown in Table 2.1. A model at the “very strong” end of the spectrum is one that accounts for the exact geometry of an object shape, even to the point of establishing its precise measurements. Strong and intermediate-level models impose looser constraints on the geometry, thus providing a more qualitative description for the shape. They do not commit to particular instances, but account for a large set of instances within a certain shape class. The difference between them is that strong models are models of entire objects, while intermediate-level models are models of generic (i.e., not object-specific) parts. Models falling at the “weak” end of the spectrum generally impose very weak constraints on shape but rely on low-level affinity rules to form groups of causally related features, and therefore, likely belong to the same object or object part’s shape (e.g., a region of homogeneous texture or edges forming a closed contour).
Table 2.1: Model Specificity vs. Model Scope: a taxonomy of shape-based recognition approaches.

Blank entries in the table correspond to configurations with no existing cases, either due to the impossibility of the configuration, or its non-practicality: when there is no explicit shape prior, no notion of part or object model is possible; similarly, when the model consists of a set of grouping rules based purely on local affinity, a notion of model specificity appears superfluous. Although the space of CAD-based modeling techniques did not include many examples of non-rigid objects [99], their global alignment techniques could be applied locally to parts, thus constituting approaches that would appear in the table’s entry for exemplar-based part models. Since such methods are not the focus of this survey, we have left that table’s entry blank as well.

We can see that there is an inverse relationship between the strength of a model and its level of abstraction, and therefore model strength becomes an important dimension by which the set of approaches attempting shape recovery can be classified. In this survey, we focus exclusively on approaches using weak, intermediate-level, and strong models, leaving out very strong models, as it would be impossible to discuss all classical work on model-based object recognition falling within that portion of the spectrum.

2.1.1 Weak models

The Weak Models portion of the model strength spectrum comprises the set of bottom-up approaches that, without assuming an explicit shape prior for the objects in the scene, pursues object recognition simply as a homogeneous feature extraction problem.
This is attempted by relying on perceptual grouping mechanisms, including smoothness and symmetry, to guide the grouping of elementary image features into an object or its parts. In particular, object recognition can be formulated as a surface-based recognition problem, and since object surfaces project to 2-D regions in the image, object recognition can thus be based on the analysis of 2-D image regions and their relations. In order to do surface-based recognition, techniques to extract surface projections (i.e., 2-D regions) from images are needed. In the absence of shape priors, the segmentation process has to rely on certain assumptions drawn from knowledge of general properties of object appearance, lighting effects and the image formation process.

These assumptions are made by the set of approaches that perform segmentation based on pixel affinity, i.e., methods that cluster pixels into regions such that regions are formed by pixels which have certain homogeneity or present some sort of spatial continuity. The approach of Comaniciu and Meer [28] is an example of a method in which pixels are grouped into regions of similar hue. Image segmentation is obtained by applying mean-shift clustering (a method to find the modes of a multi-variate distribution, originally presented by Fukunaga and Hostetler [61]) to pixel color. Mean-shift is presented as a natural method to cluster color values arising from real images. In the general case, there is no a priori knowledge of the number of color clusters in an image. Moreover, the density of each cluster can have an arbitrary shape, not being restricted to a particular model (e.g., Gaussian).

Instead of just clustering pixels with similar hue, other approaches attempt to decompose the image into regions with homogeneous texture. Such is the work of Galun et al. [62]. To cope with challenging textures, their approach adaptatively identifies the shape of texture elements, characterizes them by their size, aspect ratio, orientation, brightness, etc., and distinguishes between different (although similar) textures by means of various statistics of these properties. Texture is analyzed at multiple scales, allowing the identification of larger texture elements that are textured in themselves. The textures
are characterized by the statistics of a set of filter responses. An extensive interaction between the shape measures and filter responses happens during the algorithm, e.g., the shape of a texture element determining which filter responses are relevant in each case. The formation of texture elements is at the same time determined by particular configurations of filter responses. Good results are shown for a set of natural images containing challenging textures (i.e., where some sort of camouflage is present: squirrel on a tree trunk, polar bear on an icy background, owl on a stack of straw).

To guarantee that the segmentation decomposes the image into spatially coherent regions, some methods use a graph-based representation of the 2-D image. One such approach is that of Felzenszwalb and Huttenlocher [50]. In their model, nodes correspond to regions (initially, individual pixels) and edge weights to intensity differences between connected regions. A greedy process merges pairs of neighboring regions based on their pairwise difference (i.e., edge weights between the two regions) and the internal difference within each region (i.e., edge weights within a region). Although the process is greedy, it guarantees obtaining “good” segmentations (i.e., neither “too coarse” nor “too fine”) given a certain established criterion. The algorithm is efficient in practice, taking less than a second per $640 \times 480$ image.

Statistical Region Merging [120] is another graph-based image segmentation approach based on a statistical image generation model under the assumption that variations in color inside the same region should be smaller than those across different regions. By a process similar to the previous method, regions are grown and merged following a certain order and based on a merging predicate defined in terms of a statistical analysis of region pixel color distributions. A contour grouping approach of Arbeláez [4] is able to output region segmentations by integrating local contour cues along region boundaries in combination with region attributes. Based on similar ideas, Arbeláez et al. proposed a grouping algorithm to construct a multi-scale region segmentation from the output of any contour detector. This algorithm is capable of producing state-of-the-art segmentation
results when coupled with a high-performance contour detector.

The scope of the decision employed at each step in the clustering process of the above described approaches is essentially local, merely considering affinity between neighboring pixels or regions at the time of the step. There are also methods that also treat image segmentation as a graph partitioning problem but are able to produce final segmentations that take into account the global notion of the scene. Such is the case of Normalized Cuts, authored by Shi and Malik [154]. This method, rather than focusing on local image features and their affinity, aims to extract global impressions of a scene (i.e., the “big picture”), providing a hierarchical description of it. The image graph is segmented by repeated applications of a graph-cut criterion. The cut measure employed computes the cut cost between two partitions as a fraction of the total edge connections to all the nodes in the graph (hence its name of normalized cut). It is shown that this partition criterion measures both the total disassociation between the groups and the total similarity within groups. An efficient algorithm for computing an approximate discrete solution to the minimum normalized cut problem is presented with encouraging results.

Another graph-cut method was proposed by Wang and Sisskind [178] based on ratio cuts, which is able to recover large connected regions with minimal bleeding, similar to normalized cuts. Although the authors propose a polynomial time algorithm for the ratio-cut problem taking advantage of the planarity of image region adjacency graphs, the running time of the algorithm is still rather slow. Sharon et al. [152] approximate the solution to the normalized cuts problem obtaining a fast algorithm with a complexity that is linear in the image size. Another linear-time method solving a constrained multi-scale normalized-cuts graph-partitioning-based segmentation was proposed by Cour et al. [30]. This approach prevents the breaking of large homogeneous regions into smaller components due to its efficient use of long range graph connections.

Another set of approaches attempting image region segmentation is based on supervised learning. The approach of [60] combines region cues (brightness, color,
Chapter 2. Related Work

texture similarity between image patches) and boundary cues (intervening contours) for image segmentation. They employ an affinity function between pairs of pixels learned by training a logistic classifier over a large set of manually segmented images. Endres and Hoiem [44] introduced the idea of category-independent object proposals. Their approach produces a list of ranked regions, where the top-ranked regions are likely to be good segmentations of different objects. The method is based on generating graph-cut segmentations from seed regions guided by learned affinity functions.

A different set of bottom-up segmentation approaches within the weak portion of the model information spectrum, but in which the scope of decision is less local, is formed by a subset of methods that rely on perceptual grouping mechanisms to guide the grouping of elementary image features (whose simultaneous occurrence is likely to be non-accidental, i.e., causally related) into an object or its parts. The non-accidentalness principle or Helmholtz principle [73], introduced into computer vision by Witkin and Tenenbaum [180] and Lowe [97, 98], is a perceptual organization principle that assigns a high-probability to groupings of features whose arrangement is unlikely to have occurred by accident. That is, the less likely an arrangement of features is to occur by chance, the more likely there is an underlying unified cause or process for the arrangement.

The approach of Lowe [97, 98] was one of the first to apply the principle of non-accidentalness to computer vision. In his method, perceptually salient line groups are found based on the Gestalt laws of co-termination, parallelism, and co-linearity, which are quasi-viewpoint invariant. The probability of feature groups arising by chance is estimated and features are grouped whenever such probability is below an established threshold. Mohan and Nevatia’s [115] approach attempts to find non-accidental groups of image contours based on various geometric relationships (e.g., proximity, co-curvilinearity, symmetry, and continuity), generating a hierarchy of features corresponding to structural elements such as object boundaries and surfaces.

The Qualitative Probabilities framework of Jepson and Mann [76] is a Bayesian ap-
proach that formalized non-accidental reasoning to obtain the best plausible interpretation of an image, and provided a mechanism to reduce the complexity of searching for plausible interpretations. They demonstrated in block-world images the ability of their framework to reason about images of scenes with multiple objects and occlusion. Sarkar and Soundararajan [149] use supervised learning to group low-level features into sets corresponding to the different objects in the scene based on geometric relationships such as proximity, parallelism, and continuity, as well as photometric properties (e.g., belonging to a common region).

Segmentation methods in this subset also include those that produce a saliency transform in which image features are associated to a measure of perceptual saliency derived from Gestalt principles, such as proximity, parallelism, and good continuation. Early work in this area includes that of Shashua and Ullman [153], who proposed to compute an image saliency map where perceptually salient edgel groups with good continuation are emphasized. They developed an iterative method based on a saliency measure defined in terms of contour curvature and curvature variation. A network is initially formed by locally connecting image edgels, and associating each edgel to a processing element. By means of local computations, the approach is able to compute a local saliency value based on the aggregation of contributions from connected edgels, proportional to their agreement in edgel orientation. Globally salient structures emerge after a few iterations.

Guy and Medioni [69] proposed a non-iterative approach to obtain saliency maps from co-curvilinearity and proximity in 2-D images. Their method is based on a directional vector field computed from the contributions of edgels to their local neighborhoods in terms of their length and direction. They also presented an extension of their algorithm to 3-D, capable of inferring surfaces, 3-D curves, and junctions from sparse noisy data [70]. Their approach needs different rules to estimate orientation for different types of input; also, it is unable to handle the inference of directed curves and regions.

These drawbacks were addressed by Tensor Voting, a perceptual grouping framework
proposed by Medioni et al. [165, 114]. Based on tensor calculus and a linear voting scheme, this is a non-iterative method capable of inferring junctions, curves, regions, and surfaces from a variety of features, including points, curve, and surface elements, both in 2-D and 3-D. A tensor is associated to each site (e.g., edgel), which collects information cast by its local neighboring sites to build a saliency map for each feature (i.e., surface, curve, and point) type. Salient features are located at local extrema of these saliency maps. More recently, the Min-Cover approach of Felzenszwalb and McAllester [48] finds salient curves by interpreting that the scene is covered by a set of objects and a “background” object. Multiple object boundary detection is posed as a min-cover problem, and it is solved using an approximate greedy algorithm.

Boundary extraction and contour completion are also forms of contour grouping. The state-of-the-art in these areas is continually evolving, having achieved substantial progress in recent years with the application of supervised learning techniques. In Ren et al. [139, 138], contour completions are locally hypothesized using constrained Delaunay triangulations and completion likelihoods are globally estimated by means of a CRF based on a joint probabilistic model of continuity over all edges conditional on curvilinearity continuity features. The Boosted Edge Learning approach of Dollar et al. [41] for edge detection implicitly combines low- and mid-level info across multiple scales by means of a discriminative classifier trained on contour completion rules based on Gestalt laws, such as parallelism and modal completion. The approach of Martin et al. [113] for boundary extraction, and its extension [107], train a logistic classifier on local brightness, color, and texture features to compute a per-pixel probability of boundary (Pb) measure. The performance of this approach greatly exceeds that of edge detection algorithms based exclusively on image data.

In addition to the Gestalt principles of proximity and good continuation used by contour grouping approaches, the property of contour closure – an important perceptual grouping cue in the human visual system – has also been incorporated in a set of percep-
tual organization methods, based on the property that objects must be bound by closed contours. This is a move away from local grouping approaches, since grouping neighboring edges to form closed or convex contours demands more than just a local analysis of image features, as closure is a global property which cannot be computed locally over pairs of contours.

One of the first methods to compute contour closure in unconstrained imagery was introduced by Elder and Zucker [43]. They represent contours by their tangent vectors, which are then grouped by means of a Bayesian model of causal relations to form a sparsely connected graph. Contour closure is formulated as the search for shortest path cycles in this graph. The addition of a step to compute a global saliency relation after the computation of local edge affinity relations and before the graph search was introduced by Mahamud et al. [106]. They formulate the problem of contour closure detection as a Markov process, in which states correspond to contour fragments and local affinities between fragments are modeled by a transition matrix representing conditional probabilities of pairs of edges belonging to the same contour. A global saliency relation is computed from this matrix, whose strongly connected components correspond to smooth closed contours in the image.

Jermyn and Ishikawa [77] and Wang et al. [177] both extract closed contours via optimizing a ratio cut cost, although they apply the optimization to different graphs. The first approach is applied to the 4-connected graph over image pixels, while the second approach is applied to a graph over contour fragments, yielding the computation of a better gap measure. Although an NP-hard problem in general, ratio cuts can be solved in polynomial time in these cases given the planarity of the graphs employed. In [161], Stahl and Wang use a modification of the ratio cut method of [177], replacing total contour length by area as the denominator, promoting compactness, and thus yielding more regular shapes. Levinshtein [91, 92] applied the method of [161] to group superpixels.

Untangling Cycles is a contour grouping approach proposed by Zhu et al. [188], ca-
pable of detecting closed and open contour groups. This method works by embedding
the edge fragments into the complex domain and finding the contour groupings as circu-
lar embeddings by computing the dominant complex eigenvalues and eigenvectors of the
random walk transition matrix of a directed graph capturing the edgels’ topology. An
improvement to this method, proposed by Kennedy et al. [81], achieves better precision
in the higher-recall range.

Convexity is another perceptual salient property of closed contours. Jacobs [75] pre-
presented a scale-independent algorithm to find salient convex groups of line segments in
an image. His method is efficient on average and robust to noise and occlusion (allowing
for gaps to appear in groups) and also clutter (finding groups that meet a simple test of
saliency). He also demonstrated the use of the detected convex groups in indexing for
object recognition.

Using Jepson and Mann’s Qualitative Probabilities framework [76], Estrada and Jep-
son [46, 45] proposed a robust method for detecting salient closed contours that also works
efficiently in detecting non-convex contours. In order to reduce the number of detected
contours, they constrained the search to contours that had a certain level of smoothness
and whose compactness (measured as the ratio between the area of the contour and that
of its convex hull) was above a determined threshold. They demonstrate the suitability
and efficiency of their method in the presence of texture, clutter and repetitive image
structure.

Symmetry is another powerful perceptually salient property that has been explored for
feature grouping. An early example of a grouping approach relying on contour symmetry
working on real imagery is that of Saint-Marc et al. [144]. In their approach, they
approximate edge contours by B-splines in order to facilitate the extraction of a variety
of contour symmetries from geometric relationships between pairs of B-splines. The
method of Cham and Cipolla [25], also employing a B-spline representation, focuses
on finding global symmetries, and it can detect skewed symmetries even when contour
fragmentation, noise, and occlusion are present. Initially, points are grouped into sets of local symmetry hypotheses by means of a voting scheme. Hypotheses are finally verified using a global symmetry measure. In [26], the authors propose a measure of \textit{geometric saliency} that they use to solve the same problem.

Liu et al. [102] pose the problem of symmetry axis extraction as an optimization problem in which the best sequence of pairs of points is to be found, where Dijkstra’s algorithm is used to find the global optimum. This approach does not include a step for fitting smooth curves through the contours, as previous approaches do. They therefore need to resort to a hashing scheme to reduce the complexity of finding the best sequence among all pairs of points in the image. Loy and Eklundh’s approach [101] detects multiple occurrences of local and global rotational and bilateral symmetries even in the presence of clutter. Through a voting scheme, pairs of symmetric features are grouped into symmetric constellations corresponding to the dominant symmetries present in the image.

Stahl and Wang [160] integrate boundary symmetry with the Gestalt laws of proximity, closure, and continuity, being able to detect closed boundaries containing bilateral symmetry. The algorithm operates on contour fragments, by extracting linear edge segments with which symmetric quadrilaterals are composed. Grouping is achieved by optimizing a grouping cost function in a globally optimum fashion. This corresponds to finding the sequence of quadrilaterals minimizing the ratio of the boundary gap along the perimeter to the area, while maintaining a smooth symmetry axis. The use of the area as the normalization factor in the cost function promotes the compactness of the resulting symmetric parts. Given the combinatorial nature of the approach, and hence the large number of possible ways to fill the gaps, heuristics are employed to reduce the complexity of the problem, but the resulting running time of the algorithm still remains on the order of several minutes per image.

The previous methods detect contour groups, driven by image data without using any explicit object shape prior. Next we discuss a method that employs an extremely
simple shape prior that allows deformations driven by image data. Active contours or “snakes”, a method proposed by Kass et al. [80], is capable of finding closed contours (of arbitrary shape) surrounding smooth image regions. No constraint is imposed on object shape, as the weak prior employed by this method only assumes the smoothness of the object’s boundary (although also by the nature of the approach, it doesn’t allow shapes having deep concavities). The shape prior is represented by a parametric curve with an associated energy functional. The problem of finding object boundaries is formulated as an energy minimization problem. This method requires an initialization step, in which the deformable curve (snake) is first placed around the object’s boundary. In an iterative process, moved by the involved forces, the snake deforms, moving toward the object boundary until it finally wraps around it. The energy involved in the process includes internal curve forces that determine the elasticity and bending properties of the curve, and an external energy which depends on the image data, taking its smallest value at the feature of interest, e.g., boundaries.

Besides its high sensitivity to parameter values and its failure to detect concavities on object boundaries, the convergence of this method is dependent on the initial position of the snake, with no external forces acting on points of the curve that lie far from the object’s boundary. To overcome some of these issues, Xu and Prince [183] proposed the Gradient Vector Flow (GVF) as the external force to drive the snake, producing a more precise fit of boundaries with concavities, and having a larger capture range. The nature of the shape prior in this approach can be considered slightly weaker than that of the previous one. This approach allows more versatility for the object’s shape, since it is more effective than the previous one in segmenting objects with concavities. Since the GVF also acts inside the object boundary, the contour can be initialized across the boundary of the object and still obtain a correct segmentation. Besides its sensitivity to parameters, another drawback of this approach is that finding the GVF field is computationally expensive, rendering the method slow.
Chapter 2. Related Work

The two previous active contour methods can be classified as parametric approaches, where a parametric curve is synthesized in the image domain and moved towards features of interest, e.g., edges, by the aggregation of internal, external, and potential forces from an energy function. There are also geometric active contour methods, based on curve evolution according to intrinsic geometry measures in the image. An advantage of this type of method (over the parametric ones) is their ability to handle changes in topology in a natural way, making the segmentation of multiple objects possible because the evolving contours naturally split and merge. Also, they offer higher stability in the boundary detection when gradients have large variations as well as in the presence of gaps.

Examples of this type of method include Caselles et al. [23, 24], Malladi et al. [109], and Kichenassamy et al. [83]. The geodesic active contour approach of Caselles et al. [23, 24] (similar to Kichenassamy et al. ’s approach [83]) is based on the equivalence between a case of the classical energy snakes model and the computation of geodesics in a Riemannian space whose metric is defined by the image content. Malladi et al. ’s approach [109] is a level-set method, based on models of solid/liquid interface propagation with curvature-dependent speeds, where the speed term, induced by the image content, is used to stop the interface in the vicinity of object boundaries.

Although the results achieved by the previously cited techniques are generally good, perfect segmentation of an object from its background is generally never accomplished. Either some amount of under- and/or over-segmentation is present in most cases. Besides the previously discussed methods, which aimed to recover 2-D shapes from images, there is yet another set of approaches that attempt 3-D shape recovery from single 2-D images, also without the use of explicit shape priors. These techniques formulate the shape recovery problem as one of interpretation of line drawings. Given a line drawing of an object, these approaches attempt to recover a psychologically plausible 3-D object whose occluding contour and surface normal discontinuities project onto the lines in the input drawing. These approaches not only expect that exclusively relevant object contours
be present in the image, but the majority require that even self-occluded boundaries be visible (e.g., in a drawing of a non-degenerate view of a cube, all 12 edges of the cube are expected to be visible, i.e., including the 3 edges that are self-occluded by the cube).

These approaches are considered as possessing weak shape knowledge due to the nature of the prior employed. No explicit shape prior is used; only certain physical properties are assumed of the object’s surfaces (i.e., regions delimited by edges). They are assumed to be smooth (with some approaches constraining them to also be planar), and the junctions between two or more surfaces are required to be physically plausible, e.g., trihedral junctions. Some of these approaches assume some form of shape compactness, that being the only extent to which the overall object’s 3-D shape is constrained. Original work in this area began with Guzman Arenas in 1968 [71] and was extended by several other researches in the following two decades. Noticeable is the work of Sugihara, who in a series of papers (e.g., [163, 164]) reformulated the problem (for the case of line drawings of polyhedra) in purely algebraic terms. He provided necessary and sufficient conditions for the physical realizability of an object, and given enough additional constraints (obtained from information beyond that provided by the drawing), the space of feasible solutions could be constrained and a unique solution obtained.

The work of Marill [110, 111], who also dealt with drawings made out of solely straight lines, is one of the first to provide a completely automated system that attempted to recover a psychologically plausible 3-D interpretation of the input. Assuming an orthographic projection model, he noticed that the only unknowns of the problem are the 3-D depths of the object’s 3-D vertices which project onto the 2-D vertices of the drawing. Since there is an infinite number of 3-D wire-frame objects with such property, Marill conjectured that the object in that set having the minimum standard deviation of interior angles (MSDA) generally corresponded to the 3-D object perceived by humans when looking at the drawing. He posed this search as an optimization problem and solved it using hill climbing (a process that for certain cases fails by falling in a local minimum).
Obviating the fact that there is no theoretical justification for the presented approach, it must be said that the results shown are impressive, given the simplicity of the problem formulation and the solution presented.

Leclerc and Fischler [86] presented a solution to the problem that extended Marill’s work. They not only provided a theoretical basis to explain Marill’s success, but were able to extend the solution to deal with other cases where Marill’s failed. They conjectured that the stunning success of Marill’s approach was due to the fact that MSDA generally yielded solutions corresponding to 3-D objects that were symmetric, arguing that such is one of the perceptual preferences of the human visual system in the presence of ambiguous shape interpretations. Stating that planarity is another strong human preference, they extended Marill’s objective function to also optimize this property, improving on Marill’s results.

More recently, Pizlo and Stevenson [133] proposed that besides symmetry, compactness (i.e., $V^2/A^3$, where $V$ is volume and $A$ is surface area) is a generalization of the MSDA and planarity shape interpretation preferences for the case of curved objects, and demonstrated in various experiments the success of this principle. The case of drawings of curved objects was also dealt with in other works (e.g., [108]). It must be noted that these methods have no direct applicability to images of real scenes. Even in the case of images of real polyhedral objects (in which all lines are guaranteed to be straight), the robust detection of relevant edges (i.e., those that correspond to occluding contours and surface discontinuities) is an open problem in itself.

**Limitations, Weaknesses, Open Problems**

Only very limited success is possible when object segmentation is approached without the assumption of any higher-order shape priors. These bottom-up approaches typically terminate at the level of finding groups, and don’t attempt to fit abstractions to the recovered groups. They therefore fall short of extracting the qualitative shape of an
object in terms of its parts. Clearly, a prior on the possible constituent shapes making up the objects in an image is necessary to constrain the intractable combinatorics of not only grouping, but of shape abstraction [82], as pointed out in early work on recognition by parts, e.g., [11, 126, 37].

2.1.2 Intermediate-Level Models

The set of techniques falling in the Intermediate-Level Models segment of the spectrum addresses this issue by attempting the extraction of category-independent part-based descriptions, employing shape priors for the (finite) set of parts and their attachments from which the abstract shape of an object can be modeled. Although these approaches commit to a particular vocabulary of parts, they assume no knowledge of objects. The set of shape priors is thus finite, small, and independent of the number of different objects in the database. This modeling framework is typically invariant to part articulation, and offers the locality of representation required for recovering the visible parts of occluded objects. The decomposition of objects into parts yields a distribution on part attachment and configuration whose priors can also be exploited in part recovery.

Generalized cylinders (GC) were introduced by Binford [13] as powerful symmetry-based part models. A number of restrictions to the GC’s axis, sweep function, and cross-section were subsequently introduced by approaches that attempted bottom-up recovery of these models from images [2, 119, 112, 174, 185]. However, these approaches were only able to recover these models from images of real objects only when a one-to-one mapping between image and model contours existed since they lacked abstraction mechanisms to map image contours to abstract model contours.

Superquadric ellipsoids [6] and geons [11] emerged as alternative symmetry-based volumetric shape abstractions restricting the the complexity of the generalized cylinder. Superquadrics are volumetric models with a rich set of deformations controlled by a small number of parameters. Most successful superquadric recovery was achieved in the range
Their recovery from 2-D images was far less successful due, again, to the lack of abstraction mechanisms to bridge the gap between image contours and abstract model contours [37].

Geons, introduced by Biederman [11, 12], represented a qualitative partitioning of the space of GCs based on dichotomous or trichotomous contrastive properties readily distinguishable by humans. Wu and Levine [181] proposed an approach to fit parametric geons to multi-view range data of single-part objects. Seven qualitative shape types of geons are used in this work, including ellipsoid, cylinder, tapered and bent cylinder, cuboid, and tapered and bent cuboid. Model fitting is performed via minimizing an objective function measuring the fitting error between models and data. An obvious limitation of this approach is that it expects input shapes to correspond to single models, assuming that the challenging problem of parsing a complex object’s shape into simple part models has been solved by a previous step. Other approaches attempting geon recovery include [9, 10, 39, 38, 36, 129, 137].

Leonardis et al. [87] proposed an approach that represents the state of the art in part-based qualitative shape segmentation from range data using superquadrics. In their framework, a large number of model seeds are simultaneously initialized at discrete locations in the range image, and through an iterative recovery and selection process, these hypotheses are refined and filtered, retaining only those that more accurately model the image data. Through this methodology, starting with several models that redundantly explain the data, they are able to segment images into superquadrics even if there does not exist a decomposition of the image into non-overlapping superquadrics (a drawback that plagued previous work on segmentation of range data, e.g., [67]). The decompositions that result are very good, and stable with respect to viewpoint and noise, although no experiments with images having significant clutter and non-superquadric-like objects are shown. The parameter optimization process is sensitive to local minima, although the model-selection procedure reduces the effects of this problem by pruning most models
Chapter 2. Related Work

that represent local minima.

As mentioned above, the difficulty of qualitative shape recovery becomes even greater when a 2-D image is used as input. Some researchers have approached this problem using a vocabulary of deformable 2-D shapes to model the parts constituting the objects present in the scene. Such are the works of Shokoufandeh et al. [155] and Rosin [141, 140]. Shokoufandeh et al. use a vocabulary formed by blobs and ridges to detect qualitative shape in intensity images. The scale space of the input image is obtained by successive convolutions with Gaussian kernels of different variance. Blobs are detected as local maxima of the normalized Laplacian operator in scale-space, while ridges are detected in a similar way as maxima of a multi-scale ridge detector. A model of an object is provided in the form of a graph encoding the topology of blobs and ridges computed in an image of the object. To find such objects in an image, a graph encoding the topology of blobs and ridges detected in the image is computed, and subgraphs that are isomorphic to the model graph are searched using graph spectral information. From the results presented, it can be observed that in images of scenes with clutter, the blob/ridge detector yields many false positive and false negative responses, and cannot recover parts from textured objects.

Model-based 2-D shape recovery can be formulated as a fitting problem in which an algebraic model is fitted to image data, e.g., 2-D contours. Rosin [141] and Rosin and West [140] have explored different methods for superellipse fitting to contour data. Since a closed form solution for superellipse fitting does not exist in the general case, in the first work, Rosin explores the trade-off between efficiency and accuracy of various fitting methods. In the work with West, they present a method for segmenting curves into a series of superelliptical arcs, using a significance criteria to choose the most perceptually correct description. Zhang and Rosin [186] explore methods for fitting superellipses to partial contour data and are able to improve over classical superellipse fitting results by including gradient and curvature information in the fitting process. Their results
show that the use of gradient information reduces the number of iterations needed for convergence and improves the robustness of the fit. Besides the drawback that these approaches are sensitive to the presence of outliers, it must be noticed that all the previous contributions are limited to the fitting of superellipses without deformations.

The work of Pilu and Fisher [131] addresses this deficiency. They propose the use of a point distribution model to represent the class of shapes of a complicated model, and they apply it to the case of superellipses with deformations (linear tapering and circular bending), thus obtaining a more powerful modeling tool than a “rigid” superellipse. Apart from its sensitivity to outliers (i.e., the need for a priori knowledge of the points forming the 2-D contour), this approach requires as input complete contours. Osian et al.’s approach [123] provides a method that works even in the case of partial 2-D contours, but that solves a less ambitious problem, namely, fitting superellipses with only affine deformations. A method to segment a partial 2-D contour into a group of affine-deformed superellipses is also presented in this work.

An alternative symmetry-based approach for shape recovery from 2-D images that uses non-algebraic 2-D models is shock graphs, originally proposed for shape modeling by Siddiqi and Kimia [158], and more recently used in a recognition framework by Siddiqi et al. [159] and Shokoufandeh et al. [156]. The medial axis transform - or skeleton - of a silhouette, originally described by Blum in 1973 [14], is an area-based and boundary-based description of shape based on the loci of centers of maximal disks circumscribed by the object’s boundary. The points along the skeleton are called “shocks” since they are singularities of a curve evolution process acting on the silhouette’s bounding contour. Shocks can be classified into four types, depending on the local variation of the radius function along the medial axis: the radius function at a shock can either vary monotonically (e.g., a protrusion on the silhouette), remain constant (e.g., parallel sides), achieve a local minimum (e.g., a neck), or achieve a local maximum (e.g., the evolving curve annihilates into a single point).
Chapter 2. Related Work

The shock graph is a partitioning of a silhouette’s medial axis according to a vocabulary of qualitative parts (i.e., shock types). It is implemented as a directed graph in which nodes correspond to connected components of shocks of the same type, edges relate connected components of shocks that are adjacent in the skeleton, and edges are directed from newer to older, according to the time of formation of shocks in the curve evolution process. Medial descriptions tend to be quite unstable in the presence of minor perturbations on the silhouette’s contour. Macrini et. al [105, 103] tried to address such instability issues by proposing the bone graph, a new medial shape abstraction approach capturing a more intuitive notion of object parts based on the ligature structure of the medial axis, which offers improved stability and invariance to intra-class deformations.

The application of medial-axis approaches is rather limited, since they require as input an object’s silhouette, which is hard to obtain from a real image since the difficult problem of figure/ground segmentation has not yet been solved. The work of Levinshtein et al. [90, 93] composes approximations of object skeletons directly from real images. It generates hypotheses for medial points from a multi-scale superpixel segmentation of the input image. It then employs a learned affinity function to group the hypothesized medial points into medial branches. Groups of branches likely to correspond to the same object are grouped into a skeleton approximation by learned higher-order affinity functions. The output of their algorithm consists of a parsing of the image into object abstractions, where each object part (i.e., medial branch) is represented by an elliptical region. The skeletal branches produced by this approach are rectilinear, and therefore it has problems handling structures with high curvature variations along their symmetry axis.

An approach capable of handling such cases which also outputs curve symmetry axes was proposed by Tsogkas and Kokkinos [172], although they stop short from generating abstractions for the object parts as Levinshtein et al. do. Following a methodology similar to that of [113], they detect symmetry features at multiple scales from multiple complimentary cues (e.g., color, texture, grayscale structure, and spectral clustering in-
formation), whose proper combination for the purpose of symmetry detection is learned via supervised learning.

Recently, Narayanan and Kimia [118] introduced a bottom-up method to generate, from a real image, a bag of object part hypotheses in the form of image fragments. Their method starts from an oversegmentation of the image into *medial fragments*, each medial fragment being a basic symmetric part corresponding to the region between two proximal contour fragments and their medial axis. Part hypotheses are generated from multiple alternate groupings of medial fragments. The Gestalt rule of good continuity between medial fragments was augmented by the cue of *figural continuity*, requiring agreement between the radius of the medial axes being grouped, and *appearance continuity*, ensuring the good continuation of the appearance of the grouped fragments. They achieve an average recall of over 70% of object parts in the top 10 part hypotheses.

Rather than using 2-D models, the following set of approaches employ qualitative 3-D part models to perform shape recovery from 2-D images. The pioneering work of Brooks [19] employed generalized cylinders (GC) to represent primitive object parts, and described the decomposition of an object into its parts by means of a hierarchical graph. A specialization hierarchy of object classes is also used. Object parts are detected by fitting ribbons and ellipses to image contours, which are matched to projections of GCs, eventually generating object hypotheses via a constraint-satisfaction approach.

Another early attempt to automatically extract deformable *volumetric* part-models from a 2-D image is the work of Pentland [127]. Given a silhouette mask obtained by figure/ground segmentation of the input intensity image, the shape recovery process is realized by first segmenting the silhouette into multiple (and possibly contradicting) initial part hypotheses by a simple filtering operation, which consists of convolving a scale pyramid of the silhouette at each image position with masks corresponding to 2-D projections of the 3-D model primitives at all orientations.

Following the Minimum Description Length (MDL) principle, the subset of this set
of hypotheses that achieves the simplest description of the image is chosen. (Simplicity of description is measured in terms of the number of bits required to describe the data using the part vocabulary.) The MDL principle chooses as the best explanation for a phenomena the simplest possible explanation for it. Formally, given the observed data $X$ and a family of possible models $M$ for it defined over a parameter vector $\theta$, MDL finds the parameter vector $\hat{\theta}$ that minimizes both the description length of the data under the chosen model $M(\hat{\theta})$ and the description length of $\hat{\theta}$ (i.e., the description length of the chosen model itself). In Bayesian terms, the MDL principle reduces to a Maximum A Posteriori (MAP) likelihood estimate (e.g., see [64]), i.e., $\theta = \arg \max_\theta p(\theta|X) \propto p(X|\theta)p(\theta) = \arg \min_\theta -\log p(X|\theta) - \log p(\theta)$.

Once the best subset of part hypotheses has been selected, an initial 3-D shape recovery for each part is performed by replacing each 2-D part mask with the 3-D volume that projected onto it. If range data is available, a refinement of this initial set of volumetric part models is done by adjusting their fit to the range data evidence. This approach not only assumes that the hard problem of figure/ground segmentation has been solved, but since it works on the figure’s silhouette, it also fails to correctly segment two or more parts when their 2-D projections merge into a roughly convex blob, having problems as well when a part extends in a direction roughly perpendicular to the image plane. In the absence of range data, only an imprecise volumetric shape recovery is achieved by this method, in which all the volumetric primitives appear in a profile view.

Instead of working on the foreground silhouette, Dickinson et al. [38] proposed a method to recover shape from image contours, employing a purely qualitative vocabulary of volumetric part models. They empirically sampled the viewsphere around their chosen 3-D part vocabulary and then learned the resulting aspect hierarchy. Via a region-based method at a single image scale, they extract contours from intensity images, which are segmented and grouped by means of non-accidental pairwise relations to form boundary groups. Using the learned aspect hierarchy, probability distributions guide an inference
process that starts from these boundary groups and infers faces which, in turn, infer aspects which, in turn, infer volumetric models. Although the proposed framework is independent of the choice of part models, accommodating any qualitative part vocabulary, it assumes ideal objects in the input images, devoid of surface markings or fine structural detail, virtually corresponding to objects built themselves by composing idealized volumetric parts into contrived objects. Also, the distribution of part configurations is not exploited in their segmentation process; parts are assumed to be independent.

Deformable model fitting methods are critically dependent on model initialization and prior segmentation. Due to the intrinsic difficulty of segmentation, many such approaches have simply avoided this problem altogether, assuming a correct initialization for the models. Dickinson and Metaxas [37] proposed combining qualitative and quantitative shape recovery approaches in order to overcome this problem. The previously discussed qualitative shape recovery method of Dickinson et al. [39, 38] is used as a front end to Terzopoulos and Metaxas [168] deformable model fitting approach, to perform a segmentation of the input image into parts, and thus provides an initialization for independent model fitting processes at each part, using their corresponding contour data and exploiting the recovered qualitative pose. After the qualitative shape recovery is complete, the deformable model fitting to each part becomes a target fitting process, in which the coarse shape that the final fitted model should have is known. This knowledge is exploited to avoid local minima in the fitting process, by successively fitting each model parameter, one at a time. The recovered qualitative shape thus provides strong constraints on the fitting process.

In addition to the vocabulary independence of Dickinson et al.’s [39, 38] approach, Terzopoulos and Metaxas’s method is also general, in the sense that it can employ any quantitative shape model that can be defined using their physically-based framework. However, the choice of quantitative models is constrained to sets of models that can accurately model all instances of the qualitative shape models. This approach used ad-
hoc rules to guide the deformable model fitting process dependent on the particular qualitative shape of the part being fitted. No principled justification was given for the particular ordering of the degrees of freedom of the quantitative model that was followed in the fitting procedure.

Pilu and Fisher [129, 130] also attempt to recover generic 3-D (geon-like) solid parts from real 2-D images. However, rather than using a purely qualitative part vocabulary as Dickinson et al. do, their vocabulary is formed by superquadrics with tapering, swelling and bending deformations. Simple approximate 2-D models for the image projections of geon contours called \textit{Parametrically Deformable Contour Models} (PDCM) are proposed and used in the fitting process. The parameter space defining the PDCMs is divided into eight regions, each one yielding a topologically different contour model. Each of these regions in parameter space is called a \textit{Parametrically Deformable Aspect} (PDA).

The 3-D shape recovery is accomplished by first generating initial hypotheses for the location, orientation and scale of object parts, from which the most economical subset (following the MDL principle) is selected. Each one of the eight PDAs is initialized at a representative position and independently fitted to each part hypotheses, by maximizing the likelihood of the presence of the 3-D model (model projected contour) given the image evidence (edgels) via Adaptive Simulated Annealing. The PDA achieving the best score is considered the best fit to the part.

The use of topologically distinct aspects confines the model optimization to regions of the parameter space corresponding to models with the same projected 2-D contour topology, thus reducing the chances of falling in local minima caused by different interpretations of image features. The proposed framework is general in the sense that that it could be applicable to other part vocabularies. Apart from the quality of the segmentation being dramatically affected by the presence of noise and spurious edges, the ultimate success of the process is highly dependent on the initial aspect hypotheses which, in some of their experiments, were placed by hand. (The automatic initialization of PDAs in part
hypotheses with low-eccentricity is problematic, given the difficulty in determining their correct orientation.) Like in Dickinson et al.’s case, this approach assumes that most image contours are salient, a condition that holds only in images of textureless objects in non-cluttered scenes.

Han and Zhu [72] introduced a novel approach for recovering the shape of man-made objects from 2-D images. Their method uses attributed graph grammars to model arrangements of simple image features (e.g., lines) that are consistent with projections of 3-D shapes. They demonstrate their approach using planar rectangle primitives computed from image edge information. Although good results are demonstrated on the test images shown, none of them present significant texture or clutter. It’s conspicuous that for images having those characteristics, the use of simple primitives would demand the formulation of very complex grammar rules, rendering the framework intractable. On the other hand, to maintain the tractability of the grammar within reasonable bounds, complex shape primitives should be employed, whose detection brings us back to an unresolved shape segmentation problem.

Based on the observation that many object classes share similar features, in recent years there has been an attempt to tackle the scalability issues of single-category object detection approaches by means of building multiple-category detectors using learned parts (at various levels of structural complexity) that are common among the different categories. Part vocabularies are typically learned from object examples by a purely data-driven process. These detectors achieve a more compact representation than that of an equivalent set of single-class detectors. Typically, the number of features employed by a multi-class detector grows roughly logarithmically with the number of categories, while independently training single-category detectors requires a number of features that grows linearly with the categories [85, 171]. This type of approach not only achieves a more compact representation than that of a set of equivalent single-class detectors, but also the total computational cost of detecting all classes is substantially reduced since
shared features need to be computed only once, regardless of how many classes they are shared by. Moreover, a smaller total amount of training data is required to train these multiple-class detector.

An early example of a multi-category object detector is the approach of Torralba et al. [171], which applies supervised learning via joint-boosting on image-patch-based features. Fidler and Leonardis [57] proposed an incremental unsupervised learning approach to learn a recursive compositional shape vocabulary, thus removing the time-consuming task of manual data labeling. Starting from frequent simple and category-independent local-oriented-edge features, they learn the statistically most significant compositions of simpler parts into parts of increasing complexity and category specificity up to entire objects. Also, their approach admits the efficient addition of novel object categories without needing to retrain the entire model again. By the addition of a coarse-to-fine taxonomy of constellation models of object classes [55] to their hierarchy of parts model, they were able to achieve a speed-up in recognition times.

Recursive Compositional Models (RCM) by Zhu et al. [187], also a hierarchical approach to multi-class detection, additionally considers the problem of multiple-view detection, employing region-based features. Besides part sharing, they also model appearance sharing across multiple viewpoints. Performance is increased by this addition as it helps in cases when training data from rare viewpoints cannot be easily acquired, achieving state-of-the-art detection performance in public challenging datasets. The approaches mentioned so far have demonstrated success on databases with only a few dozen categories. Recently, Salakhutdinov et al. [148] demonstrated a successful multi-category detection approach working with a database of 200 object classes by allowing rare objects appearing in less training examples to borrow statistical strength from related objects with more training examples.

Although these approaches are typically demonstrated on rather small sets of categories, it would appear that nothing precludes them from being trained on larger
databases, expecting similar performance. (Obviously, the larger the number of categories, the higher number of primitives and intermediate parts that the system will model.) However, it has been demonstrated recently that increasing the number of categories yields an exponential drop in performance of state-of-the-art category-detection approaches [35, 63]. Similarly, performance is substantially poorer if the density of categories in the semantic space increases, even if the denser dataset has a smaller total number of categories (e.g., moving from Caltech 101 to the mammalian dataset [58] with 72 categories). A major reason for this decrease in detection performance appears to be the reduction in the effective distinctiveness of local features arising from the confusion yielded by a large number of object classes or from similarly-looking categories [166].

Hierarchical multi-class detection methods build categorical indices that are recursive compositions of local image features. It therefore seems reasonable to argue that they are plagued by the same problems that single-class detectors (based on distinctive local features) have. The indices generated by these approaches live at the level of image data, and so they lack an abstraction step that, we argue, can help them build powerful categorical mid-level indices.

Recently, there has been a renewed interest in blocks world approaches, which attempt to decompose a scene into block-like parts. Domain-independent block extraction methods applied to 2-D images were demonstrated on outdoor scenes [68] and cube-like objects in general [182], and a recent method working on RGB and depth data has been proposed [78] as well. These methods effectively output a decomposition of the recovered objects into a set of parts (i.e., blocks), whose relative dimensions and geometric relationships can potentially be used to build a categorical index.

Limitations, Weaknesses, Open Problems

The approaches described in this section were typically applied to images of simple scenes containing idealized, textureless objects in non-cluttered backgrounds. Most methods
also considered all image contours as relevant (i.e., belonging to the objects). However, in real scenes, there may be a large number of high-contrast contours, but only a few of which (if any) correspond to the object’s occluding boundaries and the lines of curvature discontinuities on the object’s surface. All the approaches that naively initialized their 3-D models using edge information are thus doomed to fail in the common case of objects possessing markings, texture, and structural detail.

Moreover, very few of these approaches attempted to do actual shape abstraction. Instead, they used test images of contrived objects virtually built by composing idealized volumetric parts models, in some cases even assuming a one-to-one mapping between model and object contours, a condition that is not necessarily true among members of the same shape category. Although qualitative models have in themselves the power to yield shape abstractions, such power could not be exploited by these approaches due to the lack of effective grouping mechanisms to single out subsets of contours corresponding to each particular part. This problem was thus avoided by using images in which most contours are relevant.

2.1.3 Strong Models

The approaches in the Strong Models region of the spectrum correspond to the set of strategies that, without imposing hard geometry constraints on an object’s shape, invoke an object-level shape model to help guide the shape recovery process. Before the first attempts to use part-based models of the 70’s, model-based recognition had already been around for over a decade in a variety of approaches employing object level shape priors accounting for the exact object geometry (generally in the form of CAD models). We classify these in the very strong portion of the model strength spectrum.

More recently, in the late 80’s and early 90’s, the vision community moved toward image-based descriptions (e.g., appearance-based categorization), where knowledge of the object’s exact geometry or appearance is assumed, thus reducing the representational gap
between image and model features. This departure was motivated by the intrinsic difficulties of the problem and the non-existence of the machinery necessary for abstracting effective shape features from real images, and encouraged by the relative success achieved by exemplar-based techniques on the recognition of certain categories of objects in images of real scenes. Starting in the early 90’s and considerably increasing in the last few years, a number of attempts have been made to model class-specific object shapes [54, 88, 52, 53, 49, 125], allowing for within-class shape variability. All these approaches use object-level shape models to guide the segmentation process, and thus fall in the strong portion of the model strength spectrum. Research is currently very active in this field; the following are some of the classic papers in this area.

Cootes et al. [29] proposed a 2-D shape model that would allow within-class shape variability. Their approach was different from the active contour models of Kass et al. [80], in that those were simply flexible templates having no bias toward any specific shape. Cootes et al. suggested the use of point distribution models to capture the statistics of a set of aligned shapes. An iterative method for image segmentation using their statistical shape templates was also proposed which, starting with the model at an initial location and pose in the image close to those of the object, is capable of adjusting its parameters to fit the image data. Since the model’s shape is deformed to better fit the data, but only in ways that are consistent with the training shapes, they called them “Active Shape Models” (ASMs). The manual registration of landmark points from all training images is required to learn the model. A close initialization of the model with respect to the object on the image is required for successful segmentation. Such initialization is quite sensitive to differences in orientation and scale, as it is to outliers and occlusion.

A formulation to implement active contours using level sets was proposed by Osher and Sethian [122], where a contour $C$ is implicitly represented by a particular level set (usually the zero level) of a certain function $\phi : \Omega \rightarrow \mathbb{R}$ defined on the image plane $\Omega \subset \mathbb{R}^2$. That is, $C = \{(x, y) \in \Omega : \phi(x, y) = 0\}$. A salient characteristic of level set methods
is that contours can split or merge as the topology of the level set function changes, and therefore they are capable of detecting more than one boundary simultaneously. More recently, level set methods have been extended to include specific object shape information in order to learn the shape specific of a class of 2-D objects [143, 31, 124]. A drawback of these methods is that they need an initialization of the model at a position and orientation close to that of the corresponding target object in the image. They also have a low convergence rate, and hence a high computational cost.

A method to find various instances of a 2-D shape in an image, without the need of initializing the model at a certain position and orientation, was presented by Liu and Sclaroff [96]. Besides using top-down statistical 2-D shape templates, they also exploit bottom-up image information, such as that obtained by a standard region segmentation algorithm. Their approach performs merges and splits of regions in search of region groups with shapes similar to a model, which is learned in an offline interactive process from a set of training images. From a bottom-up image region segmentation, obtained using some homogeneity predicate (e.g., color or texture), the space of region merges and splits is explored, aiming to minimize a cost function that takes into account the agreement between the region grouping and the shape template, the amount of template deformation, the predicate compatibility within the region grouping, and the number of region groupings.

They show experimental results where the robustness of the method to occluders and shadows is demonstrated. However, results on real images are presented not for complex shapes, such as hands, like in the previously cited works, but only for rather simple shapes (e.g., fruits, leaves, and blood cells) in which no background clutter is present. It must be noticed that the grouping of regions is not driven solely by shape, but this approach relies on the not always true assumption that adjacent regions that correspond to the object have similar homogeneity predicate values, i.e., similar color or texture. The approach heavily relies on this assumption to make the method tractable. By constraining the
possible merges to regions having similar appearance, the size of the space of possible merges is considerably pruned, thus avoiding a combinatorial explosion.

Wang et al. [176] proposed a method that similarly, starting from an oversegmented image, attempts to find region groups accounting for the 2-D shape of a particular model. Instead of minimizing an objective function, they used a stochastic approach to explore the space of region merges and splits in search of region groups with the particular shape. The object shape is modeled by a multiscale curvature representation obtained from precomputing the object’s boundary curvatures at different scales. Appearance information can optionally be incorporated into the model, as well. From a bottom-up image region segmentation obtained using a standard method (mean-shift [28] in their experiments), their approach is capable of finding multiple occluded instances of the model shape by grouping oversegmented regions. Given a shape prior \( S \) and the image data \( I \) (i.e., appearance), and assuming independence between region groups, they find the segmentation \( W \), namely, a disjoint set of image region groups that maximizes \( p(W|I,S) \), employing Markov Chain Monte Carlo sampling. At each sampling step, an attempt is made to extend partial matches between groups of regions and the model shape at all scales.

Although results for synthetic images are shown for a star shape prior, experiments on real imagery are only presented for images of leaves using a simple ellipse shape prior, similar to the case of Liu and Sclaroff. However, no experiments were performed on images from Liu and Sclaroff’s dataset or comparison made with their results, as it would have been natural to expect. Despite the fact that the use of appearance is claimed to be optional in this approach, all experiments modeled appearance via color histograms. It would have been desirable to include an experiment showing segmentation results obtained for an image with and without the use of appearance, to show how much the method actually relies on such information.

Different levels of human supervision were required for the generation of the models
in the previously cited methods. Alternatively, the following approach learns the models from a set of training examples, without the need of human intervention. The approach of Borenstein and Ullman [16] attempts the figure/ground segmentation of a known object in a cluttered image from the overlap between automatically extracted object fragments. No labeling of training images is needed, only a bit per image indicating if they belong to the class of interest. Training images are segmented into regions using a standard bottom-up approach. A set of shape primitives defining common object parts is extracted from the region segmentation in the form of 2-D rectangular fragments. Such a set is generated by first generating a large set of fragments and later selecting from it a subset that is highly informative, highly overlapping, and well-distributed across all images, such that the object in all images is likely to be completely covered by templates in the subset. The selection is performed such that the mutual information between the subset and the class is maximized. A figure/ground labeling of the fragment regions is subsequently learned based on the assumption that in a set of fragments carrying high mutual information with the class, the fragments’ regions corresponding to the object are covered by significantly more fragments than background regions. The object model is then formed by such a fragment set, capturing the local shape of small portions of the object’s boundary, as well as some internal local shape.

Segmentation of a novel image is accomplished by detecting the presence of the model fragments in the image and having the labeled regions of the fragment vote for the figure/ground labeling of the pixels falling below them. The final consensus is computed after removing inconsistent fragments that vote against the majority. Results are shown only for three classes of objects and with no significant changes in pose: horse heads, human faces and profile view of cars. No examples are shown of objects with a high contour-length / object-area ratio or containing a significant number of concavities, such as whole horses or human bodies, where the number of fragments having high mutual information with the class would tend to be considerably larger than in the case of the
Given the locality of the shape representation, it seems that the approach will have problems modeling such objects, since the proliferation of local shape primitives would naturally produce high ambiguity in the labeling of fragment regions, e.g., identical shapes having contradicting figure/ground labels, as they come from different parts of the object boundary. This would make it difficult to determine a consistent labeling of fragment regions, rendering the approach less effective, as it relies on the power of such local labellings to achieve segmentation. Also, avoiding such examples in their test set can be explained by the fact that objects with such characteristics tend to be non-rigid and hence are prone to changes in pose, which the current approach seems incapable of handling, since they are not modeled anywhere in the formulation. Finally, since shape is modeled locally, one would expect the approach to be able to cope with some occlusion, however; no examples of such have been provided.

Borenstein and Malik [15] proposed a Bayesian formulation of the same figure/ground segmentation problem tackled by Borenstein and Ullman. In contrast to such a fully unsupervised method using automatically labeled image fragments as shape primitives, the present approach employs binary templates automatically learned from a collection of binary (figure/ground) training images representing class silhouettes. A multi-scale hierarchy of bottom-up region segmentations, modeled by a graph $G$, is used to construct a prior on the possible figure-ground segmentations of the image. They seek the figure/ground labeling $X^0$ of the regions at the finest label of the segmentation hierarchy that maximizes the probability $p(X^0|G, Y) = \frac{p(Y|X^0, G)p(X^0|G)}{p(Y|G)}$, where $Y$ is the top-down segmentation found to maximize $P(Y|G)$, computed in a way similar to the previous Borenstein and Ullman approach, but using the binary shape templates instead of image fragments.

$p(X^0|G)$ is computed taking into account the similarity (in terms of texture, average intensity and boundary properties) between regions at adjacent levels in the hierarchy,
such that there is a high prior for the uniform labeling of pixels that are strongly connected to a salient homogeneous segment at a coarser level. Since the final segmentation of the image is determined by the labeling of the regions at the finest label, \( p(Y|X^0, G) = p(Y|X^0) \). This is estimated assuming that pixel labels are independent given \( X^0 \), and defining \( p(Y_q|X^0_q) = E(|X^0_q - Y_q|) \), where \( Y_q \) and \( X^0_q \) are continuous labellings in \([0, 1]\) for pixel \( q \), and \( E(x) \) is an exponential distribution with support inside the interval \([0, 1]\) and parameter \( \lambda = 1 \).

Results are provided on images presenting a greater challenge (e.g., horses and runners) than those used in the previous approach of Borenstein and Ullman. The greater robustness of the current approach in its ability to deal with more complex shapes than its predecessor is likely to be the result of two important differences between these works. The current approach integrates bottom-up image information computed at various image scales (while the former method only exploits such information at a single scale). It also uses a set of human-labeled training images, thus avoiding the possibility of learning incorrect figure/ground labels, a weakness present in the previous approach due to its unsupervised nature.

In the test examples shown, it can be seen that this method is able to handle only small changes in pose and scale, but has trouble when changes are more significant, e.g., missing horse’s or runner’s legs when their positions are at an angle substantially different from the learned model. Such inability of the approach is expected, since neither local nor global shape deformations are modeled anywhere in their formulation. On the other hand, it seems reasonable that the approach should be capable of detecting the presence of two or more exemplars, if present in an image, given its formulation. However, in test images with more than one instance of the object class (e.g., images with two runners), the approach only segments the runner falling entirely inside the image. Like in the previous approach, there are no comments or examples regarding the ability of the approach to deal with occlusion. Finally, a major drawback of both Borenstein approaches is that
they are neither rotation nor scale invariant.

No global shape model is used in both cited works of Borenstein et al., only local shape primitives. There is then no guarantee that the cover achieved by those methods corresponds to an actual object and it is not simply the result of an arbitrary arrangement of background clutter resembling random object parts. Winn and Jojic [179] proposed an unsupervised Bayesian formulation of the same problem addressed by Borenstein, but also modeled global object shape. They formulate the problem using a hierarchical generative model that incorporates a top-down global shape model and bottom-up edge and color cues. A mask probability model captures the broad global shape of the object class and typical edge locations are modeled by an edge probability distribution. Short segmentation boundaries that are aligned with intensity gradients in the image are favored by introducing a MRF prior. Scaling and translation of the mask and edge distributions are also modeled and a deformation field distribution accounts for changes in pose and viewpoint.

A key assumption made by this approach concerns the appearance of figure and background: each have a low variability within an image, but their variation is very high between different images of the same class. To model such behavior, Probabilistic Index Maps (PIM) [79] are used to model the appearance of the object and of the background. Each image layer is defined as a matrix whose entries are discrete indices to a separate palette of possible intensities, colors or other features. An image class is defined not by a single index map, but by a probability distribution over the index maps.

Using together all images in a training set containing objects of the same class, a consistent segmentation is obtained for each image by inferring the common class model, and thus determining the location, segmentation and pose of the class object in each image. Object recognition can be performed via classification by means of computing the posterior probability of the inferred model for each class on the test image, and selecting the class achieving the highest probability value.
The results achieved by this unsupervised method are superior to those obtained by the unsupervised approach of Borenstein and Ullman, and even comparable to fully supervised methods. Since shape deformations are explicitly modeled within the framework, the approach is able to correctly segment more challenging images (e.g., presenting moderate changes in pose) than the ones used in the experiments of the supervised method of Borenstein and Malik. However, there is a limit to the tolerance in the pose changes that the approach can handle, with the deformation field being incapable of coping with parts that are too far from the neutral position. Moreover, the approach presents problems in learning a consistent outline when the objects cannot be aligned due to variations in position and size among the training images beyond what the translation and scale models can handle. The major drawbacks of the approach include its lack of rotation invariance, its constraint to only detect a single object instance per image, and its incapacity to deal with occlusion, given the global nature of the shape prior employed.

The Deformable Part Models (DPM) framework, recently popularized by Felzenszwalb et al. [49], has been successful in representing highly variable object classes. Objects are represented in terms of a “root” and parts, which are modeled by Histogram of Oriented Gradients (HOG) features. The root captures the global shape of the object, while the parts model the individual shape of relevant object parts. A quadratic deformation cost is associated to the displacement of the part in relation to the root in order to accommodate within class variations in relative part location. In [56], Fidler et al. extended DPMs to reason in 3-D by enclosing 3-D object models by tight oriented 3-D bounding boxes and sharing information between adjacent aspects, outperforming the state-of-the-art in object detection.

Block-world approaches have also been applied to the problem of category detection by training block extraction models for specific object categories. For example, Schlecht and Barnard [150] have presented an approach working on furniture, and Del Pero et al. [34, 128] recovered blocks in indoor scenes. The block dimensions and their geometric
relationships were used by these approaches to determine object category.

All the methods previously discussed make the very limiting assumption that they know what object (and in which particular pose) lies in the scene. In contrast, Bataille and Dickinson [7] presented a framework capable of doing recognition for a potentially large database of different object categories, if a multi-scale hierarchy of bottom-up image region segmentations, presenting a low amount of under- or over-segmentation at coarser levels, is available. The shock graph of the silhouette generated by the 2-D projection of a 3-D object has proven to be quite stable to changes in viewpoint. It offers some robustness to partial occlusion given its locality of representation, and is capable of capturing the coarse prototypical shape of an object class, making it an ideal tool for view-based class abstraction [156]. Based on such advantages, shock graphs were the model chosen by Bataille and Dickinson in their approach, in contrast to all methods discussed above, which modeled object contours by means of statistical priors.

Starting at the coarsest level of the hierarchy, their approach attempts to explore the space of region merges. The shape of a region group is used to find models having a partial shape similarity which, in turn, is used to prune the space of region merges. Initially, groups consist of single regions at the coarsest level, which are successively extended by including adjacent regions. For each new generated group, the shock graph of its silhouette is used to efficiently index into the object database [104] to find models with partially similar shapes. These candidate models are then used to guide the addition of adjacent regions to the group that improve the shape similarity between the group and the model. When the point is reached that no improvement is possible by the addition of adjacent regions at the current level, the process descends in the hierarchy and attempts the addition of regions adjacent to the footprint of the group at the next finer level. The process continues until a certain global shape similarity threshold is achieved, or no further improvements are possible.

This approach is based on the assumption that part of the shape of at least one of
the figure regions at a coarse level significantly resembles a part of the corresponding model shape. Moreover, this sub-shape similarity needs to be captured by a subgraph (common to the region’s and model’s shock graphs) accounting for a sufficient portion of the model shape such that the indexing mechanism is capable of recovering the correct model as a candidate. Since shock graphs capture global shape, changes to a silhouette’s boundary (such as those caused by region over- or under- segmentation) are prone to produce graphs with considerably different topology, making such an assumption a very brittle one. A common case in which this is violated is when objects are composed of various textures or colors (or differ in whatever other predicate used for the region segmentation), thus leading to bottom-up segmentations in which none of the regions resemble the shape of a significant portion of the object’s model.

Limitations, Weaknesses, Open Problems

Except for the naive approach of Bataille and Dickinson, which attempts to use bottom-up image information to index into a database of object models, all other previously cited strategies make the assumption that they know what object lies in the scene. This effectively amounts to a target recognition task, which is a very restricted form of object recognition. Another important drawback of these approaches is the view-based nature combined with the fact that their models account for entire objects, which requires the storage of a different model for each topologically distinct viewpoint (i.e., aspect) of each object. Thus, target recognition does not scale up to general segmentation or recognition when the contents of the image are unknown, and it has long been known that a linear search of a large database of models is intractable [134].
2.2 Conclusions

Shape abstraction, i.e., extracting the qualitative shape of an object in terms of its parts, is an important but still unsolved problem. Most categorical systems don’t try to pull out the abstract parts of an object. Instead, they use an object model (learned from training images) to constrain segmentation and grouping, which amounts to a target recognition task. This does not scale up to general segmentation or recognition when the contents of the image are unknown. In the absence of object-specific knowledge, we must rely on perceptual grouping to identify sets of low-level image features that are unlikely to co-occur by chance. Such perceptual grouping relies on mid-level priors that are domain-independent, and reflect the regularities of our world (and their projections on our retinas). But before such non-accidental groups can effectively support object categorization, they must be abstracted or simplified. By recovering an intermediate granularity of shape, in terms of a small set of part models, effective shape indices can be extracted to efficiently invoke a model for top-down verification. The set of shape priors should therefore be finite, small, and independent of the number of different objects in the database, thus providing a modeling framework which is invariant to part articulation and viewpoint change, and offers the locality of representation required for recovering the visible parts of occluded objects.
Chapter 3

An Edge-Based Approach to Part Hypothesis Generation

In this chapter, we present a framework for recovering a set of abstract parts from a multi-scale contour image, corresponding to the initial progress that we achieved on this important problem. In this approach, we introduced a novel indexing mechanism to generate abstract part hypotheses given a collection of contour fragments. Given a user-specified part vocabulary and an image to be analyzed, the system covers the image with abstract part models drawn from the vocabulary. More importantly, correspondence between image contours and part contours is many-to-one, yielding a powerful shape abstraction mechanism. Figure 3.1 shows an input image (a) and its corresponding edge map (b) at a particular scale. In (c), we see an example input part vocabulary, and (d) shows the final covering of the image by hypotheses selected by the approach in Section 3.3 from the set of hypothesis generated by our proposed approach using parts drawn from the vocabulary. The selected abstract parts of the rice cooker reflect its coarse cylindrical structure.
Figure 3.1: Recovering abstract shape parts from an image: (a) input image; (b) extracted contours; (c) a sample part vocabulary; and (d) the covering of parts (drawn from the vocabulary) after part hypothesis selection.

3.1 Overview of the Approach

Our first approach to qualitative, part-based shape abstraction takes, as input, a 2-D image and a vocabulary of 2-D part models. The input image is processed to yield a hierarchy of edge maps at different scales. The part models can be seen as closed contour templates, and are meant to coarsely sample the projections of the surfaces that comprise a vocabulary of qualitative 3-D parts. Our 2-D vocabulary must sample the variability of projected 2-D shape due to a 3-D part’s rotation in depth as well as any within-class
part deformations, such as bending, tapering, shearing, etc. Rotation and image scale of the part vocabulary do not need to be sampled, for they are handled by rotation and scaling of the input image. For the experiments reported in this paper, our 2-D shape vocabulary consists of 25 part shapes (shown in Figure 3.1(c)), sampled from the family of superellipses with bending and shearing deformations.

Our approach simultaneously proceeds both top-down and bottom-up. In a classical top-down step, a fixed-size search window will be placed at all scales, locations, and rotations in an attempt to detect the presence of an abstract 2-D part drawn from the vocabulary. However, unlike today’s object detectors, which are typically object-based, we introduce a bottom-up step, in which local edge evidence falling in the search window is used to index into the space of part templates consistent with the local evidence. In an even more critical departure from today’s contour-based detectors, which assume that extracted image contours map one-to-one to model contours, we introduce a critical shape abstraction mechanism that allows a collection of disconnected contour fragments to map many-to-one to an abstract model contour. This is essential to support abstract part decomposition of images of real objects, in which the granularity of abstract model contours may lie far above the granularity of the extracted image contours.

When similar hypotheses compete for the same image evidence (in highly overlapping search windows), a non-maximum suppression step discards all but the model achieving the best fit. A final model selection step completes the procedure, in which from the set of redundant (and even contradicting) hypotheses, the subset achieving the best covering is selected. The result is a segmentation of an image into a set of 2-D part abstractions drawn from a vocabulary whose size is fixed and independent of the (possibly infinite) number of objects whose projections can be abstractly modeled as configurations of 2-D parts drawn from the vocabulary. Since different object domains may require different vocabularies of shape models, it is important to note that the vocabulary is an input; the method does not depend on a particular vocabulary.
3.2 Part Hypothesis Generation

Figure 3.2 presents our algorithm (pseudocode) for generating abstract part hypotheses from extracted image contours. The algorithm takes, as input, a multi-scale edge map $H$ (computed from the input image) and a set $V$ of qualitative 2-D shapes, and outputs a set $R$ of model hypotheses supported by sufficient evidence from the input edge data. In an off-line step, we first compute the *model grid* $G$, a spatial data structure used to measure distances between observed image edgels and abstract model contours, facilitating the bottom-up indexing step that hypothesizes part models in a search window, and supporting the many-to-one mapping of contours essential for part abstraction.

The on-line procedure effectively places the model grid in the image at all positions, scales (in x and y), and orientations, and votes for models at each grid placement. A discrete set of rotations of the multi-scale edge map $H$ is computed, allowing the orientation of the model grid to be fixed. A loop over all model sizes $s_x$ and $s_y$ follows, in which the relevant level $l$ of the multi-scale edge map is selected as a function of model size. By anisotropically subsampling this level $(l)$, we can effectively vary the length and width of the model grid while keeping its size fixed. This helps manage search complexity by ensuring that search windows never become large, i.e., a large window at a finer granularity can be approximated by a small window at a coarser granularity. The model grid cell size, `cell_size`, in terms of pixels at the original image resolution, is a function of both the level $l$ as well as the degree of anisotropy.

For each model size, we iterate over all rotation angles $\theta$, and for each angle, a resampled version $e$ of the edge map at level $l$ and orientation $\theta$ is computed. Before the area of support corresponding to each translated (by $(t_x, t_y)$ pixels) model grid is used to generate bottom-up part hypotheses, it is first screened to see if enough edge activity lies in the window. If enough edge activity is found, by examining the integral image $I$ in the window, the model grid $G$ is overlaid on the resampled edge map $e$ so that image edgels can vote for part models encoded in $G$. Finally, all model hypotheses having a
Chapter 3. An Edge-Based Approach to Part Hypothesis Generation 62

**Input:** an edge map hierarchy \( \{H_1, \ldots, H_L\} \), a qualitative part vocabulary \( V \)

**Output:** a set \( R \) containing the recovered models

1: Generate model grid \( G \) containing all models in \( V \) \{(Section 3.2.1)\}

2: Compute set \( E_{l,\theta} \) of all rotations of \( H_l \), \( l = 1, \ldots, L \) \{(Section 3.2.2)\}

3: \( R = \emptyset \)

4: \( s_x = \min_s \) \{(Section 3.2.3)\}

5: while \( s_x \leq \max_s \) do

6: \( s_y = \min_s \)

7: while \( s_y \leq \max_s \) do

8: Compute the relevant level \( l \) of the edge map hierarchy

9: Compute cell size

10: for \( \theta = 0 \) to \( 2\pi \) step \( d\theta(s_x, s_y) \) do

11: \( e = \text{resample}(E_{l,\theta}, \text{cell size}) \) \{(Section 3.2.4)\}

12: \( I = \text{integrate}(e) \)

13: for all \( t_x \), step pixels_per_translation_step do

14: for all \( t_y \), step pixels_per_translation_step do

15: \( \text{model_present} = \text{screen}(I, t_x, t_y) \) \{(Section 3.2.5)\}

16: if \( \text{model_present} \) then

17: \( M = \text{index_models}(e, t_x, t_y, G) \) \{(Section 3.2.6)\}

18: \( R = R \cup \{m \in M : \text{percent_covered}(m) \geq \tau\} \) \{(Section 3.2.7)\}

19: end if

20: end for

21: end for

22: end for

23: \( s_y = \text{s_ratio} \ast s_y \)

24: end while

25: \( s_x = \text{s_ratio} \ast s_x \)

26: end while

Figure 3.2: Part Hypothesis Generation (components of the algorithm are detailed in the designated subsections)

minimum percentage \( \tau \) of their contour accounted for by image evidence are added to the output set \( R \). In the sections below, we explore these steps in more detail.
3.2.1 The Model Grid

The model grid $G$ has fixed size and resolution, and represents a unit square (i.e., \([-0.5, 0.5] \times [-0.5, 0.5]\)) containing axis-aligned instances of all models in the part vocabulary, scaled to fit in the square. Each model is represented as a discrete set of roughly similar length, linear model contour fragments. (In our implementation, each model is made up of 40 contour fragments.) Each model also defines a scale-independent abstraction tolerance band along its model contour fragments, such that any image edgel falling within this band is considered to be consistent with the model (note that the grid is extended in size to encompass the band). Scale independence of the band is achieved by making its size a function of the lengths of the model axes. Finally, each cell in the grid contains a list of model contour fragments belonging to any model whose abstraction tolerance band encompasses the center of the cell. Associated with each model contour fragment in this list is its normal orientation and the minimum Euclidean distance between the fragment and the center of the cell. Figure 3.3 illustrates the model grid and the abstraction tolerance bands for two models.

3.2.2 Rotation Invariance

Rotation invariance is achieved by fixing the model grid orientation and rotating the image. Given a finite set of angles, a rotation $E_{l, \theta}$ of each level $l$ in the edge map hierarchy is computed (once, at initialization) for each angle $\theta$ in the set. If a given model is rotated through $\theta$, a normalized alignment error can be computed as the ratio of the integral of the Euclidean distance between corresponding image and model contours to the model’s perimeter. In general, this error increases as the shape’s aspect ratio deviates from unity. Therefore, to guarantee that this (normalized) error remains bounded and independent of the model size and aspect ratio, the rotation step size is dependent on the model’s aspect ratio.
Figure 3.3: The model grid is a data structure that maps a grid cell to those models whose abstract contour, including a tolerance band around the contour, intersects the cell. An image edgel will vote for all models whose abstraction tolerance bands intersect the cell containing the edgel. Votes are weighted according to relative distance and orientation with respect to the nearest model contour fragment. Two models populate the model grid shown in this example.

Specifically, the rotation angle step can be computed as a function $d\theta(s_x, s_y)$ of a parameter $\psi$, which is the ratio of the maximum error between two corresponding model points to the length of the smallest axis. Formally, let $s_{\text{max}} = \max(s_x, s_y)$ and $s_{\text{min}} = \min(s_x, s_y)$. $\psi$ specifies the distance between the point $(s_{\text{max}}, 0)$ and its position following rotation by an angle $d\theta$, normalized by $s_{\text{min}}$, i.e.,

$$\psi = \| (\cos(d\theta) - 1, \sin(d\theta)) \| \frac{s_{\text{max}}}{s_{\text{min}}} \tag{3.1}$$

It follows that

$$d\theta = \arccos(1 - (\psi \frac{s_{\text{min}}}{s_{\text{max}}})^2/2). \tag{3.2}$$

### 3.2.3 Scale and Translation Invariance

We achieve scale invariance by fixing the model grid size and moving it through different scales of the image, with the ratio $(s_{\text{ratio}})$ between adjacent scales held constant. The number of scales that are generated is a function of the minimum and maximum model
sizes (in terms of pixels at the original image resolution) \( \min_s \) and \( \max_s \). Translation invariance is achieved by translating the model grid, where the translation step size is a function of the model size which, in turn, implies that it is constant for all scales. Edge maps are resampled so as to ensure that this constant translation step size is an integer number of pixels.

Specifically, let \( t \) be the fraction of the model size that corresponds to a translation step, and let \( a \) be the fraction of the model size that corresponds to the abstraction tolerance. If \( \frac{t}{(1+2a)} \) is a rational number, then given a minimum dimension of \( K \) cells for the model grid, it is always possible to resample the edge maps such that: 1) both the model grid size (dimension) \( \tilde{K} \) and model translation step correspond to an integer number of pixels; and 2) \( \tilde{K} \geq K \). Let \( n \) and \( d \) be integers such that \( \frac{n}{d} = \frac{t}{(1+2a)} \) and \( \text{gcd}(n,d) = 1 \). Then \( \tilde{K} = [K/d] \cdot d \) is the minimum value for the model grid size satisfying the above two constraints. The corresponding number of pixels per translation step is \( \text{pixels\_per\_translation\_step} = [K/d] \cdot n \).

### 3.2.4 Edge Map Resampling

To detect anisotropic scalings of a model, we anisotropically scale the isotropic multi-scale edge map. For each rotation angle in the set of angles corresponding to a particular model size, the corresponding rotated edge map at a certain level of the hierarchy is anisotropically resampled, with the level chosen based on the size \((s_x, s_y)\) of the model. The level whose pixel size (in terms of input image pixels) is closest to \( \min(\sigma_x, \sigma_y) \) is chosen, since that is the edge map whose resolution matches that of the smaller (finer granularity) dimension of the scaled model. In our experiments, we use a hierarchy of edge maps with as many levels as sampled model sizes.

Specifically, the function \( \text{resample}(\eta, \sigma) \) resamples an edge map \( \eta \) for a model grid cell size \( \sigma = (\sigma_x, \sigma_y) \). An edgel \( p \) in the resampled edge map is considered active if there is at least one active pixel in \( \eta \) that is resampled to \( p \). The resampled location of a pixel
Chapter 3. An Edge-Based Approach to Part Hypothesis Generation

$(q_x, q_y)$ from $\eta$ is computed as $([q_x/\sigma_x], [q_y/\sigma_y])$. The $\sigma$ used for resampling is computed as a function of model size, the ratio between the resolution of the original image and that of the edge map at the current level of the hierarchy, and the value of the translation step. Cell size is computed as $(\sigma_x, \sigma_y) = (dt_x, dt_y)/\text{pixels per translation step}$, where the translation steps $dt_x$ and $dt_y$ are computed as $(dt_x, dt_y) = t \cdot (s'_x, s'_y)$. $s'_x$ and $s'_y$ are the model sizes $s_x$ and $s_y$ after rescaling them according to the ratio of the resolution of the edge map $e$ to that of the original image.

### 3.2.5 Model Screening

The process of hypothesizing model parts contained in the search window is computationally expensive, and should only be applied in windows where a part may exist. Thus, a fast screening operation is performed on the contents of the search window to identify search windows which are highly unlikely to contain a part. This process can be thought of as a visual attention or saliency detection mechanism, aimed at focusing computational resources on the most pertinent image regions from a scene [44, 3]. Such a fast screening process is based on edge activity and how it is spatially distributed in the window, i.e. checking that there is a minimum amount of edge activity that is distributed in the four quadrants of the window. Edge activity is of interest because abstract part boundaries map to edgel evidence. The focus on the non-local distribution of the edges is motivated by the nature of the spatial distribution of the very model contours in the model grid, deriving from the fact that part models are compact shapes and their size is roughly that of the model grid.

The process can be efficiently computed using integral images (or summed-area tables [33]). The integral $I$ of an edge map $e$ contains at position $(i, j)$ the sum of active pixels with height $i$ and width $j$, i.e.,

$$I(i, j) = \sum_{i' \leq i} \sum_{j' \leq j} e(i', j'). \quad (3.3)$$
Chapter 3. An Edge-Based Approach to Part Hypothesis Generation

$I$ can be computed recursively by the formula:

$$I(i,j) = \begin{cases} 
I(i, j - 1) + I(i - 1, j) + e(i, j) - I(i - 1, j - 1), & i, j \geq 0 \\
0, & \text{otherwise}
\end{cases}$$

(3.4)

The number of active pixels in a rectangular window (at position $(i,j)$ with height and width $(h,w)$) in edge map $e$ can be efficiently computed using four references to $I$ as

$$I(i,j) + I(i+h,j+w) - I(i,j+w) - I(i+h,j).$$

(3.5)

Each quadrant can be computed in this manner. A count of the number of quadrants whose edge activity exceeds a threshold reflects the degree to which there is coherent edge activity distributed about the center.

3.2.6 Model Indexing

Hypothesizing models in a search window $w \subset e$ at a particular position $(t_x, t_y)$ of the resampled edge map $e$ is performed by a process equivalent to overlaying the model grid $G$ on the window and having each active edge pixel $p \in w$ vote for (or index) those models in $G$ having a model contour fragment whose abstraction tolerance band encompasses $p$ (see Figure 3.3). A voting table $T$ is used to keep track of the evidence supporting each model contour fragment. Specifically, entry $(m,f) \in T$ accumulates evidence in support of model contour fragment $f$ belonging to model $m$ according to the edgel activity falling within $f$'s abstraction tolerance band. The voting process proceeds as follows. An edgel at position $(i,j)$ in the search window lies within the abstraction tolerance bands of all model contour fragments in $G(i,j)$. For each of these candidate model contour fragments, their corresponding entry in $T$ is updated according to the scoring function below. After all edgels in the window have been processed, those models (i.e., rows in $T$) with an insufficient number of model contour fragments receiving non-zero evidence are discarded.
3.2.7 Model Scoring

A model score reflects how well the image evidence accounts for a particular model at a certain position, orientation and scale. Abstraction is implemented in this calculation by taking into consideration all image edgels falling within the model’s abstraction tolerance band. The contribution to the model score by each such image edgel depends on three intuitive components common to classical work in chamfer- or Hausdorff-based target recognition (e.g., [121, 17]):

1. **distance** As the distance between an edgel and the nearest model contour fragment increases, the edgel’s contribution to the model score decreases.

2. **orientation** As the difference in orientation between the edgel’s normal and that of the nearest model contour fragment increases, the edgel’s contribution to the model score decreases.

3. **continuity** Edgels that form longer, more salient contours should contribute more to the model score than spurious, disconnected edgels that may be due to noise.

For continuity, rather than performing a full connected components analysis to determine the length of the contour that subsumes each edgel, we instead compute a local measure of continuity of all image edgel evidence in support of a model contour fragment. Specifically, let $C_f$ be the set of image edgels within the abstraction tolerance band that are closer to $f$ than to any other model contour fragment, let $N_f$ be the total number of edgels in the edge map (at original resolution) that were resampled to edgels in $C_f$, and let $E_f$ be the number of these edgels that are endpoints. For model contour fragment $f$, the ratio $\frac{E_f}{N_f}$ reflects how disconnected the edgel support for the model contour fragment is.
Combining these measures yields the following equation to compute the score of a model contour fragment $f$:

$$S_f = \left(1 - \frac{E_f}{N_f}\right) \frac{1}{|C_f|} \sum_{c \in C_f} g(d_c)h(\gamma_c),$$  \hspace{1cm} (3.6)

where $d_c$ is the distance from edgel $c$ to the fragment $f$, $\gamma_c$ is the difference in normal orientation between $c$ and $f$, and $g$ and $h$ are symmetric weight functions achieving a maximum in 0, and decaying away from it. The division by $C_f$ makes the measure independent of the number of image edgels contributing to $f$. If $C_f$ is empty, $S_f$ is defined as 0. In our implementation, we obtained good results using a function of exponential decay for $g$, namely, $g(x) = e^{-\frac{x^2}{a^2}}$, where $a$ is the fraction of the model size that corresponds to the abstraction tolerance, and $h(x) = \cos^2(x)$. Thus, $g$ can be interpreted as an un-normalized Gaussian (i.e., achieving a maximum value of 1 at its mean) with mean 0 and standard deviation $a/2$. With this choice for $g$, edgels close to the model contour will have the maximum contribution to the score, with such contribution fading away as the edgel gets further from the model, and virtually extinguishing when the edgel is beyond the abstraction tolerance band. Analogously, our choice of $h$ ensures that an edgel’s contribution to the score diminishes as its angle with the model contour increases, having its minimum when the edgel and model contour are orthogonal.

Having defined the score for a given model contour fragment, we can now define the score of an entire model. In addition to summing the scores of its component fragments, we would like the overall score to reflect the spatial coherence of the local evidence. We therefore reward spatially coherent sets of consecutive model contour fragments supported by the image evidence. Specifically, we compute the final model score as the sum of augmented fragment scores, where the augmented score $\tilde{S}_f$ of a fragment is 0 if $S_f = 0$; otherwise, it is the convolution with a Gaussian filter $G$ of the scores of the fragments in a neighborhood centered on $f$. (In our implementation we used a 5-tap filter.) Thus, the more fragments in $f$’s neighborhood that are accounted for by image evidence, the
larger $\tilde{S}_f$ is. That is the case because the convolution with the filter ensures that each model fragment supported by image evidence will contribute to the score multiple times, as many times as near-by supported model fragments are, with each contribution having a weight proportional to the distance to the near-by model fragment. Therefore, the more clusters of close-by supported model fragments there are, the higher the final score will be. Formally, let $f_0, \ldots, f_{t-1}$ be the list of model contour fragments making up the model’s contour, sorted by their position along the contour, and let $S_i$ be the score of fragment $f_i$. The total score of the model is computed as:

$$S = \frac{1}{t} \sum_{i=0}^{t-1} [S_i \neq 0] \sum_{r=-d}^{d} G(r) S_{(i+r) \mod t}$$

(3.7)

### 3.3 Part Hypothesis Selection

Part hypothesis generation yields both redundant and competing hypotheses, from which a final selection must be made. Our selection strategy, as illustrated in Figure 3.4, begins with a local, non-maximum suppression step, focusing on redundant hypotheses, i.e., models with similar shape that account for the same image evidence. Specifically, only that model with the best score is kept whenever two or more models from the same shape class are detected in highly overlapping windows.

The second phase of our selection strategy is global, selecting the smallest subset of maximally-sized, highest-scoring part hypotheses that covers the object surfaces visible in the image. The proposed selection scheme is depicted in the example of Figure 3.5. Hypotheses are selected according to a preference measure for pairs of hypotheses that favors hypotheses proximity, good boundary and junction alignment, and penalizes overlap.

We adopt an optimization formulation in the spirit of Pentland [127], whose goal was to maximize the savings in the encoding length of describing a binary image with a subset of binary mask hypotheses. However, unlike [127], in which explicit region
information was available, we have only boundary data (contours) to work with. We therefore introduce a hybrid strategy in which the boundary-based hypothesis score is used to weight the area of the hypothesis, favoring the selection of larger hypotheses, but penalizing them if they lack strong boundary support.

Interaction between pairs of hypotheses also needs to be treated differently. Whereas [127] measured both the agreement and disagreement in image region data where two hypotheses intersect, we have no explicit surface data to work with. We therefore compute the cost of encoding the overlap (area of intersection), representing the cost of encoding the fact that two surfaces cannot occupy the same region in the image. This clearly penalizes overlapping part hypotheses which, as we shall see in Section 3.4, is well-motivated.
(and effective) only in the absence of significant occlusion or self-occlusion. Finally, we introduce a part compatibility term that reflects the degree to which two part hypotheses fit together well, measured in terms of the area of intersection of their abstraction tolerance bands. While the resulting hybrid objective function is not technically a description length, we will adopt the term “savings” to refer to the benefit of selecting one or more hypotheses.

Let $H$ be our set of part hypotheses that survive the non-maximum suppression (phase 1 selection). The savings in encoding the entire image (in terms of individual pixels) using part hypothesis $h_i$ is:

$$S(h_i) = k_1 a_{ii} - k_2 e_{ii} - k_3,$$

where $a_{ii}$ is the part hypothesis' score ($S_{h_i}$) scaled by its area, and $e_{ii} = 0$. $k_1$, $k_2$, and $k_3$ can be interpreted as encoding costs for a pixel’s area, for an incorrect pixel’s area, and for the hypothesis, respectively.

The global solution is found by searching the power set of $H$ to find the subset of part
hypotheses that maximizes savings. If $x$ is the binary vector whose length is the number of hypotheses, then a subset of hypotheses can be encoded by setting those elements in the subset (and clearing the others). In this case, our objective function becomes:

$$S(x) = k_1 x A x^T - k_2 x E x^T - k_3 x x^T$$

(3.9)

where $a_{ij} = a_{ji}, i \neq j$, is a function of the intersection of the abstraction tolerance bands of parts $i$ and $j$, and $e_{ij} = e_{ji}, i \neq j$ is (half) the area of intersection of parts $i$ and $j$. We seek the subset of part hypotheses that maximizes this savings, and employ a standard quadratic programming optimization procedure to solve the problem.

### 3.4 Demonstration

We demonstrate our approach on some anecdotal images to help illuminate both the strengths and weaknesses of this framework. Figures 3.6 and 3.7 illustrate the results of our framework applied to the examples shown in the first row. Rows two through five contain the edge maps, the top 200 (scoring) hypotheses prior to model selection, the ground truth solution manually chosen from the generated hypotheses, and finally the solution as selected by our system. The ground truth, representing a subset of the generated hypotheses, clearly reflects the ability of our framework to generate model-based shape abstractions of noisy, disconnected contour data without assuming a one-to-one correspondence between extracted image contours and abstract model contours.

In (a), our search strategy converges on a plausible set of abstract part surfaces, including correct parts for the apple and can surfaces as well as four out of the five surfaces of the slot machine. In (b), the correct part abstraction has been fit to the top of the cup, the cup’s body has been slightly oversegmented, and an entire surface has been recovered for the silhouette of the cup’s handle. This larger hypothesis is preferred over the two handle hypotheses shown in the ground truth, where the handle has been oversegmented due to the fact that it cannot be completely covered by a single part from
Figure 3.6: Abstract Part Recovery (see text for discussion).
Figure 3.7: Abstract Part Recovery (cont’d - see text for discussion).
Chapter 3. An Edge-Based Approach to Part Hypothesis Generation

the vocabulary.

In (c), the two abstract surfaces of the jar are recovered, but slight problems exist on the oatmeal bag. One of the bag’s surfaces has been oversegmented, while the end of the bag has been fit with an elliptical part instead of a parallelogram. This is understandable, for the ground truth clearly reflects the human selector’s bias toward the block-like regularization despite the elliptical surface evidence in the original image. Both these types of errors (oversegmentation and misidentification) can be easily corrected when the identities of nearby hypotheses are taken into account. Recall that other than generic boundary overlap, explicit part interactions are not modeled. We will return to this in Chapter 6.

Clearly, our objective function needs further improvement to strike an optimal balance between hypothesis score, overlap, size, compatibility, and cardinality. This is a challenging task whose goals can be in conflict. For example, in (d-e), the overlap penalty has prevented us from recovering the cup’s body and the hat’s body, respectively, instead favoring the larger saucer and brim hypotheses. In such cases, we need to better reason about occlusion, and not penalize the overlap when it can be explained by occlusion. In (f), we see that some of the ground truth parts (desk objects) have been undersegmented while others oversegmented. Still, it is important to note that only a small set of correctly recovered parts may be necessary to invoke stronger top-down models with which to disambiguate competing hypotheses and to guide hypothesis selection; perfect, bottom-up part segmentation is not a realizable goal.

3.5 Conclusions

Unexpected object recognition requires the recovery of generic parts and their relations to support effective indexing into large databases. While contours may reflect important shape information, a single image contour or fragment may not be generic to a category,
and assuming one-to-one correspondence with a model contour can be highly restrictive. However, a collection of local contours may reflect a more abstract regularity that may be shared by many categories. Such abstract parts require not only that a noisy, broken collection is grouped, but also abstracted.

We have described a model-based framework for such grouping/abstraction that combines a mid-level shape prior in the form of a small (arbitrary) input vocabulary of part models (independent of the number of objects that can be constructed from the vocabulary) with a bottom-up part indexing framework that maps contour collections to abstract part models. The hypothesis selection model presented in this chapter is, however, rather simple, and serves mostly to reflect the availability of good hypotheses among those generated by the proposed framework.

There are some important caveats of the presented contour-based part hypothesis generation method. The first problem with this approach is its very high computational cost. This is due to its brute-force sliding-window-like nature, in which not only all locations and scales require an exhaustive exploration, but also all window rotations and aspect ratios as well as each relevant model deformation are to be taken into account.\footnote{Due to the full independence between the hypothesis generation processes at each window, the presented approach is completely parallelizable, being directly implementable in today’s ubiquitous SIMD architecture available in general purpose GPUs.} Another issue of the proposed method is that the number of generated model hypotheses at each window is often very high. This is because a large number of models from the vocabulary can generally be hallucinated from sets of disconnected nearby edges. Furthermore, that number grows very rapidly as the allowed degree of abstraction (i.e., width of the abstraction tolerance band) is increased.

In the next chapter, we introduce a data-driven region-based approach to part hypothesis generation. In contrast with the top-down method described in the present chapter, the proposed bottom-up approach enjoys a substantially reduced computational complexity as it has no need to explicitly explore combinations of different model positions,
sizes, rotations, or deformations.
Chapter 4

A Region-Based Approach to Part Hypothesis Generation

In this Chapter, we present a novel method to address the problem of contour-based perceptual grouping using a user-defined vocabulary of simple part models. Figure 4.1 shows an overview of our approach. We train a family of classifiers on the vocabulary, and apply them to a region oversegmentation of the input image to detect closed contours\(^1\) that are consistent with some shape in the vocabulary. Given such a set of consistent cycles, they are both abstracted and categorized through a novel application of an active shape model also trained on the vocabulary. From an image of a real object, our framework recovers the projections of the abstract surfaces that comprise an idealized model of the object.
4.1 Overview of the Approach

Our approach begins by computing a region oversegmentation (Figure 4.2(b)) of the input image (Figure 4.2(a)). The resulting region boundaries yield a *region boundary graph* (Figure 4.2(c)), in which nodes represent region boundary junctions where three or more regions meet, and edges represent the region boundaries between nodes (See Figure 4.3.); the region boundary graph is a multigraph, since there may be multiple edges between two nodes. Our approach can be formulated as finding simple cycles in the region boundary graph whose shape is consistent with one of the model shapes in the input vocabulary (Figure 4.2(d)); these are called *consistent cycles*. There is an exponential number of simple cycles in a planar graph [20], and simply enumerating all cycles (e.g., [170]) and comparing their shapes to the model shapes is intractable. Instead, we start from an initial set of starting edges and extend these paths, called *consistent paths* (or CPs), as long as their shapes are consistent with a part of *some* model. To determine whether a given path is consistent (and therefore extendable), we approximate the path at multiple

---

1 Someone could reasonably propose that instead of an approach that searches for closed contours (afflicted by occlusion and undersegmentation), 2-D part hypotheses could be generated directly from subsets of open contours. Such an approach, however, involves the combinatorics of dealing with such partial contour data, something that we want to avoid.
Figure 4.2: Problem Formulation: (a) input image; (b) region oversegmentation; (c) region boundary graph; (d) example vocabulary of shape models (used in our experiments); (e) example paths through the region boundary graph that are consistent (green) and inconsistent (red); (f) example detected cycles that are consistent with some model in the vocabulary; (g) abstractions of cycles consistent with some model; (h) example cycles inconsistent with all models.

scales with a set of polylines (piecewise linear approximations, providing a set of low-dimensional boundary abstractions), and classify each polyline using a one-class classifier trained on the set of training shapes (Figure 4.2(e)). When a consistent path is also a simple cycle, it is added to the set of output consistent cycles (Figure 4.2(f)).
Figure 4.2(d) shows the input vocabulary used in our experiments: four part classes (superellipses plus sheared, tapered, and bent rectangles, representing the rows) along with a few examples of their many within-class deformations (representing the columns). It is important to note that our approach is independent of the vocabulary of parts. While our demonstration vocabulary is ideally suited to the projected surfaces of ellipsoids, and straight and bent rectangular blocks and cylinders, representing a simple volumetric part vocabulary, the approach can accommodate any set of shapes, parameterized or otherwise. Each shape model is allowed to anisotropically scale in the x- and y-directions as well as rotate in the image plane. Since we employ scale-, rotation-, and translation-invariant features to train the classifiers, we need to generate only (approx.) 1,500 instances (by varying the aspect ratio and deformation parameters) belonging to these four shape classes. A single classifier is trained on all the component polylines of length (i.e., number of piecewise linear segments) $k$ spanning the complete set of shape models and their deformations. Therefore, if $K$ is the upper bound on the length of a polyline approximating a shape in the vocabulary, then $K$ classifiers are trained. An ideal vocabulary defines a small set of “building blocks” common to a large database of objects. As such, the complexity of the vocabulary shapes is low, and even at the finest
scale of polyline partitioning of a vocabulary shape’s contour, $K$ remains low; for our vocabulary, $K$ is 13.

The algorithm outputs cycles of contours that are consistent with one of the model (training) shapes. A cycle consists of actual contours (edges in the region boundary graph) in the image, and therefore does not explicitly capture the abstract shape of the contours. Moreover, the cycle has not yet been categorized according to the shapes in the vocabulary. To abstract (or regularize) the shape of a cycle and to categorize it, we follow a standard, iterative 2-step active shape model (ASM) fitting framework [29] trained on about 600,000 model instances, generated by varying their aspect ratio, orientation, and a finer sweeping of the deformation parameters than the one used to train the polyline classifiers. We iterate over the classical two-step ASM procedure, consecutively aligning and deforming the mean training shape with the cycle until convergence. However, we depart from a standard ASM framework in two key ways.

In a standard ASM framework, the training shapes belong to a single shape class, and the allowable, often limited deformations are typically captured (using PCA) in a low-dimensional shape space that can be approximated by a multidimensional Gaussian distribution. Moreover, at run time, the model must be properly initialized, for if the model is grossly misaligned, the deformations required to warp the model into the image may fall outside the space of allowable deformations. In our case, given a consistent cycle, we don’t know which category of vocabulary shape it belongs to, and hence which ASM model to apply (if we assumed one model per category in the vocabulary). Moreover, even if we knew its category, we assume no correct or near-correct initial landmark correspondence. We overcome the first problem by having a single ASM that’s trained on all instances of all the shapes in the vocabulary, and overcome the second problem by training on all possible cyclical landmark correspondences (alignments) across these shapes.

After ASM convergence, the training shape closest to the deformed model identifies
the category of the cycle. In the previous step, the consistent cycle classifier’s precision rate is never 100% at reasonable recall rates, and some of the recovered consistent cycles (of contours) may yield shapes that are qualitatively different from those in the vocabulary. Therefore, following ASM convergence, shapes that are still significantly different from the training shapes are discarded. Figure 4.2(g) illustrates the vocabulary shapes abstracted from the consistent cycles in Figure 4.2(f); for each detected shape, the algorithm also yields its shape category. Finally, Figure 4.2(h) illustrates some of the false positives discarded by the shape abstraction process.

### 4.2 Finding Consistent Cycles

In the following sections, we elaborate on the steps of our algorithm for finding consistent cycles, i.e., cycles whose shape is consistent with one of the model shapes; the two main steps of the algorithm are path initialization and path extension. Section 4.3 will focus on the problem of abstracting/categorizing the shape of the cycle.

#### 4.2.1 Path Initialization

The first step in the algorithm generates an initial set of single-edge paths that will be iteratively extended into cycles by repeated executions of the path extension step. This set of edges should be as least redundant as possible, to avoid generating the same cycle more than once (from different edges in the same cycle). Moreover, all possible graph cycles should be realizable by path extensions starting from edges in this set. Such an optimal set corresponds to the feedback edge set, which is the smallest set of edges whose deletion results in an acyclic graph. The feedback edge set is computed as the edge complement of a spanning tree. Favoring edges that represent longer, and thus more informative contours, we will choose as our initial edge set the edge complement of the minimum spanning tree, where edge weight equals contour length. While every cycle
contains at least one of these initial edges, two or more of these edges may be part of the same cycle. This is problematic, since extending these initial paths will yield the same cycles, which is highly inefficient. We avoid this problem by imposing a total ordering on the edges, and allowing a path to be extended only by an edge whose rank is greater than that of the initial edge of that path; the edges in the minimum spanning tree are all assigned a rank of $\infty$. The rank of a path is the rank of its initial edge. The set of initial edges, and their ranks, form our initial set of paths, and they are added to the queue of paths to be extended.

### 4.2.2 Path Extension

At each iteration of the algorithm, one of the paths is taken off the queue. If the path is a cycle, its consistency with the vocabulary of model shapes is checked. If it’s consistent with at least one shape in the vocabulary, it is added to the output list of consistent cycles. If, however, the path is not a cycle, its consistency is also checked. If the path is consistent with a portion of the boundary of at least one shape in the vocabulary, then the path’s possible extensions by an edge whose rank is greater than the path are added to the queue. The algorithm continues until the queue is empty, and outputs the consistent cycles.

Checking consistency begins by approximating the shape of the cycle or path with a
polyline computed at different scales using the Ramer-Douglas-Peucker algorithm [42]. For each resulting polyline, we compute a feature vector that encodes the angles and normalized lengths of the linear segments making up the polyline. As illustrated in Figure 4.4, the length of the feature vector is a function of the number of linear segments comprising the polyline. Each feature vector is passed to a one-class classifier (there is a classifier for each feature vector length) that determines if the feature vector is geometrically close to one of the training feature vectors. Those scales at which their corresponding polylines are consistent are associated with the path. If a path at a particular scale is not consistent, then no extension of that path can be consistent at that scale. Thus, when a path is initialized, it is associated with all scales, and when it is extended, its associated set of scales can only remain constant or be reduced to a subset of it. If there is no scale at which the path is consistent, the path is discarded, and will not be extended further. Figure 4.5 shows an example illustrating the steps involved in the path extension and consistency check processes.

### 4.2.3 Training the Classifiers

The feature vectors used to train the classifiers are generated from contour fragments of model instances. Axis-aligned instances of within-class deformations of each model are generated at varying aspect-ratios, and Gaussian noise is added to each generated contour with a standard deviation proportional to the model size (defined as the average distance from a model contour point to the model’s centroid). A number of equidistant points along each generated contour are sampled, and the two subcontours between every pair of such points (traversed in both directions) are used as training examples. A feature vector is generated for each subcontour from its polyline approximation, computed using a tolerance that is proportional to model size. Finally, the dimensionality of the feature vectors is reduced using PCA. Classification is performed on these reduced dimensionality vectors. For our model vocabulary, we observe that at least 99% of the variance is, in
Chapter 4. A Region-Based Approach to Part Hypothesis Generation

Figure 4.5: Consistent cycle detection (see text for discussion).

general, captured by the top $N$ PCA components for the case of feature vectors of dimension $2N - 1$, corresponding to polylines with $N$ linear segments.

In our implementation, the number of linear segments comprising the longest polyline approximating a model’s contour is bounded by 13. For this reason, at consistency check time, a path whose polyline approximation is longer than this value is discarded as inconsistent. Moreover, in the case of the path being a cycle, if its polyline approximation at a certain scale has less than three linear segments, the cycle is deemed inconsistent at that scale, since the scale is obviously too coarse. In the case of an open path with a polyline approximation with less than two segments, there is not enough information to
decide on the path’s consistency, since the path is evidently still too short at the given scale. In this case, the path is treated in the algorithm as if it had been consistent, leaving the decision to a future iteration, after it has been eventually extended to a length where a consistency decision is possible.

For our experiments, approximately 4 million contour fragments were employed to train the classifiers. Due to the difficulty in generating an adequate set of training examples of inconsistent model contour fragments, a one-class classifier was used instead of a binary classifier. Since the consistency check needs to be performed a large number of times (once per path extension), an efficient implementation calls for a method with a low classification complexity for this task. We obtain good classification performance and fast classification rates using a Nearest Neighbor Data Description approach\textsuperscript{2} [167] implemented via a fast approximate nearest neighbor search data structure [116].

\section*{4.3 Abstracting the Shape of a Consistent Cycle}

As mentioned in Section 4.1, we employ an ASM to both abstract the shape of a consistent cycle and to categorize it. Recall that we train a single ASM on all deformations of all vocabulary shape classes over all possible landmark correspondences. This avoids a proliferation of ASM models (one model, regardless of the size of the vocabulary), and allows the model to be initialized anywhere on the cycle. To train the Point Distribution Model (PDM), we sweep the parameter space of the shapes in the model vocabulary. Specifically, we generate contours for all models and degrees of deformation (i.e., bending, tapering, and shearing) at a dense set of discrete aspect ratios and all possible cyclical landmark alignments; in our implementation, we generate a total of approximately 600,000 such training cases. Rotation invariance is achieved by training on all model landmark alignments corresponding to all possible $N$ cyclical rotations of the landmark.
marks, while scale invariance is achieved during ASM fitting by the rigid transformation estimation.

The list of landmark points $\vec{m}_1, \ldots, \vec{m}_N \in \mathbb{R}^2$ in a training example corresponds to a fixed number of equidistant points sampled along the contour ($N = 64$ in our experiments). The first landmark $\vec{m}_1$ is the one for which the vector from the centroid $\vec{m}$ to the landmark has the lowest (absolute value) angle with respect to the x-axis, i.e., $(1, 0)^T \cdot (\vec{m}_i - \vec{m}) / \|\vec{m}_i - \vec{m}\|$ is maximum for $i = 1$; in case of a tie, we choose the point with maximum $\|\vec{m}_i - \vec{m}\|$. The indices of the other landmarks in the list keep their natural order along the contour. Following the standard ASM approach, a vector is formed for each contour by rasterizing the contour’s landmark coordinates. A PCA basis is computed for the training set, and the lowest-order principal components that capture most of the variance are chosen. For our training set, 99.9% of the total variance is captured by the top 21 components.

Fitting is performed by the successive iteration of two steps: one that finds the rigid transformation that best aligns the current deformed model to the query contour, and a second step that adjusts the shape parameters that deform the model to better improve the fit. This adjustment is constrained such that the deformed model shape is consistent with that of the training examples. Enforcing this constraint is accomplished by checking that the adjusted shape parameters do not fall outside of the distribution of the training shapes; if the parameters do fall outside, they are set according to the closest shape in the distribution.

We differ from a standard ASM in how a shape adjustment is constrained to lie in the space of training shapes. In a classical ASM framework, where training shapes belong to the same class and exhibit relatively minor deformations, the shape points live near a low-dimensional subspace and are approximately Gaussian distributed. Under this condition, enforcing this constraint amounts to simply verifying that the adjusted shape parameters do not deviate more than a certain number of standard deviations (e.g., 3) from the mean.
shape; beyond that, the point is scaled down to correspond to the closest point within
the distribution. In our case, the set of training shapes is quite heterogeneous (spanning
multiple categories and all possible initial correspondences), yielding a complex shape
space boundary whose enclosed distribution is not well approximated by a Gaussian.
We therefore need a different way to constrain a given adjusted shape to fall within the
shape space spanned by the training set. Since our training set densely covers the space of
shapes of interest and a low-dimensional subspace captures most of the shape variance, a
Nearest Neighbor Data Description method [167] provides a fast mechanism for checking
if a query shape belongs to the target distribution. If an adjusted shape does not belong
to the distribution, it is constrained to be in the distribution by replacing it by a nearby
neighbor in the distribution. To avoid falling in local minima, we randomly choose the
replacement from among the \( k(t) \) nearest neighbors, where \( k(t) \in \mathbb{N}_{>0} \) is a non-increasing
function of the iteration number \( t = 0, \ldots, T - 1 \). In our experiments, we obtained good
results by using \( k(t) = 5 + \lfloor 70 \times (1 - \frac{t}{T-1}) \rfloor \).

Since we attempt to bridge the gap between image contours and ideal model contours,
a simple distance field between image and model contour landmarks is inappropriate as
a driving force to guide the model deformation process. Such an approach would give
the same weight to all contour landmark correspondences and may fail to deform the
model appropriately in the case of minor region undersegmentation or minor contour
shape deviation from the model. In order to cope with these conditions, we define the
deformation force for each landmark \( i \) as:

\[
\vec{\delta}_i = (1 - \alpha(t))\vec{\delta}_L^i + \alpha(t)\vec{\delta}_C^i,
\]

(4.1)

where

\[
\vec{\delta}_L^i = \vec{q}_i - \vec{m}_i,
\]

(4.2)

\[
\vec{\delta}_C^i = \text{closest}_{i}(\vec{q}(\vec{m}_i)) - \vec{m}_i,
\]

(4.3)

\( \vec{q}_1, \ldots, \vec{q}_N \in \mathbb{R}^2 \) are image contour landmarks sampled at equidistant positions along the
contour, \( \text{closest}_{\vec{q}}(\vec{m}_i) \) is the point along the image contour \( \vec{q} \) that is closest to model landmark \( \vec{m}_i \), and \( \alpha(t) \in [0, 1] \) is a non-decreasing surjective function of the iteration number \( t \). (In our implementation, we defined \( \alpha(t) = \frac{t}{T-1} \).) In this way, at the beginning of the fitting process, the attraction forces between the image and model contours are globally driven purely by landmark correspondences. This roughly aligns the model to the cycle. As the iterations proceed, the model deformation becomes increasingly driven by local contour attraction forces, giving more weight to the consensus of the image contours that are closest to the model, and thus letting the deformation process overlook significant image contour departure from the abstract model as well as some undersegmentation. Note that our fitting problem is more constrained than a standard ASM framework, since all landmark correspondences between a consistent cycle and the model ASM are known, and the influence of outlier landmarks on a consistent cycle (and their resulting incorrect correspondences) is decreased over time (iterations). Thus, the ASM is initially fit to an entire closed contour (not just a portion), but it converges to fit the relevant portion. Figure 4.6 has an example showing the steps of our robust ASM fitting method for part shape abstraction of a consistent cycle.

At first glance, it may seem unfeasible that a single ASM may be able to learn the PDM of different vocabulary shapes and for all cyclical landmark alignments. Such an approach works in our case due to a few distinctive characteristics of our problem. In our case, all correspondences between query and model contour landmarks are established, something that is not true in a standard use of an ASM. This is a very important advantage that greatly simplifies the problem, eliminating the possibility of wrong landmark correspondences. A second important characteristic of our approach is the previously mentioned mechanism used to constraint a deformed model to fall within the distribution, in conjunction with the fact that for any deformation of the part models there is a close sampled shape in the PDM (for any possible cyclical landmark alignment). The distribution associated with each distinct shape in the vocabulary yields a cluster
Closest Point Forces $\delta_i^C$

Landmark Forces $\delta_i^L$

Landmarks

Active Shape Model

Consistent Cycle

Initial Alignment

$\alpha = 0$

$\alpha = 0.25$

$\alpha = 0.5$

$\alpha = 0.8$

$\alpha = 0.9$

$\alpha = 1$

Cycle Abstraction

Figure 4.6: Consistent cycle abstraction (see text for discussion).
in the overall distribution of all training shapes. The fitting process starts anywhere in
the overall distribution, and the process may initially jump between clusters for a while.
Eventually, as the fitting converges, it will focus on a particular type of model part with
the shape deformations becoming smaller. Thus the successive deformed shapes will
tend to stay in the same cluster, with the fitting process resembling a standard ASM on
a single shape.

Finally, it is is possible that inconsistent cycles are misclassified as consistent. After
convergence, if the distance between the cycle and the model exceeds a threshold, or
the cycle coverage by the model (i.e., proportion of cycle contour covered by model
landmarks) is poor, the cycle is discarded as a false positive. We compute the scale-

\[ d(\vec{q}, \vec{m}) = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\vec{\delta}_i\|}{\sqrt{\frac{1}{N} \sum_{i=1}^{N} ||\vec{q}_i - \vec{\bar{q}}||^2}} \]

between the fitted model \( \vec{m} \) and the cycle \( \vec{q} \), where the denominator is a normalizing
factor corresponding to the geometric mean distance between cycle landmarks \( \vec{q}_i \) and
their centroid \( \vec{\bar{q}} \), thus making \( d(\vec{q}, \vec{m}) \) a scale-independent measure with an intuitive
geometric interpretation; in our experiments, we obtained good results using a threshold
of 0.15. Figure 4.7 shows a consistent cycle and its model abstraction, as well as the
distance between each discreetly sampled model point and its closest cycle point.

### 4.4 Results

In order to evaluate our framework, in the absence of competing approaches and bench-
mark datasets for evaluating part-based shape abstraction using a vocabulary of simple
part models, we created an annotated dataset with 67 images\(^3\), containing object exem-
plars whose 3-D shape can be qualitatively described by cylinders and bent or tapered

\(^3\)Available at [http://www.cs.toronto.edu/~psala/datasets.html](http://www.cs.toronto.edu/~psala/datasets.html) at link “ECCV 2010 Dataset”.

cubic prisms. The abstract visible surfaces of each 3-D shape were hand-labeled using 2-D models drawn from our vocabulary. Figures 4.8 and 4.9 illustrate the output of our system on a number of examples in the dataset: column (a) shows the input image; column (b) shows the region oversegmentation used as input to our algorithm, computed using the local variation approach by Felzenszwalb and Huttenlocher [50] with a fixed parameterization on all images; column (c) shows the consistent cycles from which the shapes in column (d) were abstracted, representing the recovered parts closest to the ground truth in column (e). The numbers inside recovered abstract parts in column (d) indicate the rank of the part among all recovered parts in that image, computed as a function of the distance to the contours of the cycles that they are abstracting. The target regions can sometimes rank low if their degree of abstraction is high compared to non-target regions in the image (whether real or segmentation artifacts) that require less abstraction. Note that in some cases, e.g., the blender body in row 3 of Figure 4.9, the ideal ground truth part (e.g., corresponding to the projection of the body of a tapered cylinder) did not exist in the vocabulary.

Our ability to abstract the shape of a cycle of contours with high local irregularity (shape “noise”) means that many false positive parts will be recovered. As a result,
the rankings of some of the ground truth shapes among the hypotheses are poor. This is entirely due to the naive scoring mechanism (absolute fitting error) that tends to favor small, well-fitting shapes over larger abstractions. While more inspired scoring functions may increase the ranks of the target shapes, we mean only to illustrate the significant extent to which the target shapes are indeed generated. It is at a later stage, when contextual constraints are added, where an aggressive pruning of false positives
is expected. In Chapters 5 and 6, we explore powerful contextual relations, including temporal-coherence and 3-D shape information to prune many of these false positives. For example, if the surfaces in our images can indeed be the projections of volumetric parts, such as cylinders or prisms, then there are strong constraints on the shapes and relations of the component faces (parts) of their aspects. Other constraints are also possible, such as pruning smaller surfaces that are subsumed by larger surfaces. Our focus in the presented approach is mainly on the initial recovery of the primitive parts. As can be seen from Figures 4.8 and 4.9, our framework is able to recover and abstract
many of the surfaces of the objects.

To provide a quantitative evaluation of our framework’s ability to recover the correct abstractions amid the abstract shapes (hypotheses) recovered from an image, we analyze the rank of a ground truth shape among the ranked hypotheses. (A ground truth part was considered to have been recovered when a 2-D part from the same shape family and with high overlap was detected, also having similar dimensions and orientation.) Figure 4.10(a) shows the distribution of ground truth part rankings. The x-axis corresponds to the order $n$ in which ground-truth parts are detected, and the y-axis shows the distribution of ranks for the $n$-th detected ground truth part. For example, the median rank (red bar) of the first detected ground truth part (box-plot in leftmost column) in an image was 2 among the ranked hypothesis in the image. The median rank of the second detected ground truth part (box-plot in second column) was 10, and so forth. The number of ground truth parts, on average, was 3, while the number of part hypotheses generated for an image, on average, was 71. No attempt was made to eliminate redundant models (i.e., models of the same category with roughly the same parameterization), and no
size filtering was performed. From these results, we conclude that the target (ground truth) shapes reside in a manageable number of hypotheses, and we expect that with the application of contextual constraints, the false positive shapes can be drastically reduced. Figure 4.10(b) illustrates precision and recall for our database. While precision is low due to the high number of false positives (due to lack of contextual pruning and non-maximum suppression), our recall of ground truth shapes is reasonably good, typically failing in the presence of significant region undersegmentation. In terms of running time, a typical run of the consistent cycle detection algorithm requires an average of about 40,000 iterations, which takes about 3 seconds in our MATLAB/C++ implementation running on a laptop. The model abstraction algorithm, implemented in C++, takes a second or two to process all consistent cycles detected in an image.

Exploring the results in more detail, we see that Figures 4.8(d) and 4.9 (d) show the ability of our approach to abstract object surfaces that are locally highly irregular due to noise or within-class variation, but capture a model shape at a higher level of abstraction. In some cases (e.g., Figure 4.8 row 5, and Figure 4.9 rows 1 and 3), we see misalignment with a neighboring shape. This can be due to two reasons: (1) the vocabulary may not contain the appropriate shape to model the surface; and (2) the shapes are recovered independently, with no alignment constraints exploited; such constraints, as well as other constraints, will play an aggressive role in pruning/aligning hypotheses in our future work. In all the examples, we can see that the model abstraction process is able to cope with region undersegmentation when it is restricted to a relatively small section of the contour. Figure 4.8 (row 1) and Figure 4.9 (row 9) show examples of cases in which, although some portions of the correct surface boundaries are missing, the models are still correctly fit due to the consensus of the correct surface contours.

Figure 4.11 illustrates some weaknesses of our approach. The top row shows a case in which an object’s surface is missing (i.e., box’s left face) due to strong undersegmentation of the input to our algorithm. Since the consistent cycle detection mechanism already
keeps incremental hypotheses of partial contour matchings, in future work we plan to allow informative consistent paths to be abstracted (using a similar framework). This will not only accommodate region undersegmentation, but also region occlusion and partial part abstraction. In the second row of Figure 4.11, we see a case in which a consistent cycle was abstracted by a model of an incorrect category (i.e., the rim of the central bowl). This is because either the abstraction approach was trapped in a local minimum or there is an inherent shape ambiguity in the noisy contour. This can be remedied by allowing the abstraction process to return not just one model, but a list of candidate models that lie within a certain distance from the consistent cycle. We expect our future use of relational constraints to help overcome such ambiguity, and in this case “flip” the rectangle to an ellipse. Finally, the third row of Figure 4.11 shows a case where the ranking of the correct models is poor due to the presence of many uninteresting image region groups whose shape is consistent with vocabulary model shapes (i.e., there is a large number of region groups forming regular quadrilaterals). The use of context or non-maximum suppression can eliminate many of these false positives.
4.5 Comparison of the Edge-Based and Region-Based Part Hypothesis Generation Methods

We include here a brief comparison between the 2-D abstract part hypothesis generation approaches of Chapters 3 and 4. In Chapter 3, we introduced the Model Grid, an edge-based abstraction mechanism for 2-D abstract part hypothesis generation. In Chapter 4, we introduced a region-based abstraction framework based on the search of Consistent Cycles and active shape model (ASM) regularization. Both approaches are based on a small user-provided vocabulary of abstract 2-D parts. The model grid is constructed from training images generated from deformed instances of the vocabulary parts, yielding an indexing mechanism where edge evidence for each part is accumulated. The consistent cycle search is achieved by means of classifiers trained on the vocabulary parts and their subcontours, and an ASM is used to regularize the consistent cycles.
Although both approaches take the same input (a part vocabulary and contour evidence from the input image) and have the same goal, namely, to yield a manageable set of abstract 2-D part hypotheses including most, if not all, true positives, there are some fundamental differences in their performance, strengths, and weaknesses. In terms of speed, the model grid is the slowest of the two approaches, as it has to place the grid at every location of the image, at all discrete rotations and at all possible anisotropic scales. Typically, the runtime of this approach is on the order of 20 minutes per image. The algorithm can be formulated in a SIMD fashion, and so it is possible to speed it up via a GPU implementation. Consistent cycle search and regularization, on the other hand, is generally fast, on the order of a second for an average image. The runtime actually depends on the total number of consistent cycles that are present in the region boundary graph constructed from the input image region segmentation. The number of consistent cycles can be very large when there is repetitive regular structure in the graph (e.g., a checkerboard). In such cases, the algorithm can take longer to yield all such cycles. The time used by the ASM to regularize each of the consistent cycles is small, typically on the order of a fraction of a second.

The size of the space of hypotheses generated by each method is determined by different factors. In the case of consistent cycles, the space of hypotheses is constrained to lie along the edges of the region boundary graph. As mentioned earlier, the presence of repetitive regular structure has a direct impact in the size of this space. The space of hypotheses of the model grid method is, in general, much larger since hypotheses are not constrained to lie along region boundaries but only to be supported by edge evidence. Therefore, the number of hypothetical parts with sufficient support can be very large in a cluttered or textured scene. In the case of the consistent cycles approach, the impact of texture can be reduced if its input comes from a texture-based region segmenter.

A negative effect of the restriction of hypotheses to live along region boundaries, which achieves a drastic reduction in the size of the hypothesis space, is that the approach is
very sensitive to segmentation errors. If the rough boundary of a true positive part is not found in the region segmentation, due to undersegmentation, that part will not be detected. On the other hand, excessive oversegmentation may yield a fine lattice structure along region boundaries, where virtually any part can be detected, e.g., in a rather fine superpixel segmentation, any abstract 2-D part can be hallucinated.

A regular and unavoidable visual event in a scene is occlusion. Part occlusion yields partial 2-D part visibility, which is a problem for the consistent cycles method, which expects the entire part to be accounted for by region boundaries. The model grid approach can handle occlusion to a larger extent, as long as there is sufficient distributed boundary support for the part. That is, partial occlusion is fine, and only cases in which a significant portion of the part is unaccounted for by edge evidence may be problematic.

In conclusion, we have presented two methods for abstract 2-D part hypothesis generation, each with their own strengths and weaknesses. A better solution to the hypothesis generation problem may lie somewhere in between these two approaches. Specifically, we envision a method that relies on image edges to hypothesize abstract parts, like in our edge-based approach. However, instead of exhaustively trying image windows at all locations, scales, and orientations, it would use the structure of a hierarchy of image region segmentations and the underlying geometry of their region subsets to efficiently guide the localization of such windows. Another potential solution worth exploring is a modification of the region-based hypothesis generation method to generate part hypotheses directly from consistent paths (i.e., non-closed contours) short of becoming consistent cycles.

4.6 Conclusions

In this chapter, we have presented a framework for grouping contours (region boundaries) into parts according to a user-defined vocabulary of abstract parts models. Our contri-
butions are threefold: (1) we train a classifier on all possible component fragments of a vocabulary of parts, and use the resulting set of classifiers to guide a grouping process that searches for cycles of locally irregular contours that are consistent, at some level of abstraction, with some model shape; (2) the consistent cycles are abstracted and categorized using a novel application of an ASM model which captures the entire vocabulary of shapes with a single model and which needs no proper initialization; (3) the resulting framework reports promising first steps toward part-based shape abstraction from images of real objects, and establishes a number of important directions for future work.

In Chapter 5, we improve precision by employing spatiotemporal shape coherence priors for part hypothesis selection when a video sequence in which there is a relative motion between the camera and the objects in the scene is available. In Chapter 6, we strengthen the role of part relations in the selection process by explicitly including 3-D constraints in the process, with the ultimate goal of recovering a set of abstract volumetric parts from an image. As illustrated in Figure 4.12, 3-D constraints naturally enforce stronger (i.e., specific) part relations (such as part co-occurrence) than generic relations like proximity or alignment.
Chapter 5

Part Hypothesis Selection by

Spatiotemporal Coherence Priors

In Chapters 3 and 4, we presented two frameworks in which a small vocabulary of abstract part shape models was used to both group and abstract image contours, yielding a covering of the image with a set of 2-D abstract parts which model the projections of the surfaces of a set of abstract volumetric parts that describe the coarse shape of the object. Thus, rather than invoking an object-level shape prior (detector), which we don’t have since we don’t know what we’re looking at, we instead invoke a small, finite set of intermediate-level, domain-independent shape priors to drive the grouping and abstraction processes (we assume only that the parts can be assembled to describe a significant portion of any object in the database). While the methods showed clear promise, there is a fundamental trade-off between abstraction and ambiguity; as a greater degree of abstraction of a set of image contours is allowed, the more ambiguous the abstraction, i.e., the abstraction is consistent with an increasing number of shape models.

In this Chapter, we exploit the dimension of temporal coherence to help cope with the ambiguity of a shape abstraction inherent in a single static image. Figure 5.1 shows an overview of the presented approach. Like in the previously mentioned approach, we rely
on a small, user-defined, abstract shape vocabulary to drive the process of perceptual grouping in a single frame. However, unlike that approach, which restricts its analysis to a single image, here we assume access to a video sequence in which there is relative motion between the camera and the object, and exploit the spatiotemporal coherence of a perceptual group to reduce false positives that are abundant in a single image. If a perceptual group of contours is consistent with an abstract part model, and is stable over time in terms of its shape (continues to match the same part model) and pose, then we consider the perceptual group to be non-accidental. We introduce a novel probabilistic, graph-theoretic formulation of the problem, in which the spatiotemporal consistency of a perceptual group under camera motion is learned from a set of training sequences. In a set of comprehensive experiments, we demonstrate (not surprisingly) how a spatiotemporal framework for part-based perceptual grouping significantly outperforms a static image version.
5.1 Related Work

Although we know of no approaches dealing with the problem of finding spatiotemporally coherent perceptual groups, this can be considered, in a sense, to be similar to the tracking problem. Tracking approaches often require some type of initialization to indicate the location, in an initial frame, of the region or object of interest that is to be tracked. Moreover, if during the tracking process the tracker’s focus of attention drifts away from the objects of interest, some recovery mechanism needs to be in place to recover from such errors. Our method, however, requires neither an initialization nor a drift-recovery step, since the hypothesis detection process applied at each frame acts as an interest operator, yielding the set of image regions of interest in each frame.

The solution proposed in this chapter to the problem of determining multiple sequences (i.e., trajectories) of closed contours, each corresponding to the boundary of a particular object surface across frames, is formulated in graph-theoretical and probabilistic terms, and solved efficiently using the Viterbi algorithm. Quach and Farooq [136] have applied Viterbi to solve the data association problem for single-target tracking in a maximum likelihood fashion, assuming that object motion is a Markov process. More recently, Yan et al. [184] have used Viterbi for single-target tracking of a tennis ball in video. These approaches only admit a single-target, and require both an initialization step and a step to identify the object of interest at the end of the sequence. Our method, however, is not only multi-target, modeling both shape and appearance to disambiguate surface correspondences across frames, but also does not require any type of initialization or recovery mechanism. Moreover, our formulation models second-order relationships between the position, orientation and scale of the surface contours across frames rather than simply modeling first-order smoothness of the tracked feature’s location across frames.
5.2 Overview of the Approach

As shown in Figure 5.1, the input to our spatiotemporal perceptual grouping framework is a video sequence and a vocabulary of shape primitives. First, hypotheses are independently recovered from each frame using the method explained in Chapter 4. The algorithm outputs cycles of contours that are consistent with one of the model (training) shapes. However, as mentioned in Section 4.3, the consistent cycle classifier may yield many false positives at reasonable recall rates. Some of the recovered consistent cycles may yield shapes that are qualitatively different from those in the vocabulary, while in other cases the shapes may be consistent but accidental, e.g., a number of the detected consistent cycles might not correspond to actual scene surfaces. By exploiting spatiotemporal consistency of these consistent cycles across a video sequence, we can filter out many of these false positives. That is, we assume that the only cycles that are likely to be caused by the projection of an actual scene surface are those whose shape and internal appearance remain stable or vary smoothly across consecutive frames.

We formulate the problem of finding sequences of consistent cycles with temporally coherent shapes across frames of a video sequence in graph-theoretical and probabilistic terms. We refer to such sequences as trajectories. The potential correspondences between consistent cycles detected at different frames are modeled by constructing a graph in which a maximum-weight path corresponds to a trajectory with maximum joint probability of including all and only those consistent cycles in the sequence that correspond to the same scene’s surface boundary. Specifically, nodes in the graph encode pairs of potential matches between consistent cycles in nearby frames, edges connect pairs of nodes that share a common consistent cycle, and edge weights encode the probability of correctness of the cycle matches connected by the edge conditioned on geometric and photometric properties of the cycles involved. We learn this probability distribution from a few hand-labeled training sequences. The top trajectories of temporally coherent consistent cycles are obtained by iteratively applying the Viterbi algorithm on the graph.
to find paths with maximum joint probability, and removing from the graph the nodes involved in such paths.

5.3 Temporal Coherence of Consistent Cycles

Figure 5.2 (a) shows an example video sequence with four frames, the contour hypotheses detected in each frame (each named with a different letter), and line segments indicating the potential temporal correspondences between hypotheses. We formulate the problem of finding temporally coherent consistent cycles in a video sequence in graph-theoretical terms as the search for maximum-weight paths between the source and sink nodes of a particular directed graph \( G = (V, E) \). In order to obtain a more robust model of consistent cycle correspondence across frames, we not only model the first derivative of a cycle’s pose function (i.e., the cycle’s frame-to-frame change in position, scale and orientation), but we also model its second derivative, i.e., the change in the pose transformation function between corresponding consistent cycles. For this reason, instead of modeling the problem via a trellis graph, in which nodes represent consistent cycles and edges model potential cycle correspondences across frames, we actually define \( G \) as the dual of such a graph. (See Figure 5.2 (b).) Namely, nodes represent potential matches between consistent cycles detected in close spatial and temporal proximity, and there is an edge between each pair of nodes that share a common consistent cycle. Two special nodes, a source and a sink, also exist, which are connected to every other node in \( G \). Edge weights correspond to a log-probability conditional to various attributes of the cycles involved in the edge, such that a maximum-weight path from source to sink corresponds to the trajectory with the highest joint probability of containing the densest sequence of correct consistent cycle matches.
5.3.1 Retrieving Consistent Cycle Trajectories

The construction of graph $G$ is as follows. The set $V$ contains two special nodes, $s$ and $t$, called source and sink, respectively. All other nodes in $V$ correspond to potential matches between consistent cycles detected at different frames and are referred to as internal. Formally, if $C_i$ is the set of all consistent cycles detected at frame $i$, the set of nodes $V$ is defined as $V = V^{\text{internal}} \cup \{s, t\}$, where $V^{\text{internal}} \subset \bigcup_{i<j} C_i \times C_j$. An internal node involving consistent cycles $x$ and $y$ is denoted by $<x, y>$. There is an edge connecting every pair of nodes that share a common consistent cycle. The direction of these edges, referred to as internal edges, is determined by the frame numbers at which
the non-common consistent cycles in the pair were detected. Namely, edges leave from
the nodes whose non-common consistent cycles are detected at earlier frames. There is
also a directed edge from $s$ to every internal node, as well as directed edges from all
internal nodes to $t$. The former edges called initial, while the latter are called final.
Formally, $E = E_{\text{initial}} \cup E_{\text{internal}} \cup E_{\text{final}}$, where $E_{\text{initial}} = \{(s, <x, y>) : <x, y> \in V\}$,
$E_{\text{internal}} = \{( <x, y> , <y, z> ) : <x, y> , <y, z> \in V \text{ and } n(x) < n(y) < n(z) \}$, and
$E_{\text{final}} = \{( <x, y> , t ) : <x, y> \in V \}$, where $n(x)$ denotes the frame at which cycle $x$ was
detected.

A match $<x, y>$ is said to be correct iff consistent cycles $x$ and $y$ correspond to
projected boundaries of the same image surface. The cardinality of $V$ (and thus the
total running time of the algorithm) can be kept low by not including in $V_{\text{internal}}$ cycle
correspondences that are highly unlikely to be correct. This can be done by assuming that
a cycle undergoes smooth changes in location, scale, shape, and appearance across frames.
Therefore, potential matches can be considered only between cycles whose distance along
these dimensions falls within given threshold values proportional to the distance between
the frames in which they were detected. Also, consideration can be restricted to matches
of cycles detected at frames that are within a specified maximum frame distance $W$. This
maximum frame distance should be chosen such that the likelihood of a consistent cycle
being undetected (e.g., due to undersegmentation) for that many consecutive frames is
low.

We model the change in appearance between two potentially corresponding cycles
by first approximating the shape of each consistent cycle by a polygon whose vertices
are points sampled at equidistant positions along the cycle. The internal appearance of
the first cycle is then modeled by computing a homogeneous triangulation of its poly-
gon (e.g., a Delaunay triangulation constraining triangle angles and areas to ensure an
approximately uniform sampling of the image region inside the cycle at a fine enough res-
olution). The triangulation is mapped onto the second cycle, for each cyclical alignment
between the two cycles’ polygonal points, by means of the estimated geometrical transformation between the cycles, and their appearance distance is measured in terms of the absolute difference between sampled image color values at the centroids of corresponding triangles. The change in appearance is set to the minimum of these absolute differences over all cyclical rotations. (Although this is a very simple attempt to assess appearance change, it suffices for the purpose of our investigation. Such a color-constancy assumption does not hold in general, e.g., when in the presence of non-Lambertian surfaces or when the relative position between a surface and a light source changes.)

From all trajectories of consistent cycles corresponding to some particular scene surface, we are interested in finding the trajectory that is the densest, i.e., the one that does not miss any frame where a consistent cycle accounting for the specific surface exists. A correct match $\langle x, y \rangle$ is said to be consecutive iff no consistent cycle corresponding to the same surface boundary as $x$ and $y$ was detected in a frame $k : n(x) < k < n(y)$. Let $x \sim y$ represent the relation “$\langle x, y \rangle$ is a correct and consecutive match”, and let $\neg B(x)$ ($\neg A(x)$) symbolize the predicate “no consistent cycle that correctly matches $x$ was detected before (after) frame $n(x)$.” If $\langle x_i, x_j \rangle$ is a potential match, then $T_{ij}$ represents the geometric transformation between cycles $x_i$ and $x_j$. For the definition of the edge weights, the scoring of the paths is what matters. It is important that paths corresponding to the largest sequences of correct and consecutive matches score highest among all other paths. We formulate this scoring in probabilistic terms. The weight $w(\cdot)$ of an edge is as a log conditional probability defined depending on the type of edge:

$$w((s, <x_1, x_2 >)) = \log (p(\neg B(x_1))p(x_1 \sim x_2|\theta_{12}))$$

(5.1)

$$w((<x_1, x_2 >, < x_2 , x_3 >)) = \log (p(x_2 \sim x_3|x_1 \sim x_2, \phi_{123}))$$

(5.2)

$$w((<x_1, x_2 >, t)) = \log (p(\neg A(x_2)))$$

(5.3)

where $\theta_{ij} = \langle \tilde{t}_{ij}, \delta n_{ij}, \delta sh_{ij} \rangle$ and $\phi_{ijk} = \langle \tilde{t}_{jk}, \delta n_{jk}, \delta sh_{jk}, \delta T_{ijk} \rangle$ are attributes of the consistent cycles involved in the edge. Namely, $\tilde{t}_{ij} \in \mathbb{R}^2$ is the change in contour position
between \( x_i \) and \( x_j \), \( \delta n_{ij} = |n(x_j) - n(x_i)| \), \( \delta sh_{ij} \) is the shape distance between cycles \( x_i \) and \( x_j \), and \( \delta T_{ijk} \) is the difference between the transforms \( T_{ij} \) and \( T_{jk} \) computed at each consistent cycle correspondence. (See Figure 5.2 (b).) We define \( \delta sh_{ij} \) as the minimum normalized point-to-point contour distance between the two cycle shapes over all their possible cyclical alignments, i.e.,

\[
\delta sh_{ij} = \min_{d=0,\ldots,N-1} \sum_{k=1}^{N} \| \overrightarrow{s_{h_{k}i} - s_{h_{k+d \mod N}}} \| / \min(r_i, r_j),
\]

where \( N \) is the number of equally sampled points along the cycle contours, \( s_{h_k} \) are the coordinates of the \( k \)-th sampled point along the contour of cycle \( x_h \), and \( r_h \) is the “mean radius” of cycle \( x_h \), i.e., the mean distance between the cycle’s centroid and each sampled point along its contour.

With this edge weight specification, a path \((s, <x_1, x_2>, \ldots, <x_{r-1}, x_r>, t)\) from source to sink achieving maximum weight corresponds to the trajectory of consistent cycles \( x_1, \ldots, x_r \) maximizing the probability

\[
p(\neg B(x_1))p(\neg A(x_r))p(x_1 \sim x_2 | \theta_{12}) \prod_{i=2}^{r-1} p(x_i \sim x_{i+1} | x_{i-1} \sim x_i, \phi_{i-1,i,i+1}).
\]

Now, under the following natural assumptions:

1. \( f \sim g, \neg B(f), \) and \( \neg A(g) \) are mutually independent,

2. \( x_i \sim x_j \) and \( \phi_{k,l,m} \) are independent if \( i \neq l \) or \( j \neq m \), and

3. \( x_i \sim x_j \) and \( \theta_{l,m} \) are independent if \( i \neq l \) or \( j \neq m \),

equation 5.4 is equivalent to the joint probability

\[
p (-B(x_1), x_1 \sim x_2 \sim \ldots \sim x_r, \neg A(x_r) | \theta_{12}, \{\phi_{i-1,i,i+1}\}_{i=2}^{r-1}),
\]

thus yielding \( x_1, \ldots, x_r \) as the trajectory of consistent cycles most likely to be the longest and densest trajectory of correct consistent cycle correspondences in the video sequence. Trajectories of consistent cycles can thus be efficiently generated in decreasing order of probability by the procedure shown in Figure 5.2 (c). That is, by iteratively applying
the Viterbi algorithm [59] on $G$ to find the maximum-weight path from $s$ to $t$, and then removing from $V$ all internal nodes belonging to such a path.

Due to undersegmentation errors in the low-level region segmentation of a frame $n$, which is the input to the consistent cycle detector, it is possible that no consistent cycle is detected in frame $n$ that corresponds to a surface boundary for which consistent cycles have been indeed detected in nearby frames. In these cases, the retrieved trajectories will be missing the frames in which the undersegmentation occurred. A surface’s position and shape can however be interpolated in a missing frame from its known position and shape in nearby trajectory frames. In our approach, we compute an initial guess for the position and shape of the surface boundary in frame $n$ by linearly interpolating the transformation between the corresponding detected consistent cycles in the closest frames around $n$. This guess is refined by optimizing the normalized cross-correlation between the image data internal to the consistent cycle in a nearby frame where it was detected, and the image data inside a 2-D window around the initial position estimate in frame $n$. The surface boundary is thus interpolated into frame $n$, unless the image appearance inside the contour in the estimated position of frame $n$ and the contour appearance in the closest frames differs significantly. In that case, the surface is assumed to be occluded in frame $n$.

### 5.3.2 Probability Density Estimation

In order to compute the edge weights, we need to model the probability distributions involved in Equations 5.1, 5.2 and 5.3. By applying Bayes’ rule, the probability function from Equation 5.1, $p(x_1 \sim x_2 | \theta_{12})$, can be rewritten as

$$
p(\theta_{12} | x_1 \sim x_2)p(x_1 \sim x_2)
p(\theta_{12} | x_1 \sim x_2)p(x_1 \sim x_2) + p(\theta_{12} | x_1 \sim x_2)p(x_1 \sim x_2), \tag{5.6}
$$

and the probability function $p(x_2 \sim x_3 | x_1 \sim x_2, \phi_{123})$ from Equation 5.2 as:

$$
p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2)p(x_2 \sim x_3, x_1 \sim x_2)
p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2)p(x_2 \sim x_3, x_1 \sim x_2) + p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2)p(x_2 \sim x_3, x_1 \sim x_2), \tag{5.7}
$$

We can thus estimate these probability distributions from training sequences.
Notice that we can factor \( p(\theta_{12}|x_1 \bowtie x_2) \) as

\[
p(\vec{t}_{12}|\delta n_{12}, \delta sh_{12}, x_1 \bowtie x_2)p(\delta n_{12}, \delta sh_{12}|x_1 \bowtie x_2),
\]

where \( \bowtie \in \{\sim, \bowtie\} \). In our experiments, we quantized the space of \( (\delta n_{12}, \delta sh_{12}) \) values, discretely modeling \( p(\delta n_{12}, \delta sh_{12}|x_1 \bowtie x_2) \) via a probability table. And \( p(\vec{t}_{12}|\delta n_{12}, \delta sh_{12}, x_1 \bowtie x_2) \) (for each quantized value of \( (\delta n_{12}, \delta sh_{12}) \)) was modeled by a multivariate Gaussian, which appeared to be a good approximation to this distribution. Analogously, \( p(\phi_{123}|x_2 \bowtie x_3, x_1 \sim x_2) \) can be factored as

\[
p(\vec{t}_{23}, \delta T_{123}|\delta n_{23}, \delta sh_{23}, x_2 \bowtie x_3, x_1 \sim x_2)p(\delta n_{23}, \delta sh_{23}|x_2 \bowtie x_3, x_1 \sim x_2),
\]

and so we modeled \( p(\delta n_{23}, \delta sh_{23}|x_2 \bowtie x_3, x_1 \sim x_2) \) by a probability table, and

\( p(\vec{t}_{23}, \delta T_{123}|\delta n_{23}, \delta sh_{23}, x_2 \bowtie x_3, x_1 \sim x_2) \) by a multivariate Gaussian distribution for each quantized value of \( (\delta n_{23}, \delta sh_{23}) \). The value of \( p(x_2 \bowtie x_3, x_1 \sim x_2) \) is empirically estimated from the training sequences from relative frequencies. Finally, lower-bounds for \( p(\neg B(x)) \) and \( p(\neg A(x)) \) were obtained by \( q^{n(x)-1} \) and \( q^{F-n(x)} \), respectively, where \( F \) is the total number of frames in the sequence and \( q \) is a tight lower bound of \( p(x \sim y|n(y) = n(x)+1) \) empirically estimated from the training sequences.

### 5.4 Results

We are not aware of any benchmark dataset for evaluating spatiotemporal contour grouping using abstract part models. Therefore, to evaluate our proposed approach, we generated an annotated dataset consisting of 12 video sequences\(^1\) (a total of 484 frames), containing object exemplars whose 3-D shape can be qualitatively described by cylinders, bent or tapered cubic prisms, and ellipsoids. The visible surface contours of each object’s 3-D shape that are consistent with 2-D models from our vocabulary were hand-labeled.

\(^1\)Available at [http://www.cs.toronto.edu/~psala/datasets.html](http://www.cs.toronto.edu/~psala/datasets.html) at link “ACCV 2010 Dataset”.
Figures 5.3 and 5.4 illustrate the output of our approach on two selected frames (closer to the beginning and end) of six sequences in the dataset: row (a) shows the input frames; row (b) shows the consistent cycles closest to the ground-truth detected at each static frame (obtained by the method in Chapter 4); row (c) shows the temporally coherent detected consistent cycles closest to the ground-truth; and row (d) shows the ground-truth surface contours. Notice that images in rows (c) and (d) also show the boundaries of the region oversegmentation used as input to the consistent cycles detection approach (computed using the “statistical region merging” approach of Nock and Nielsen [120] with its parameters fixed for all frames from all sequences). The numbers in the top-right corner of each image in rows (b) and (c) correspond to the total number of consistent cycles in each case. The numbers appearing in the centroid of the recovered hypotheses in these rows indicate the rank of the hypothesis among all recovered hypotheses in the frame. In the case of static consistent cycle detection, such ranking is a function of the fitting error between the consistent cycle and the model abstracting the cycle (as in Chapter 4). A hypothesis is ranked based on the average distance from equidistantly sampled points along the abstracting model’s contour to their closest points on the hypothesis’
contour, normalized by the mean distance from the hypothesis’ centroid to its contour. (As in Chapter 4, a significant portion of the hypothesis’ contour has to be explained by the model for an abstraction to be considered correct.). In the spatiotemporal case, hypotheses are ranked by the length of the consistent cycle’s temporal flow (i.e., the number of frames in which the cycle is found to be temporally consistent).

These ranking values were obtained after a non-maximum suppression step was applied to eliminate redundant cycle hypotheses in the static and dynamic cases, by discarding all but one of the similar consistent cycles competing for the same image evidence. (The cycle achieving the smallest shape distance to all other competing cycles was kept.) In the static case, (as in Chapter 4), detected hypotheses with a high fitting error to their abstraction shapes were also discarded. By comparing the rankings of the recovered hypotheses corresponding to ground-truth parts in the static (row (b)) and dynamic (row (c)) cases, we can see that employing temporal coherence outperforms the static version, as the rankings in row (c) are consistently higher than those in row (b). In some cases, even the rankings of ground-truth parts in row (c) correspond to the top ones. Moreover, the total number of candidate hypotheses in the static case is generally higher than in the dynamic version, demonstrating the superior performance of the dynamic approach to prune false positive hypotheses.

A quantitative evaluation of our spatiotemporal grouping framework is shown in the precision-recall curves of Figure 5.5, where it is compared to the approach of Chapter 4 as a baseline. There, it can be seen that both precision and recall increase substantially when temporal coherence is taken into account. The increase in precision can be explained as the result of the pruning ability of our temporal coherence framework on false positive consistent cycles. Since such hypotheses are produced by accidental arrangements of texture or image structure in a single frame, they are unlikely to be temporally stable. Moreover, in the spatiotemporal case, hypotheses are ranked by their persistence, which proves to be a better measure of hypothesis relevance than ranking by the fitting error.
between a consistent cycle’s contour and its model abstraction contour, as employed in the static case. The improved recall is the result of interpolating hypotheses when gaps of false negatives (mostly due to undersegmentation) have a length not greater than the maximum frame distance $W$ used in the construction of graph $G$. (In our experiments, $W = 6$.) In terms of running time, the entire process of searching for consistent cycle trajectories in a video sequence takes an average time of less than 5 seconds per frame, in our MATLAB implementation running on a laptop.
5.5 Conclusions

The results presented in this chapter demonstrate that the semantic gap between real scene contours and the abstract parts that make up categorical shape models can be bridged with the help of a small vocabulary of part models. In particular, they indicate the potential of the presented frameworks to recover abstract part structure from images of real objects. If a few correct parts can be identified, they may be sufficient to form a powerful index into a collection of object candidates which, in turn, can be used in a top-down manner to guide part selection.

Yet as the degree of abstraction between image contours and abstract parts increases, so too does the ambiguity of a perceptual group of image contours – if abstraction is viewed as a process of “controlled hallucination”, the more you hallucinate, the greater the possible mappings to different parts. We have shown that by imposing spatial and spatiotemporal constraints on the grouping process, we can significantly reduce such ambiguity, ensuring greater precision of the recovered abstract parts which, in turn, can facilitate the indexing and recognition of categorical shape models.
Chapter 6

Recovering Abstract Volumetric Parts

In Chapter 4, we introduced an image abstraction process that used a collection of abstract 2-D part models (closed contours) to drive both perceptual grouping and perceptual abstraction, yielding a covering of an image with a set of 2-D abstract part models, as illustrated in Figure 6.1(a-c). The framework could be seen as a controlled shape “hallucination” process, whereby the coarse shape of a noisy cycle of contours was compared to a given part model. While such abstraction is critical to bridging the gap between actual image features and true categorical features, such hallucination can be highly ambiguous, for the more you’re allowed to hallucinate, the more models you can “imagine” from your data. As a result, our proposed method exhibited good recall but offered poor precision, as shown in Figure 6.1(c). In the previous chapter, we exploited spatiotemporal coherence to improve precision. But what if motion information is unavailable? Where can we find other constraints? We will find them in the interactions among the shapes, for they are clearly not independent.

In this chapter, we extend the framework of Chapter 4 (including the improvements detailed in Appendix A) in two very important ways. First, we exploit the interactions
Figure 6.1: Extending the Framework of Chapter 4 to 3-D: (a) original image; (b) the vocabulary of abstract 2-D part models used to drive the perceptual grouping and abstraction process; (c) the resulting “covering” of the image with abstract parts drawn from the vocabulary. Such model-based abstraction is an ambiguous process and the assumption of part independence yields many ambiguous hypotheses resulting in poor precision; (d) the vocabulary of abstract 3-D part models used to drive our perceptual grouping and abstraction process; (e) our framework takes the output of the method from Chapter 4 and yields a “covering” of the image with part aspects drawn from the possible projections of the 3-D vocabulary. The relational constraints among the faces in an aspect significantly improve 2-D part precision. (f) from the recovered aspects, we learn to infer the 3-D shape and volumetric pose, yielding a configuration of viewpoint-invariant 3-D abstract shape features that supports efficient 3-D model retrieval.
of the 2-D shapes to yield a powerful set of constraints that significantly improve precision. By thinking about the problem in 3-D, we instead define the 2-D shapes as the projections of the surfaces of a vocabulary of 3-D qualitative volumetric parts, as shown in Figure 6.1(d). Moreover, the surface adjacencies of covisible sets of surfaces define aspects over these 2-D shapes [38]. By exploiting the structure of these aspects, we now have a powerful set of constraints with which to select and group a low-precision set of hypotheses, as shown in Figure 6.1(e).

In our second major contribution, we learn a mapping from the relative distortions of the 2-D parts (or faces) in a given aspect to the actual 3-D shape and pose of the volumetric part “behind” the aspect, as shown in Figure 6.1(f). In effect, we have extended [146] from a 2-D framework, whereby a vocabulary of 2-D parts is used to drive a 2-D image abstraction process, to a 3-D framework, whereby a vocabulary of 3-D parts (and their 2-D projections) is used to drive a 3-D image abstraction process. The advantage of a 3-D abstraction process is clear. If we can recover a configuration of causally related 3-D parts, including their shapes and poses, then whatever indices we compute over the configuration are viewpoint invariant, supporting a database of object-centered 3-D models, and offering a dramatic space and search complexity savings over view-based object representations.

### 6.1 Abstract Volumetric Part Recovery Overview

Our approach to abstract volumetric part recovery takes, as input, an image, a small vocabulary of 3-D abstract volumetric part categories, and a set of 2-D abstract face hypotheses extracted from the image. Faces are assumed to be instances of projected surfaces of volumetric parts drawn from the vocabulary. We adopt the face detection approach of Chapter 4, with two very important differences. Rather than learn a set of face detectors from deformed and noise-perturbed instances of faces sampled from a
vocabulary of 2-D part models, we learn a set of face detectors from faces derived from the projected surfaces of deformed and noise-perturbed instances of volumetric parts sampled from the input vocabulary. A simple, illustrative example is shown in Figure 6.2, in which the set of face hypotheses in Figure 6.2(b) are generated by detectors trained on the input vocabulary shown in Figure 6.2(c) and applied to the input image in Figure 6.2(a).

The face hypotheses and their relations are captured in a face relational graph containing two types of nodes. A face node represents a single face hypothesis (potentially) corresponding to the projection of an abstract volumetric part surface, while a relational node represents the grouping of two proximal face hypotheses (potentially) corresponding to the projections of two adjacent surfaces of an abstract volumetric part. Similarly, there are two types of edges in the face relational graph. A relational edge connects a pair of nodes (i.e., two relational nodes or a relational node and a face node) that share a common face hypothesis, and is used to enforce the local consistency of the labels of the common face hypothesis across the nodes that share it. A selection edge connects a pair of face nodes whose corresponding face hypotheses have high area and contour overlap, and is used to ensure that competing hypotheses are not simultaneously selected by an interpretation. Returning to our illustrative example, Figure 6.2(d) depicts the face relational graph derived from the detected face hypotheses in Figure 6.2(b). Black dots represent face nodes, green squares represent relational nodes, solid lines correspond to relational edges, and dashed lines correspond to selection edges.

Our challenge is to select, from among the low-precision set of face hypotheses, a set of faces that represent the projected surfaces of volumetric abstractions of the actual 3-D parts that make up the objects in the image. We formulate this as a graph labeling problem in which nodes are labeled according to the set of possible face and relational labels derived from the volumetric part vocabulary. For example, Figure 6.2(c) shows some of the face labels (red) and relational node labels (red-blue) derived from the simple 3-part vocabulary; the labels for the single-face aspects of the cuboid and cylinder and
Figure 6.2: Face relational graph and labeling example: (a) input image; (b) generated 2-D part hypotheses for input image; (c) input vocabulary and induced face and relational labels; (d) face relational graph with face nodes in black, relational nodes in green (blue and red connections in a relational node indicate how the faces in its relational label are to be assigned to its adjacent face nodes), relational edges shown as continuous lines, and selection edges shown as dashed lines; (e) Labeling of the graph corresponding to the correct 3-D interpretation.
for the two-face aspect of the cuboid are not shown. Note that there is a special face node label $\perp$, which indicates that the node’s face hypothesis is not selected by the interpretation, i.e., it is deemed to not correspond to the projected surface of a volume. Similarly, there is a special relational node label $\approx$, which indicates that the node’s face hypotheses are accidentally related, i.e., they do not correspond to the projections of adjacent, covisible surfaces in a volumetric part. Finally, because a relational label defines two component face labels, we need a mapping from a relational node label to the two face nodes to which it is attached, as shown by the small red and blue dots on the relational (green) nodes in Figure 6.2(c).

We use a conditional random field (CRF) model to compute the consistent labeling of the graph that yields the 3-D interpretation of the image that best explains the set of 2-D face hypotheses given that they’re projections of surfaces of volumetric parts drawn from the vocabulary. We define a probability distribution over the cliques’ labels conditioned on the shape and image data of their associated face hypotheses, and we aggregate these conditional clique label probabilities into a global probability model for the entire graph label field. The graph labeling that maximizes this conditional probability over the possible labelings yields a face hypothesis selection (i.e., all surviving face nodes with a label different from $\perp$), a 3-D interpretation for them (indicated by their labels), as well as a surface adjacency interpretation (all surviving relational nodes with a label different from $\approx$). Returning to our illustrative example, Figure 6.2(e) shows the labeling of the graph that corresponds to the correct interpretation of the scene.

Finally, in order to efficiently recover the actual pose and parameterization of each volume selected in a labeling, we employ a novel shape-indexing mechanism that learns a mapping between a topological collection of faces (and their shapes) and the surfaces on a particular volume (and its orientation). Indices are represented as high-dimensional vectors encoding the shapes of the faces in the collection. The surface interpretations of the faces and the volume pose is recovered by finding the nearest neighbor of the index
vector in the vector space associated to the volume and aspect deemed as the 3-D interpretation of the face group. We are therefore able to infer, from a set of face hypotheses interpreted as a particular aspect, the identity, parameterization, and orientation of the volume whose surfaces project to the faces in the group.

6.2 Graph Construction

The face relational graph is constructed from a set $\mathcal{H}$ of generated 2-D part hypotheses. However, not all part hypotheses in $\mathcal{H}$ are used to build the graph. In order to keep the size of the generated graphs small, we discard 2-D part hypotheses deemed to have low image support. The set of nodes $V$ is formed by the disjoint union of two sets of nodes, i.e., $V = V^f \cup V^r$, where $V^f$ is called the set of face nodes and $V^r$ is the set of relational nodes. There is a face node for each part hypothesis in $\mathcal{H}$ deemed to be supported by image evidence. (See Section 6.2.3 for details on how image support of 2-D part hypotheses is estimated.) There is a relational node for each pair of proximal and non-overlapping part hypotheses in $\mathcal{H}$ deemed as non-accidental. (See Section 6.2.4 for details on how part hypotheses are paired into relational nodes.) Such pairs of part hypotheses can potentially correspond to projections of adjacent surfaces on the same volume, so the purpose of the relational nodes is to allow for the grouping of part hypotheses into aspects of 3-D parts. (See Section 6.2.4 for a detailed discussion.) A function $\mathbb{H} : V \rightarrow 2^\mathcal{H}$ is defined over the set of nodes such that $\mathbb{H}(v)$ specifies the set of 2-D part hypotheses associated with node $v$. (Notice that $|\mathbb{H}(v)|$ is 1 when $v$ is a face node or 2 if $v$ is a relational node.)

Two disjoint sets of edges exist in the face relational graph. The set $E^{ff} \subset V^f \times V^f$ of selection edges contains edges connecting face nodes that represent part hypotheses that compete to explain roughly the same image data (i.e., their associated closed contours have a high area and contour overlap). The other type of edges are called relational
edges. Two different nodes \(v\) and \(v'\) are connected by a relational edge if and only if 
\(H(v) \cap H(v') \neq \emptyset\). (Notice that therefore there are no relational edges between face 
nodes.) Thus \(E = E^{ff} \cup E^{fr} \cup E^{rr}\), where \(E^{fr} \subset V^f \times V^r\) is the set of relational edges 
between a face and a relational node, and \(E^{rr} \subset V^r \times V^r\) is the set of relational edges 
between relational nodes.

6.2.1 Labels

A node’s label specifies a 3-D interpretation of the 2-D part hypotheses represented by 
the (face or relational) node. Let \(V\) be the set of all volumes in the input 3-D part 
vocabulary, let \(A\) be the set of aspects for all volumes in \(V\), and let \(F\) be the set of 
faces for all aspects in \(A\). Labels are represented as a 3-tuple \((v, a, F) \in V \times A \times F\) 
indicating a particular volume \(v\) from the input 3-D part vocabulary, a specific aspect \(a\) 
of the volume, and a list \(F\) of the particular faces in aspect \(a\) to be matched to the part 
hypotheses. In the case of a face node label, \(|F| = 1\), and in the case of a relational label, 
\(F\) contains two adjacent faces in the aspect. The set of possible labels for a node is thus 
determined by the geometry of the 3-D part models in the input vocabulary. Figure 6.3 
shows an example of the list of possible labels for a simple 3-D part vocabulary formed 
by a cuboid, a cylinder, and an ellipsoid, without deformations (except for changes in 
dimension).

Besides the vocabulary-induced labels, there are two special labels. Face node label 
“\(\perp\)” interprets the node’s corresponding 2-D part hypothesis as not being a projection 
of any relevant scene surface, i.e., effectively not selecting the node’s part hypothesis for 
the final scene’s interpretation. Relational node label “\(\sim\)” indicates that either the two 
part hypotheses (spanned by the relational node) are unrelated (not belonging to the 
same volume) or one or more of them are labeled “\(\perp\)”. Nodes labeled “\(\perp\)” or “\(\sim\)” are 
referred to as unselected, while the remaining nodes are referred to as selected.

Let \(L\) designate the set of all labels. Now \(L = L_f \cup L_r\), where \(L_f \subset V \times A \times F\) is the
set of face node labels, and $L_r \subset V \times A \times (F \times F)$ is the set of relational node labels. We define functions $V : L \to V$ and $A : L \to A$ to retrieve the identities of the volume and aspect in a label, respectively. The list of faces in a label is obtained through two different functions, depending on the type of label. For face labels, function $F_f : L_f \to F$ returns the single face in the label. In the case of a relational label, the assignment of faces to the relational node’s represented part hypotheses is dependent on the order given to the part hypotheses in the definition of the node, so the function to get face assignments to part hypotheses in a relational node takes the form $F_r : L_r \times V_r \times H \to F$, where the last parameter of the function is one of the two part hypotheses participating in the relational node given as its second parameter.
Label consistency is enforced locally on cliques of size 2. Recall that each face node represents a part hypothesis, and a relational node represents the part hypotheses of the two face nodes adjacent to it. Consider a 2-clique formed by a face node \( v \) (associated with face hypothesis \( h \)) and an adjacent relational node \( w \). A labeling of such a clique is locally consistent if either \( w \) is unselected, or hypothesis \( h \) is interpreted as the same face, aspect, and volume by the labels of nodes \( v \) and \( w \). In the case of a clique formed by two adjacent relational nodes, by graph construction there exists a unique face node adjacent to both relational nodes. A labeling of such a clique is locally consistent if either at least one of the two relational nodes is unselected or the volume and aspect identities corresponding to their labels are the same, their common 2-D part hypothesis is interpreted as the same aspect’s face, and their two non-common part hypotheses are interpreted as different aspect faces.

Since the semantics of a selected relational node specify that its two corresponding part hypotheses are projections of surfaces in the same volume, then each connected component of selected (face and relational) nodes corresponds to a group of face hypotheses interpreted as belonging to the same 3-D part instance. Nonetheless, it is possible that non-local inconsistencies arise in these groups; that is, more than one 2-D part hypothesis can be labeled as the same volume’s aspect face. Section 6.4 explains how inconsistencies in a connected component are dealt with.

### 6.2.2 Aspect Representation

The aspect of a volumetric part plays two critical roles in our framework. It specifies the relative geometries of the faces making up the aspect, providing a model against which a collection of proximal face hypotheses can be compared (or tested), i.e., a model which can be used to estimate the conditional probability of a graph clique’s label given the 2-D shape and image data associated to its participating face hypotheses. It is also precisely these relative geometries of an aspect’s faces from which the shape and pose parameters
of the volumetric part are inferred. We therefore seek a vector representation (enabling efficient nearest-neighbor search) of an aspect that is invariant to translation, planar rotation, and scale. Moreover, we want the mapping between our vector representation to be distance-preserving in the sense that similar 3-D shapes and viewpoints yield similar (i.e., close) vectors and vice versa, and to be continuous in that small perturbations to the relative orientation, translation, and scale between the projected faces in an aspect yield small perturbations in the vector. Finally, we want the mapping to be flexible so that it can be applied to any arbitrary vocabulary of 3-D parts and their aspects.

To meet these representational needs, we introduce a novel aspect representation, called the aspect signature vector, as illustrated for a two-face aspect in Figure 6.4. Formally, let \( F = f_1, \ldots, f_K \) be the list of faces for which an ASV is to be generated \( (K \leq 3) \). Let \( d \) be the direction of face contour traversal (i.e., clockwise or counter-clockwise) fixed a priori. Let \( C_k \) be an ordered list of \( T \) equidistantly sampled points along the contour of face \( f_k \) (in the order resulting from traversing the contour in direction \( d \)) and let \( C = C_1, \ldots, C_K \). The number \( T \) of sampled contour points is fixed a priori and remains the same for all faces and ASVs. \( (T = 12 \text{ in the example of Figure 6.4}) \) This determines a constant distance \( \delta_k \) between adjacent sampled contour points within each face \( f_k \). An ASV consists of the rasterization of (a cyclical rotation of) the coordinates of points in \( C_1, \ldots, C_K \) (in that order) expressed in a canonical 2-D coordinate system determined by the same set of points. The bottom-right of Figure 6.4 illustrates the rasterization of one two-face aspect from among its many instantiations shown immediately above.

The origin and orientation of the coordinate system used to compute the ASV is computed from averages of contour point coordinates, providing robustness to noise. The origin \( o \) of the coordinate system is set to the average center of mass of all faces in \( F \), and the coordinate system is oriented such that its x-axis extends from \( o \) towards the center of mass of face \( f_1 \). (For aspects in which \( ||\vec{r}|| \) is too low or when \( K = 1 \), an arbitrary orientation is used. At the time of searching for nearest neighbors, a query
ASV is generated for a number of discrete orientations – 64 in our implementation – of the query view, and the closest of the nearest neighbors of the rotations is chosen.) The sampled points in each list \( C_k \) are cyclically rotated such that the first point in each list in the signature is the one that has a non-negative polar angle (from the face’s center of mass in direction \( \vec{x} \)) closest to zero. (In case of ties, the point closest to the face’s center of mass is selected.) Finally, the dimensionality of the ASVs is reduced using PCA, as more than 99% of the total variance of the ASV spaces generated in our implementation is explained by the top ten or fewer components; in more than half of the spaces, three or fewer components were enough. Figure 6.4 illustrates how the rasterization of an aspect’s...
boundaries leads to an ASV, whose dimensionality is then reduced.

Point coordinates in an ASV are scaled so that the Euclidean distance between two nearby ASVs approximates a scale-invariant measure of shape dissimilarity in terms of the proportion of total face area that the misalignment between contours amounts to and such that the unit of distance is the same for all ASV spaces. Each signature point is scaled by a factor that depends on the face where the point is from. Specifically, points from face $f_k$ are scaled by factor $\xi_k = \frac{\delta_k}{\sum_{k=1}^{K} A_k} \sqrt{KT}$, where $A_k$ is the area of face $f_k$. (See Appendix B for a proof.)

In order to generate the ASV spaces, we discretely sweep the space of parameters and viewpoints of each volume in the vocabulary. Specifically, for each volume in the vocabulary, for each of its parameterizations, for each of its possible aspects, and for each $k$-face subset of the faces comprising the aspect, (with $k \in \{1, 2\}$), we compute an ASV for each possible 3-D volume orientation that yields the aspect. The ASVs for each distinct face subset are stored in a geometric database, which can be efficiently queried to yield nearest neighbors. Figure 6.4 shows how for all possible viewpoints of a cube, the ASVs computed for the 2-face subsets with the label “(cube, aspect with three visible faces, front and side faces)” are stored in a geometric database. We thus end up with a collection of ASV spaces, each representing a different labeling (i.e., 3-D interpretation) of a graph’s clique.

We refer to the set of abstract face hypotheses associated to a clique as the clique’s face configuration. Given a face configuration $Q$ and a clique’s label $L$ involving as many faces as $|Q|$, the distance between the ASV of $Q$ and its nearest-neighbor in the ASV space $L$ is called the interpretation distance of $Q$ under $L$ and is denoted $d_L(Q)$. The set of distances between corresponding contour points on the un-rasterization of the ASV of $Q$ and the un-rasterization of its nearest-neighbor in the ASV space $L$ is called the interpretation distance set of $Q$ under $L$ and is denoted $D_L(Q)$. Figure 6.5 shows an example of the computation of the interpretation distance of a two-contour configuration
Chapter 6. Recovering Abstract Volumetric Parts

under the label interpretation “(cube, aspect with three faces, front and side faces)”. The face configuration’s ASV is computed in (1) (shown as a magenta point in the ASV space), and the interpretation distance is computed in (2) as the distance between the ASV and its nearest neighbor (green point) in the ASV space. The interpretation distance sets of a clique’s face configuration under all possible clique labelings are used as features in the determination of the conditional clique’s label probability.

By associating each ASV in a label’s ASV space to the pose and parameters of the volume used to generate it, we are able to retrieve the closest 3-D model projecting to a given face configuration when that label is assumed as the correct interpretation of the face configuration. Figure 6.5 (3) shows the recovery of the closest volume view for a query contour configuration under the space’s interpretation.
Chapter 6. Recovering Abstract Volumetric Parts

Top PCs explaining 99% of total variance

ASV Spaces

1

2

3

4

5

6

10

Figure 6.6: ASV spaces by the number of top principal components explaining at least 99% of their total variance.

Dimensionality Reduction of ASV Spaces

In our experiments, we sample each face contour using $T = 64$ equidistant points. That yields signature vectors with 128, 256, and 384 dimensions. Since the shape signatures are a rasterization of sets of 2-D points projected from 3-D surfaces with rather simple geometries and deformed by a few parameters, a linear embedding of the space of ASVs is expected to have a rather low dimensionality. Indeed, by applying principal component analysis, 99% of the total variance of the ASV spaces generated in our implementation is explained by the top ten or fewer components. Figure 6.6 lists the ASV spaces by the number of top principal components explaining at least 99% of their total space variance. (Notice that symmetric and topologically equivalent ASV spaces are not listed.) Since the linear embeddings have such a low dimensionality, nearest neighbor search can be efficiently performed using k-d trees. In our implementation, we employed the fast approximate nearest neighbor search implementation of Mount and Arya [116].
6.2.3 Face Nodes

To avoid graphs of excessive size, face nodes are generated for part hypotheses in $H$ only if they are estimated to have sufficient image support. This evaluation is based on the shape of the part, image data, and the set of 3-D priors in the input vocabulary. Even though the generation of the set $H$ of 2-D parts may have included a false positive filtering step based on image and shape features, such a step was based only on 2-D shape priors. At this point, we are also taking into account the agreement in shape between the 2-D part and the projected faces of the volumetric models in the vocabulary.

The image support of 2-D parts was determined by an AdaBoost binary classifier trained using Random Undersampling [151] on image- and shape-based features extracted from a set of 159 manually annotated images. Training examples consisted of all detected consistent cycles in each image and their 2-D abstractions. Those cases corresponding to the projected faces of abstract 3-D parts formed the set of positive cases. In order to increase the amount of training data, synthetic examples were also generated by applying small random perturbations to image consistent cycles and then computing their 2-D abstractions. Since the number of positive cases (i.e., 2-D part hypotheses with actual image support) in each image was the minority, we obtained a very imbalanced training set. For this reason, we employed a random undersampling approach to train the classifier in order to alleviate the class imbalance.

The set of of shape-based features used for classification attempts to capture the agreement in shape between the part hypothesis and any of the possible 3-D interpretations for the single face. These are based on the ASV of the abstract 2-D part hypothesis $Q$ as well as the ASV of its associated consistent cycle $C$, and include the mean, median, and standard deviation of $D_L(Q)$ and $D_L(C)$ as well as normalized 4-bin histograms of $D_L(Q)$ and $D_L(C)$ for each possible label $L$ of one-face cliques. In the case of two aligned similar shapes, the function (with one parameter for contour position) corresponding to the distance between points along one of the contours and their closest points on the
other contour is expected to be a small varying function, even in the case of a small misalignment between the two shapes. This variation is harnessed by features corresponding to the mean and median of the derivatives of $D_L(Q)$ and $D_L(C)$ (when the distance sets are considered as functions, induced by the ordering of the points along the contour).

Features capturing the image evidence for the hypothesis validity attempt to measure expected properties of an actual 2-D boundary, including contour saliency and a low boundary complexity (in opposition to one induced by noise). Contour saliency is measured by the generalized boundary measure of Leordeanu et al. [89] and the average strength along the cycle’s contour. Features assessing boundary complexity by measuring its sinuosity include: 1) the average number of changes in “curvature sign” along the consistent cycle, approximated by the average number of changes in angle sign between consecutive polyline segments in a cycle’s polyline approximation computed at a scale significantly smaller than the cycle’s proper scale; 2) the mean absolute angle between consecutive polyline segments in the polyline approximation; and 3) the integral of absolute angles between consecutive polyline segments at curvature sign change points weighted by the inverse of the lengths of the incident polyline segments, normalized by the contour’s perimeter and the integral of absolute angles over the entire contour. (This last feature focuses on rapid changes in curvature sign and takes into account the amount of change in contour curvature at inflection points.)

The performance of the classifier trained on single faces is shown in Figure 6.7 (a). This ROC was generated using 10-fold cross-validation on the training set. The ROC curve shows the high ambiguity present in attempting to determine the validity of an abstract face hypothesis only from local image information and 3-D model data. In the evaluation of our approach in Section 6.5, we’ll see that recovered single faces not supported by any adjacent face are more prone to be false positives.
6.2.4 Relational Nodes

The purpose of relational nodes in the face relational graph is to relate 2-D part hypotheses that can potentially correspond to projections of adjacent visible surfaces on the same volume so that graph labelings explicitly yield these part hypothesis groups.

Generation of Relational Nodes

In Chapter 3, we measured 2-D part compatibility, i.e., the degree to which two part hypotheses fit together well, in terms of the area of intersection of their abstraction tolerance bands. (See Section 3.3.) In the present chapter, we use a more elaborate approach involving various shape and image features. Since two adjacent visible surfaces on the same volume project onto shapes that are proximal and non-overlapping, all pairs of proximal 2-D part hypotheses without a significant overlap are initially considered as candidates for a relational node. For each pair of 2-D part abstraction hypotheses in this initial set, we need to estimate the likelihood of the two parts being the projections of two adjacent surfaces of the volumetric abstraction of a scene’s 3-D part. From Table A.1, we see that for our volumetric vocabulary, there are only three valid abstraction shape-family combinations, namely, (ellipse, bent-rectangle), (ellipse, bent-tapered-rectangle), (ellipse,
Figure 6.8: 2-D Part contour segmentation (see text for discussion).

annuloid)\(^1\) and (parallelogram, parallelogram). Only when at least one valid combination of abstraction hypotheses exists for a pair of nearby parts are they considered for part adjacency. Furthermore, we will generate relational nodes only for pairs of 2-D parts that are likely to be non-accidental groupings. We determine non-accidentalness of a candidate pair by means of an AdaBoost binary classifier trained using Random Undersampling on image and shape features as well as features based on the geometric relationship between the two 2-D shapes extracted from a set of 159 manually annotated images. As in the case of single face hypotheses, the ground truth adjacent face pairs were in the minority, and so random undersampling was used to deal with the imbalance in the number of training cases for each class.

The features used to measure contour alignment between the parts are automatically computed based on a natural segmentation of the 2-D abstraction contours into rectilinear and curvilinear contour fragments, according to their shape-family. Ellipses and bent-ellipses consist of a single curvilinear fragment, parallelograms and tapered-rectangles have four rectilinear fragments, and bent-rectangles and bent-tapered-rectangles are formed by two rectilinear and two curvilinear fragments (see Figure 6.8). The 2-D part hypotheses are automatically segmented into curvilinear (red) and rectilinear (blue) frag-

\(^1\)We did not include annuloids in our implementation, which correspond to close-to-degenerate views of tapered cylinders.
ments at contour normal discontinuity points (green)\(^2\). To assess the contour alignment between pairs of 2-D part abstractions, features are computed on pairs of fragments from different 2-D parts. The employed features are based on the type of fragment, the distance between them, their relative lengths, and the angle between the normals at corresponding points.

Let \( s(f) \) denote the proper scale of the 2-D part of which \( f \) is a contour fragment. Let \( q_{f_1}^{f_2}(p) \) denote the point along fragment \( f_2 \) that is closest to point \( p \) on fragment \( f_1 \). Let \( \text{cover}_{f_1}^{f_2} = |\{ q_{f_1}^{f_2}(p) : p \in f_1 \}|/|\{ q \in f_2 \}| \) denote the coverage of fragment \( f_2 \) by points in fragment \( f_1 \). Let \( D_{f_1}(f_2) \) denote the set of distances between each point along fragment \( f_1 \) and its closest point along fragment \( f_2 \), i.e., \( D_{f_1}(f_2) = \{ \| p - q_{f_1}^{f_2}(p) \| : p \in f_1 \} \). Let \( d_{f_1}^{f_2}(f_2) \), with \( T \in \{ \text{min}, \text{max}, \text{mean} \} \), denote the minimum, maximum, and mean value over \( D_{f_1}(f_2) \), respectively. Let \( \sigma_{f_1}(f_2) \) denote the standard deviation of values in \( D_{f_1}(f_2) \), and let \( \cos_{f_1}(f_2) \) denote the mean cosine of the angles between the unit normal vectors at each point in \( f_1 \) and the unit normal vectors at their closest point in \( f_2 \). The features employed are:

1. type of fragments (i.e., rectilinear, non-closed curvilinear, ellipse)

2. proportion of each fragment that is covered by the other fragment, i.e.,
   \( \text{cover}_{f_1}^{f_2} \),

3. mean cosine of the angle between normal vectors at corresponding fragment points,
   in the case of two rectilinear fragments, \( (\cos_{f_1}(f_2) + \cos_{f_2}(f_1))/2 \),
   in the case of an ellipse \( f_e \) and a curvilinear fragment \( f_c \), \( \cos_{f_c}(f_e) \),

4. minimum, maximum, and mean distance between the fragments normalized by the minimum scale of the two parts.

\(^2\)Although, in general, automatic segmentation of general curves is an open problem, that is not our case since we are dealing with ideal 2-D shapes.
in the case of two rectilinear fragments, \( (d_{f_1}^T(f_2) + d_{f_2}^T(f_1))/(2\times\min(s(f_1), s(f_2))) \),

in the case of an ellipse \( f_e \) and a curvilinear fragment \( f_c \), \( d_{f_e}^T(f_c)/\min(s(f_e), s(f_c)) \),

5. standard deviation of distances between fragments normalized by the mean distance,

in the case of two rectilinear fragments, \( \sigma_{f_1}(f_2)/d_{f_1}^{\text{mean}}(f_2) + \sigma_{f_2}(f_1)/d_{f_2}^{\text{mean}}(f_1)/2 \),

in the case of an ellipse \( f_e \) and a curvilinear fragment \( f_c \), \( \sigma_{f_c}(f_e)/d_{f_e}^{\text{mean}}(f_e) \),

6. ratio between fragment lengths,

in the case of two rectilinear fragments, the ratio of the length of the shortest fragment to the length of the longest one,

in the case of an ellipse, the ratio of the distance between the end-points of the curved fragment to the length of the ellipse’s major axis,

7. minimum, maximum, and mean normalized distance between fragment end-points,

in the case of two rectilinear fragments, the distances between each end-point of one fragment and its closest end-point along the other fragment, normalized by the length of the shortest fragment,

in the case of an ellipse and a curvilinear fragment, the distances between each end-point along the curvilinear fragment and its closest point on the ellipse normalized by the distance between the curvilinear fragment’s end-points,

8. agreement in contour curvature between an ellipse and a curvilinear fragment computed as the ratio of the distance between the center point of the curvilinear fragment and the line through its end-points to the length of the ellipse’s minor axis,

9. angle between fragments,

in the case of two rectilinear fragments, the absolute value of the cosine of the angle between the fragments,
in the case of an ellipse and a curvilinear fragment, the absolute value of the cosine of the angle between the ellipse’s major axis and the line through the endpoints of the curvilinear fragment.

To assess image support for the pair of face hypotheses as well as the agreement in shape between the 2-D parts and the projected faces of their possible 3-D interpretations, a set of features similar to the ones described in Section 6.2.3 is used, but with \( L \) living in the space of two-face clique labels. Figure 6.7 (b) shows an ROC (computed using 10-fold cross-validation on the training set) with the performance of the classifier trained to detected non-accidental adjacent face groups. This classifier performs better than the one for single faces. This is an expected behavior given that this is a less local decision, based on supporting contextual information from the presence of adjacent face hypotheses with good alignment and image support, thus reducing the chance of false positives.

**Need for Relational Nodes**

Consider the example of Figure 6.9, where the need for relational nodes is demonstrated. (Assume the same 3-D part vocabulary and labels from Figure 6.3.) In (a), we see a set of generated 2-D part hypotheses, (b) shows the “face relational graph” for the part configuration in (a), and (c) shows the “face adjacency graph” for the part configuration in (a). This is a graph in which there is a node for each 2-D part hypothesis, and nodes are connected by a relational edge if and only if their corresponding part hypotheses can potentially correspond to projections of surfaces of the same volume.

A particular labeling of the face adjacency graph is shown in Figure 6.9 (d), but since this graph’s labeling does not indicate which faces are in the same volume, or if all arise from different volumes, there are various possible 3-D scene interpretations that are consistent with this labeling. In the case of the example’s face relational graph, each possible graph labeling interpreting face nodes as in (d) has an unambiguous interpretation. This is because the grouping of faces into volumes is explicitly stated by the labels
of the relational nodes. The only three possible locally consistent labelings of the face relational graph are shown in (e-g), each with its corresponding unique interpretation of how the part hypotheses are to be grouped into different volumes. (Notice that the interpretation that the three part hypotheses are projections of surfaces from the same cuboid is inconsistent with the face node labels used in the example, as the top and bottom part hypotheses in (a) are both interpreted as the same aspect face by their face labels.)
6.3 Grouping Faces into Aspects

As described in Section 6.1, we formulate the problem of grouping face hypotheses into volumetric part aspects as a graph labeling problem in which labels are assigned to the face and relational nodes in the face relational graph. Finding the best volume-aspect interpretation of the image is equivalent to finding the most probable labeling of the graph conditioned on the image evidence (in the form of a collection of generated face hypotheses and their associated image data) and the shape priors induced by the 3-D input vocabulary. We model the graph labeling problem using a conditional random field (CRF), and seek to maximize the conditional probability over the label field \( L \) given the input set \( S \) of 3-D shape priors and the input set \( H \) of 2-D face hypotheses, specified by the following log-linear model:

\[
p(L|S,H; \Lambda) = \frac{1}{Z(S,H)} \prod_{c \in C} \exp \left\{ \sum_{k=1}^{K_c} \Lambda_{ck} f_{ck}(L_c, S, H) \right\},
\]

where \( Z(S,H) \) is the partition function of the distribution, \( C \) is the set of cliques in the face relational graph, \( K_c \) is the number of features for clique \( c \), \( f_{ck} \) are feature functions that measure the consistency and probability of a labeling of the face hypotheses in a clique given the set of face hypotheses and 3-D priors, and \( \Lambda \) are weight parameters.

To keep inference and parameter learning tractable, we simplify the probability model of Equation 6.1 by considering only cliques of size at most two. There are five possible types of such cliques in the face relational graph, i.e., cliques formed by a single face node, by two face nodes, by a single relational node, by a face node adjacent to a relational node, and by two adjacent relational nodes. We note these types of cliques as \( f, ff, r, fr, \) and \( rr \), respectively. We also set parameters \( \Lambda \) to only depend on the type of clique and feature function, and so they are the same for all cliques of the same type.

We employ four types of feature functions in our model, namely \( \phi, \bar{\phi}, \xi, \) and \( \psi \). Features \( \phi(L, S, H) \), defined over cliques of type \( f, r, \) and \( rr \), score the quality of the clique’s labeling based on the part hypotheses and their associated image data as well as
shape prior information. Features $\phi(L)$, defined as binary indicators of type $f$ clique’s labels being different from $\perp$, are used to penalize the total number of selected hypotheses in a graph interpretation. Features $\xi(L_1, L_2)$, defined over type $ff$ cliques (i.e., two face nodes connected by a selection edge), prohibit interpretations selecting more than one face hypothesis competing to explain the same image data by having a value of minus infinity in that case, and zero otherwise, i.e.,

$$
\xi_{ff}(L_{v1}, L_{v2}) = \begin{cases} 
0, & L_{v1} = \perp \lor L_{v2} = \perp \\
-\infty, & \text{otherwise.} 
\end{cases}
$$

Features $\psi(L, S, H)$ enforce the local label consistency of cliques of types $fr$ and $rr$ by having a value of zero in cases of consistent labelings, and minus infinity otherwise. A labeling of a type $fr$ clique is consistent if either the relational node is labeled $\sim$ or if the face node label and relational node label assign the same volume, aspect, and face interpretation to the nodes’ common face hypothesis, i.e.,

$$
\psi_{fr}(v, w, L_v, L_w) = \begin{cases} 
0, & L_w = \sim \lor \\
& \left( L_v \neq \perp \land V(L_v) = V(L_w) \land A(L_v) = A(L_w) \land \right. \\
& \left. F_f(L_v) = F_r(L_w, w, \mathbb{H}(v)) \right)
\end{cases}
$$

$$
\psi_{fr}(v, w, L_v, L_w) = \begin{cases} 
0, & L_w = \sim \lor \\
& \left( L_v \neq \perp \land V(L_v) = V(L_w) \land A(L_v) = A(L_w) \land \\
& \right. \\
& \left. F_f(L_v) = F_r(L_w, w, \mathbb{H}(v)) \right)
\end{cases}
$$

where $\lor$ is the “exclusive or” operator. A labeling of a type $rr$ clique is consistent if either at least one of the two relational node labels is $\sim$ or if the contour hypothesis common to both nodes is assigned the same volume, aspect, and face interpretation by both relational node labels and the two face hypotheses non-common to both nodes are

---

3In our implementation, we only consider for each node the set of labels that are consistent with the shapes of the 2-D parts associated with the node. For example, for a face node associated with a quadrilateral face, only labels corresponding to quadrilateral aspect faces are considered, and for a relational node associated with an elliptical face adjacent to a bent-rectangular face, only the label corresponding to the 2-face aspect of the cylinder is considered.
interpreted as different volume faces, i.e.,

\[
\psi_{rr}(w_1, w_2, L_{w_1}, L_{w_2}) = \begin{cases}
0, & \left( L_{w_1} = \infty \lor L_{w_2} = \infty \right) \\
\left( \forall (L_{w_1}) = \forall (L_{w_2}) \land \mathcal{A}(L_{w_1}) = \mathcal{A}(L_{w_2}) \land \\
\mathcal{F}_r(L_{w_1}, w_1, \mathbb{H}(w_1) \setminus \mathbb{H}(w_2)) \neq \mathcal{F}_r(L_{w_2}, w_2, \mathbb{H}(w_2) \setminus \mathbb{H}(w_1)) \right)
\end{cases}
\]

\[\neg \infty, \text{ otherwise.} \tag{6.4}\]

Our probability model thus becomes:

\[
p(L|S, H; \Lambda) \propto \exp \left\{ \sum_{h \in C_f} \left[ \lambda_f \phi_f(L_h) + \lambda_f \phi_f(h, L_h) \right] + \right. \\
\sum_{h \in C_r} \lambda_r \phi_r(h, L_h) + \sum_{h \in C_{rr}} \left[ \lambda_{rr} \phi_{rr}(h, L_h) + \psi_{rr}(h, L_h) \right] + \\
\sum_{h \in C_{ff}} \xi_{ff}(L_h) + \sum_{h \in C_{fr}} \psi_{fr}(h, L_h) \right\}, \tag{6.5}\]

where \(C^x\) is the set of face hypothesis configurations of cliques of type \(x\), \(L_h\) is the labeling of face hypothesis configuration \(h\) as defined by \(L\), and the model parameters are \(\Lambda = \{ \lambda_f, \lambda_r, \lambda_{rr} \}.\)

We define features \(\phi_f(v, L_v)\) and \(\phi_r(w, L_w)\) as the conditional log-probabilities of their corresponding 1-node-clique’s labelings. To compute them, we used two multi-class AdaBoost classifiers over the same set of features described in Sections 6.2.3 and 6.2.4, respectively, and trained on the same set of 159 manually annotated images. These classifiers were trained only for labels different from \(\perp\) or \(\infty\). Formally,

\[
\phi_f(v, L_v) = \begin{cases}
\log p_b^f(\perp), & L_v = \perp \\
\log p_b^f(\mathcal{X}) + \log p_{mc}^f(L_v), & L_v \neq \perp
\end{cases}
\]

and

\[
\phi_r(w, L_w) = \begin{cases}
\log p_b^r(\infty), & L_w = \infty \\
\log p_b^r(\sim) + \log p_{mc}^r(L_w), & L_w \neq \infty
\end{cases}
\]

where \(p_b^t\) and \(p_{mc}^t\), \(t \in \{f, r\}\) are the probability values estimated by the binary and
multi-class classifiers, respectively. Feature \( \phi^{rr}(w_1, w_2, L_{w_1}, L_{w_2}) \) is defined as

\[
\phi^{rr}(w_1, w_2, L_{w_1}, L_{w_2}) = \begin{cases} 
-\delta_{L}(Q_{w_1w_2}) & L_{w_1} \neq \approx \land L_{w_2} \neq \approx \\
0, & \text{otherwise,}
\end{cases}
\]

where \( Q_{w_1w_2} \) is the face configuration of the three 2-D part hypotheses involved in the \( rr \) clique \( w_1w_2 \). This feature is based on the agreement in shape and geometrical relationship between the face hypotheses and the projected faces of a 3-D model in the vocabulary. It is defined such that it penalizes the simultaneous selection of two adjacent relational nodes jointly explaining three faces of an aspect when the shape spanned by the three abstract faces deviates from the shape of the projected faces of an abstract volume in the space associated with the labels.

Tables 6.1 and 6.2 show the confusion matrices for the face and relational label classifiers, where each row corresponds to the ground truth class, and each column corresponds to the predicted class. These tables were computed using 10-fold cross-validation on the training set. Notice that a single class was devoted to all elliptical faces, since in the absence of contextual information, there is no way to disambiguate their identity given only local image data. The 1-face aspects not included in Table 6.1, corresponding mostly to degenerate views, are cases for which there was not sufficient training data, and so their corresponding 3-D interpretations were not considered in the label set. Besides degenerate viewpoint cases missing from Table 6.2, the two-face aspect of the tapered-cylinder sitting on its larger circular face is not included due to lack of examples of this kind in our training set. Cases that are topologically equivalent to those in the table (e.g., front and side faces of the 3-face aspect of the cube) are not included.

We can see the interpretation ambiguity present in the case of single faces. In some cases, the confusion is probably due to an abstraction of the incorrect class, like an elliptical face being confused with a romboidal face, or a bent-rectangular face with little bending being confused with a rectangular face, or a bent-ellipsoid face with almost no bending being confused with an elliptical face. In other cases, the confusions are
more justifiable, like bent-rectangular faces, bent-tapered-rectangular faces, and bent-ellipsoidal faces being confused with each other. In the case of relational nodes, the performance of the classifier is almost perfect. Only very little ambiguity is present, and in most cases it is understandable, like between a cylinder and a tapered-cylinder.

6.3.1 Labeling Optimization and Parameter Learning

The generated graphs for the images in our training and test datasets are rather small. (See middle column of Figures 6.12 – 6.20.) The total number of consistent labelings of a graph is highly constrained by their small size, the modest number of relevant interpretations for each node, the large number of constraints yielded by selection edges, and the consistency requirements between adjacent edge labels. Since the resulting number of consistent graph labelings is small, we exhaustively generate all of them via a dynamic programming approach. Perfect inference is thus possible by computing the unnormalized
probability (i.e., ignoring the partition function) of each graph labeling and picking the
one maximizing it. The small number of graph labelings also makes parameter learning
via maximum likelihood estimation simple, since it is possible to compute the partition
function in every case.

6.4 From Labelings to 3-D Segmentations

An inferred graph labeling is free from local\textsuperscript{4} label inconsistencies due to the definition of
the constraint feature functions $\psi$ in the probability model. Consider now the graph $\hat{G}$
induced by a labeling $L$ of a graph $G$, and remove from $\hat{G}$ any node that is unselected by
$L$ in $G$ (i.e., any node labeled $\perp$ or $\nmid$) as well as any edge incident to an unselected node.
The constraint functions guarantee that each remaining connected component of face
and relational nodes in $\hat{G}$ corresponds to face hypotheses that are interpreted by labeling

\textsuperscript{4}That is, within cliques of size at most two.
as projections of adjacent, covisible surfaces on a 3-D abstraction of an object part in the scene. Thus a graph labeling effectively groups subsets of selected face hypotheses into 3-D parts.

The actual 3-D interpretation (i.e., a volume’s identity, parameterization, and pose) of the face configuration of a connected component is recovered via the ASV space of the component’s labeling. This is done by retrieving the parametrization and viewpoint of the 3-D vocabulary parts that yield the $K$ nearest neighbor ASVs of the component’s face configuration ASV. From these candidates, the volume achieving maximum compactness is selected as the volumetric interpretation. The compactness of a 3-D shape is calculated as $\frac{V^2}{S^3}$, where $V$ and $S$ are the volume and area of the 3-D shape, respectively. This strategy of selecting the most compact volume that produces the percept ensures that in the case of an ambiguous percept, we choose the 3-D interpretation that is more likely to coincide with human visual preferences, as demonstrated by Pizlo, Scheessele, and Stevenson [132, 133].

Although an inferred graph labeling is free from local label inconsistencies, it is still possible for inconsistencies to arise in a label-induced connected component (whose nodes are interpreted as aspect faces in the same volume) when “distant” nodes are interpreted as the same aspect face. Notice that the impact of labeling inconsistencies is thus limited to the connected component where they arise. From an inconsistent labeling of a connected component, we can generate all possible maximally consistent labelings, that is, labelings of a subset of nodes in the connected component that correspond to a maximal number of interpreted aspect faces such that each face is interpreted by exactly one part hypothesis.

In general, the face interpretation assigned by a maximally-consistent labeling to a set of parts may not be in good correspondence with any of the views of the corresponding volume surfaces. This is illustrated in Figure 6.10 (a), where a configuration of three part hypotheses and a labeling of their face relational graph is shown. The parts are
interpreted as faces in a cuboid, however, the topology of the three parts, when considered together, does not match well the projection of the surfaces in any view of a cuboid. However, this is not an optimal labeling, because the potential of the $rr$ clique (determined, in particular, by functions $\phi^r$ and $\phi^{rr}$) achieves a higher value if one of the two relational nodes is labeled $\sim$. 

Figure 6.10: Labeling inconsistencies (see text for discussion).
When the labeling of each 1- and 2-clique in a connected component is such that the configuration of represented part hypotheses corresponds to a valid view of the volume surfaces as specified by the labeling, we expect the $\phi$ functions to achieve high probability values and hence a low energy value for the component. We refer to such labelings as \textit{low-energy} labelings. Often, low-energy competing maximally-consistent labelings correspond to roughly similar volumetric interpretations. In the example of Figure 6.10 (b), there are two competing maximally-consistent labelings of an inconsistently-labeled connected component with four nodes. Note that this type of labeling inconsistencies are not only rare (generally disallowed by the presence of selection edges) but they also have a small effect on the final parameterization and viewpoint of the recovered volumetric part for the component because all maximally consistent labelings roughly have the same 3-D interpretation. In these cases, we can select as the best consistent interpretation the component’s maximally-consistent labeling that maximizes the potentials of the component’s cliques.

A sufficient condition for label inconsistencies being limited to just this type is that the set of labels be induced by a 3-D part vocabulary in which the parameterization and viewpoint of each possible volume view is entirely determined (except perhaps for depth) when the projection of any single or pair of connected volume surfaces and their aspect face identities is known. That is, knowledge of the projected boundaries of any one or two connected volume surfaces and their aspect face identities is sufficient to determine the location of the projected boundaries of all other surfaces. This property is generally fulfilled by simple volumetric models, like the ones expected to be used to model the coarse shape of object parts while abstracting away structural detail. That is the case for the volumetric part vocabulary that we used in our experiments (containing cuboids, cylinders, tapered cylinders, ellipsoids, and bent ellipsoids), shown in Figure 6.11.

If the aforementioned property is not true for some volume’s aspect, a low-energy consistent labeling could potentially interpret the involved part hypotheses as volume
Figure 6.11: Instances of volumetric parts sampled from the input 3-D vocabulary used in our experiments.

surfaces in a way that is incompatible with all possible volume views. Figure 6.10 (c)
shows an example of a 3-D part, two possible side views corresponding to the same
volume’s aspect, and a part configuration and its face relational graph with a low-energy
consistent labeling that does not correspond to any valid view of the part model. In this
case, valid volumetric interpretations correspond only to the labeling when restricted to
certain node subsets (e.g., the subset of nodes labeled 1, 2, and 3 or the subset labeled
2, 3, and 4 in the example of the figure). The best labeling can then be determined
by choosing the largest cardinality subset of nodes maximizing the probability of their
volumetric interpretation – specified by the node’s labels – given the configuration of
parts associated with the nodes.

6.5 Evaluation

In order to evaluate our proposed approach, we created an image dataset containing 100
images of real objects extracted from Caltech 101 and Caltech 256, as well as the Internet.
In general, each image contains a non-degenerate view of a single object appearing on a
simple background and with no occlusion, with a number of objects having some texture
and structural detail. The prototypical shape of the selected objects can generally be
described with a single abstract part. Some of the images in the dataset can be seen in
Figures 6.12 – 6.20, and Figures 6.21 – 6.23. The images in the dataset were annotated with ground truth abstract volumetric parts (from a given part vocabulary) corresponding to the objects (or object parts) in the scene. The 3-D part vocabulary used in our experiments and to annotate the dataset images contains cuboids, cylinders, tapered cylinders, ellipsoids and bent ellipsoids. Figure 6.11 shows instances of each type of volume in the vocabulary.

Some qualitative results from our approach are shown in Figures 6.12 – 6.20, where the first column shows the input images, their corresponding face relational graphs appear in the second column, their recovered volumetric abstractions are shown in the third column, and the ground-truth abstractions are displayed in the fourth column. The recovered abstractions correspond to the final output of our system, i.e., they are the set of 3-D volumes corresponding to the optimal labeling of the face relational graph induced by the set of 2-D part hypotheses generated from each image. Face relational graphs are shown to give an idea of the variation in their complexity from image to image, determined by the initial set of recovered face hypotheses and their spatial relationships. Blue and red circles correspond to face and relational nodes, respectively. Blue edges connect face nodes to their adjacent relational nodes, magenta edges connect pairs of relational nodes (i.e., involving two face adjacencies sharing a common face hypothesis), and dashed red lines correspond to selection edges between pairs of face nodes involving competing face hypotheses.

The recovered abstract volumes are displayed at their estimated pose on top of each image in the third column. The consistent cycles whose abstractions induced the projected visible faces of the volumes are shown by dashed red, blue, and green lines. In the case of a selected face node that is not adjacent to a selected relational node, depending on the type of the face abstraction, it may be infeasible to infer the volumetric abstraction (i.e., the volume’s identity, its parameters, and/or pose) from the face label alone due to the lack of contextual constraints. That is the case for the abstract parallelogram
and elliptical faces. An elliptical face by itself may correspond to the face of an ellipsoid or the circular face of a cylinder or tapered-cylinder. In these last two cases, the size of the cylinder cannot be determined from the shape of the elliptical face. Similarly, a parallelogram face strongly suggests the presence of a cuboid, but the dimensions and pose of the cuboid are unconstrained in the absence of other selected cube faces. For visualization purposes, we display these two cases in the following way. If the face abstraction is elliptical, we show an ellipsoid as the 3-D abstraction of the face. (Although, as noted above, the elliptical face may very well be the projection of a cylinder’s circular face and not an ellipsoid.) If the abstraction of the face is a parallelogram (and so it can potentially correspond to the projection of a cuboid’s face), only the contour of the 2-D abstraction (i.e., the parallelogram corresponding to the projection of the cuboid’s face) is shown (in magenta).

Notice the ability of our approach to recover volumetric abstractions that do not overlap well with corresponding image contours. (See for example, Figure 6.12 (b), Figure 6.13 (b) and (d), Figure 6.14 (c) and (d), Figure 6.16 (b), Figure 6.17 (a), Figure 6.18 (c) and (d), Figure 6.19 (a), (b), and (c), and Figure 6.20 (b).) Even in cases where the object’s shape is close to that of an ideal volume in the vocabulary, due to image region undersegmentation or noise, the actual detected consistent cycles, which our approach ultimately uses, do not follow well the volume contours. (See Figure 6.12 (d), Figure 6.13 (b), Figure 6.14 (a) and (b), Figure 6.15 (d), Figure 6.17 (a), and Figure 6.18 (a) and (c), for example.) Also, due to foreshortening or because adjacent object surfaces are not orthogonal, the resulting spatial layout of the face hypotheses does not match well the projected faces of the abstracted volume (e.g., Figure 6.12 (c), Figure 6.15 (a), (b), (c), and Figure 6.17 (a), (b).)
6.6 Limitations

Our 3-D abstraction approach expects its input set of 2-D part hypotheses to contain a 2-D hypothesis for at least one face of each projected abstract volume surface in the image, and it fails if that is not the case. In our case, such an input set is obtained as the output of the method of Chapter 4 including the enhancements described in Appendix A. Such an approach, which in itself involves several steps (i.e., consistent cycle detection, non-maximum suppression, 2-D cycle abstraction, and false positive filtering) with a potential for false negatives, also depends heavily on the quality of its input, namely, a region oversegmentation of the input image.

Moreover, the approach expects the existence of a consistent cycle in the region bound-
Fig. 6.13: 3-D Abstraction (cont’d – see text for discussion).

ary graph (induced by its input region segmentation) accounting for each abstract face. However, it is possible that no such cycle exists even in the presence of a “perfect” image region segmentation, i.e., one that is accurate according to the photometric properties of the image. (For example, due to illumination and/or surface properties, it is possible that the 3-D contour induced by the surface normal discontinuity between two surfaces on the object does not produce a photometric discontinuity on the image along the entire 2-D projection of the 3-D contour, and so the projected regions of the two faces will become a single region in the image region segmentation.) The approach is also limited by its abstraction tolerance parameter, which controls the amount of deviation in shape between the projected surface boundary and its corresponding projected face in the abstract model. Setting that parameter to a high value may recover the correct 2-D part
but at the cost of producing too many false positives.

Background clutter and rich texture are also problematic for the approach of Chapter 4, as they have the potential to greatly increase the number of false positives. The presence of regular structure (e.g., a checkerboard pattern) can also induce an exponential set of consistent cycles. The non-locality of the approach due to its relying on the shape of entire closed contours makes it unable to deal with occlusion, because the shape of a cycle around the visible portion(s) of a projected abstract face no longer matches a model in its set of 2-D abstractions. All these shortcomings in the method to generate the input set of 2-D part hypotheses directly affect the performance of the approach presented in this Chapter. Besides these, false negatives in the face and relational node generation steps of our 3-D abstraction method also have a direct impact on its performance.
Figures 6.21 – 6.23 show some examples of the limitations of our approach. In Figure 6.21 (a) and (b), the recovered volume is incorrect due to undetected adjacent faces. In both cases, the 2-D hypotheses for the cylinder side face were missing. Cases (c) and (d) are typical hallucination examples that can be blamed on false positive 2-D parts with a consistent shape and supported by salient image contours. At the same time, the actual 2-D parts were not recovered due to undersegmentation issues. In the case of the pan, the shape of the group of regions corresponding to its side face, involving thin and long curved “arms” surrounding its opening on both sides, was likely deemed as inconsistent. The boundary of the group of regions corresponding to the tambourine’s side face includes very intricate concavities on its left side, which likely yielded an inconsistent cycle.

In Figure 6.22 (a–d), 2-D parts were available, not for entire faces, but for subparts
with an abstract 3-D shape modeled by parts in our vocabulary. The approach only selects a few nodes in the face relational graph of the image in Figure 6.23 (a), although face nodes exist in the graph for all but one of the bananas. This is due to a heavy penalization by the feature function $\varphi(L)$, whose parameter $\lambda_f$ was learned from examples mostly with few ground truth selected face nodes. Finally, in the cases of Figure 6.23 (b–d), the inferred 3-D abstractions (although some are correct) rely on hallucinated noisy contours.
6.7 Conclusions

We have presented a method for selecting and grouping an initial set of 2-D part hypotheses (representing potential abstract volumetric faces) into 3-D volumetric abstractions according to a user-defined vocabulary of abstract volumetric models. Our contributions include a formulation of the 2-D part hypothesis selection and grouping as a graph labeling problem, and a novel shape-indexing mechanism to map a topological collection of faces to abstract volume surfaces. The resulting framework reports promising initial steps on part-based volumetric shape abstraction from images of real objects, establishing a number of important directions for future work.
Figure 6.18: 3-D Abstraction (cont’d - see text for discussion).
Figure 6.19: 3-D Abstraction (cont’d - see text for discussion).

Figure 6.20: 3-D Abstraction (cont’d - see text for discussion).
Figure 6.21: Limitations (see text for discussion).
Figure 6.22: Limitations (cont’d - see text for discussion).
Figure 6.23: Limitations (cont’d - see text for discussion).
Chapter 7

Conclusions

7.1 Review of Contributions

In this thesis, we explore the problem of model-based perceptual grouping and shape abstraction for the purpose of generic object recognition. Unexpected object recognition requires the recovery of generic parts and their relations to support effective indexing into large databases. The goal of this task is to recover the coarse, prototypical, viewpoint-invariant part structure of an object which, in turn, can form a powerful, discriminative indexing structure into a large database of object-centered 3-D models. This is not an attempt to perfect global volumetric recovery, but to recover enough volumetric part and relational part structure from the image to index into the object database in order to reduce the number of candidates to be verified.

While contours may reflect important shape information, a single image contour or fragment may not be generic to a category, and assuming one-to-one correspondence with a model contour can be highly restrictive. However, a collection of local contours may reflect a more abstract regularity that may be shared by many categories. Such abstract parts require not only that a noisy, broken collection is grouped, but also abstracted. The results presented in this thesis demonstrate that the semantic gap between real
scene contours and the abstract parts that make up categorical shape models can be bridged with the help of a small vocabulary of intermediate-level part models\(^1\). Such a vocabulary, which is provided by the user as an input to each framework, is formed by a collection of intermediate-level shape models, and does not assume any knowledge of the objects present in the scene. In particular, the presented results indicate the potential of the proposed framework to recover abstract part structure from images of real objects. If a few correct parts can be identified, they may be sufficient to form a powerful index into a collection of object candidates which, in turn, can be used in a top-down manner to guide part selection and object verification.

In our first contribution, we describe an edge-based framework for contour grouping and abstraction from a single 2-D image, combining mid-level shape priors with a bottom-up part indexing framework that maps contour collections to abstract part models. The proposed method works on images of objects containing some structural detail, unlike most previous methods, which are restricted to silhouettes or scenes of textureless objects. The approach takes as input a hierarchy of edge maps computed at different scales from a single image and a user-defined 2-D part vocabulary. Using a novel translation-, rotation-, and scale-invariant shape indexing structure generated from the part vocabulary, a set of abstract 2-D parts is hypothesized from contour information in the edge maps. The obtained results demonstrate the availability of good hypotheses among those generated by the method. Under the assumption that surfaces that are causally related tend to fit together in the world, we presented a novel surface compatibility prior that, although very generic, was able to effectively prune false positives from the set of generated 2-D part hypotheses.

In our second contribution, we introduce a data-driven region-based approach for

---

\(^1\)The proposed framework assumes that objects can be decomposed into simple parts, but does not depend on a particular part vocabulary, which is an input to the system. Particularly, the presented methods are potentially well-suited for vocabularies where parts are simple and have a high degree of symmetry (as is found in such part models as superquadrics, generalized cylinders, and geons).
grouping contours (region boundaries) into 2-D parts according to a user-defined vocabulary of abstract parts models with a substantially reduced computational complexity than that of our first proposed approach. The approach takes as input an image region oversegmentation of a 2-D image and a vocabulary of simple 2-D part models. Unlike most previous approaches, the presented method works on images of scenes containing lightly textured objects. By means of a novel model-based region grouping approach, guided by learning classifiers on all possible component fragments of model shapes, 2-D part hypotheses are generated for each group of adjacent regions whose merged shape is consistent, at some level of abstraction, with one or more models in the input vocabulary. Through a novel application of an ASM model which captures the entire vocabulary of shapes with a single model and which needs no proper initialization, the hypothesized contours are abstracted and categorized. The obtained results demonstrate the ability of the approach to recover the abstract shape of ground-truth object surfaces despite irregularities in the actual surface contours and minor undersegmentation.

As the degree of abstraction between image contours and abstract parts increases, so too does the ambiguity of a perceptual group of image contours – if abstraction is viewed as a process of “controlled hallucination”, the more you hallucinate, the greater the possible mappings to different parts. In our third contribution, we show that by imposing spatial and spatiotemporal constraints on the grouping process, we can significantly reduce such ambiguity, ensuring greater precision of the recovered abstract parts which, in turn, can facilitate the indexing and recognition of categorical shape models. We introduce a novel probabilistic, graph-theoretical framework to 2-D part hypothesis selection based on spatiotemporal coherence priors when a video sequence, in which there is relative motion between the camera and objects in a scene, is available. Without assuming any object prior, the proposed approach was able to efficiently and effectively select 2-D part hypotheses that correspond to actual object parts across frames, thus achieving a great improvement in the precision and recall of the generation of ground-truth hypotheses,
which significantly outperforms a static image version.

3-D constraints naturally enforce stronger (i.e., specific) part relations (such as part co-occurrence) than generic relations like proximity or alignment. In our final contribution, we strengthen the role of 2-D part relations in the selection process by explicitly including 3-D constraints, with the ultimate goal of recovering a set of abstract volumetric parts from a single image. We present a framework for selecting and grouping an initial set of 2-D part hypotheses (representing potential abstract volumetric faces) into 3-D volumetric abstractions that exploits aspect-based relational constraints according to a user-defined vocabulary of abstract volumetric models. We formulate 2-D part hypothesis selection and grouping as a graph labeling problem, and introduced a novel shape descriptor supporting an efficient indexing mechanism to map a topological collection of faces to abstract volume surfaces. Our method is able to recover the abstractions of 3-D parts from real images of real scenes containing lightly textured objects and some structural detail. The obtained results demonstrate how the presented approach is able to recover abstract 3-D shape in spite of the irregularities in the original image contours.

The results reported by this thesis show how a small vocabulary of intermediate-level shape models is able to drive a perceptual grouping process to generate powerful abstractions without the need to assume any knowledge of the objects present in the scene. These are promising initial steps towards part-based shape abstraction from images of real objects and establish a number of important directions for future work.

7.2 Future Work

We have presented some promising results for the problem of 2-D and 3-D perceptual grouping and shape abstraction, opening several avenues for future research in this area. Although our methods push the state-of-the-art in 3-D shape abstraction from images of real objects, they suffer from a number of drawbacks that should be addressed. We
conclude this thesis with a discussion of possible improvements and extensions.

By its own nature, a high level of uncertainty is already involved in the abstraction problem that we are attempting to solve. Naturally occlusion, object texture and structural detail, as well as scene clutter further increase the problem’s ambiguity. Moreover, only a minor reduction in uncertainty can be expected from involving part models in the process, when such models are intermediate-level models, rather than strong object-level models. Such uncertainty makes it difficult to generate a set of abstract surface hypotheses of manageable size, which at the same time also includes most true positives. Even if such set of hypotheses is successfully generated, uncertainty still makes the problem of correctly selecting hypotheses that correspond to ground-truth parts very difficult.

In particular, our approaches for abstract part hypothesis generation are, to different extents, limited in the presence of occlusion. Although the edge-based method presented in Chapter 3 may be able to deal with some occlusion, in the case of the region-based approach introduced in Chapter 4, part occlusion will more likely yield under- or oversegmentation errors, which in turn will render the approach ineffective to recover such parts. Even if occluded parts cannot be effectively recovered, the assumption is that some parts survive occlusion and can be used as effective indices for object categorization. In spite of this clear disadvantage of the region-based approach, its computational efficiency makes it still preferable over the very slow edge-based method. A better solution to the abstract hypothesis generation problem may lie somewhere in between these two approaches. Specifically, we envision a method that relies on image edges to hypothesize abstract parts, like in our edge-based approach. However, instead of exhaustively trying image windows at all locations, scales, and orientations, it would use the structure of a hierarchy of image region segmentations and the underlying geometry of their region subsets to efficiently guide the localization of such windows. Another potential solution worth exploring is a modification of our region-based hypothesis generation method to generate part hypotheses directly from consistent paths (i.e., non-closed contours) short
of becoming consistent cycles.

In principle, when dealing with scene clutter, surface markings, object texture and structural detail, one could simply run the same abstract part detectors on the input images and just expect a very large set of part hypotheses. But this shifts the demand for higher accuracy from the hypothesis generation to the hypothesis selection process. In our experiments, we demonstrated that part relations form a powerful mechanisms to perform effective hypothesis selection. But selection becomes a difficult problem when some of the related parts are undetected (e.g., the top of an abstract cylinder is detected but not its side), or in the presence of single abstract surface objects (e.g., objects whose abstraction is an ellipsoid or a bent ellipsoid, e.g., blimp, apple, banana). In these cases, we are unable to decide if the hypothesis is a true positive based on contextual information alone. Selection mechanisms relying more strongly on other image cues need to be investigated to deal with such cases whenever other contextual information is unavailable.

Finally, 3-D part attachment regularity, i.e., attachment relations between recovered 3-D part hypotheses was not explored in this thesis. This process is not only a key component in the generation of object indices [11], but it is also another important mechanism that can be exploited for part hypothesis selection.
Appendix A

Improvements to the Consistent Cycle Detection Algorithm

In this appendix, we describe some improvements done to the consistent cycle detection approach of Chapter 4. Such improvements include:

- computing better image region segmentations as input to the consistent cycle detection process,

- replacing the Nearest Neighbor classifier, used to decide path and cycle consistency, by a Randomized Decision Forest (RDF) classifier,

- changing the path initialization step, adding weight to the edges in the region boundary graph, and adding an edge weight-based condition for path extension in the consistent cycle search,

- adding a non-maximum suppression step to the output of consistent cycle detection,

- using 2-D projections of actual 3-D shapes to generate the Point Distribution Model used during cycle 2-D shape abstraction, and
simultaneously maintaining competing 2-D abstraction hypotheses for each consistent cycle.

A.1 Image Segmentation

The consistent cycle detection algorithm requires as its input an image region oversegmentation that is fine enough such that its corresponding region boundary graph includes the boundaries of relevant parts in the image, and at the same time not excessively fine so as to make the algorithm hallucinate a large number of consistent cycles along graph edges that are not supported by image contours. In general, no low-level image region segmentation method by itself can be expected to achieve the above-mentioned goals.

In our previous experiments, we used Felzenszwalb and Huttenlocher’s Local Variation (LV) \cite{felzenszwalb2004efficient} and Nock and Nielsen’s Statistical Region Merging (SRM) \cite{nock2007statistical} region segmentation methods. The first and second rows of Figure A.1 show the region segmentations obtained by the LV and SRM methods, respectively, when directly applied to the input images that appear at the top row of Figure A.2. We observe that LV tends to produce region “bleeding” yielding many non-compact regions with one or more thin “arms” extending away. Also, LV often produces oversegmentation along image edges in the form of very long and thin adjacent parallel regions following the edge’s contour. Examples of these segmentation failures are visible in the three example scenes. On the other hand, SRM tends to produce undersegmentation along low-contrast edges, e.g., the mug’s top edge, and the rice cooker’s handles, lid’s knob, and one of its legs, as well as at blurry edges, like at the eggs’ boundaries.

To reduce the oversegmentation and undersegmentation failures produced by these graph-based algorithms, instead of applying them directly to the graph yielded by the image pixel lattice, we apply them to the graph yielded by an oversegmentation of the image. Such oversegmentation is fine enough that no significant undersegmentation of
Figure A.1: Image segmentation. Rows from top to bottom correspond to LV on original image, SRM on original image, LV on MS oversegmentation, and SRM on MS oversegmentation.
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

relevant image edges occurs, and that bleeding and spurious regions are infrequent. By restricting in this way the graph over which the segmentation algorithms can operate, we are allowing them to only produce segmentations in which the resulting regions are formed by the merging of one or more regions from the initial oversegmentation. The aim of this strategy is that the segmentation algorithms will merge oversegmented regions without incurring significant undersegmentation.

We qualitatively compared the performance of various methods, namely, Turbopixels [94], SLIC superpixels [1], Veksler et al. ’s superpixels [175], and Mean-Shift (MS) [28] to realize the initial oversegmentation of the image. (Parameters of the algorithms are fixed in our implementation; the superpixel algorithms were set to yield about 900 pixels per image, and the mean-shift algorithm was parameterized to favor oversegmentations.) Figure A.2 shows qualitative results of these methods on three different scenes. The superpixel approaches generally have problems with boundary adherence at image edges that are noisy, blurry, or have low contrast. Examples of these failures are the mug’s top rim, the interior’s rim of the milk bottle spout, and the rice cooker’s handles. Turbopixels and the Veksler method fail on the blurry left side of the milk-bottle, as well. In general, Veksler’s approach seems to struggle with boundary adherence at most image edges.

Mean-shift, although not yielding compact and similarly sized regions as its superpixel counterparts, appears to do a better job at segmenting relevant image edges in most cases. Our consistent cycle detection algorithm does not impose a restriction on region size or compactness in the input region oversegmentation but requires that the segmentation is fine enough to include contour cycles for all relevant part’s surface contours. Since this is generally the case for the oversegmentations yielded by MS, we used this method in our implementation to compute the initial oversegmentation of the input image.

Qualitative results of the LV and SRM segmentation approaches, when applied to the region graph yielded by the initial MS oversegmentation, are shown in the third and fourth rows of Figure A.1, respectively. Although an improvement is visible in these
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

Figure A.2: Image Oversegmentation. Rows from top to bottom correspond to original image, Turbopixels, SLIC superpixels, Veksler superpixels, and mean-shift segmentations.

results when compared to those achieved by the algorithms when applied directly to the input image (shown at top and second rows in the figure), each of these results has substantial undersegmentation failures. For example, the bottle’s spout’s top merges with its side, and the mug’s interior top rim merges with its interior side, in the case of LV
on MS. And the right side of the bottle’s top merges into the egg’s box top, and the rice
cooker’s right leg and left handle are undersegmented, in the case of SRM on MS. Other
undesirable properties of LV and SRM methods still remain, e.g., if $\sigma_1 < \sigma_2 < \sigma_3$ are
three close values of the scale parameter, there is no guarantee that their corresponding
image region segmentations will follow the same order of granularity. Also, there is no
fixed segmentation scale parameter that can guarantee an input region segmentation with
adequate granularity for our consistent cycle detection approach, over all input images.

In general, image region segmentation methods have a scale parameter which deter-
mines the granularity of the produced segmentations. However, the granularity of the
region segmentations of two different images obtained by applying the same method with
the scale parameter value are not necessarily similar. In order to obtain a multi-scale
region segmentation approach with a more consistent relationship between its scale pa-
rameter value and the granularity of its segmentations across different images, we propose
a simple meta-algorithm that takes advantage of one or more off-the-shelf graph-based
region segmentation methods available to the user. The proposed multi-scale region seg-
mentation procedure is based on a voting scheme, aggregating the region segmentation
decisions of various approaches at all scales. The approach starts with an initial overseg-
mentation of the input image obtained by any method (e.g., a superpixel segmentation
approach), and it computes its region boundary graph $G$. (Notice that a segmentation-
based method applied to $G$ basically returns a subgraph of $G$ obtained by deleting zero
or more of its edges.)

Let $S$ be a graph with same topology as $G$, but in which all its edges initially have
zero weight. Each chosen off-the-shelf graph-based segmentation method is applied to
$G$ for a set of scale parameters sweeping its range of scale values. For each obtained
region segmentation graph $R$, we increment the weight of each edge in $S$ that is in
$R$. At the end of this process, the weight of an edge in $S$ accounts for the number of
times that the edge was not deleted by a segmentation, for all segmentation methods
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

at all scales. Finally, the weights of $S$ are normalized to the interval $[0, 1]$ by dividing them by the largest edge weight. The multi-scale region segmentation $\mathcal{S}_\sigma(G)$ returned by the framework when applied to $G$ for a scale parameter $\sigma \in [0, 1]$ is computed from graph $S$, as the subgraph of $S$ that includes all edges with a weight at least $\sigma$. In our implementation, we applied this meta-algorithm using the LV and SRM methods on an initial oversegmentation obtained by the Mean-Shift algorithm.

Let $N_\sigma$ be the number of regions in the segmentation obtained with parameter $\sigma$, and let $N_G$ be the number of regions of the initial region oversegmentation $G$. The correspondence between the scale parameter $\sigma$ and the granularity of the segmentations obtained by the framework is not consistent among different images. If we use as the scale parameter the ratio of the number of regions in a segmentation of the hierarchy to the number of regions in the finest segmentation, we obtain a correspondence between scale and granularity across different images that we can exploit in our consistent cycle detection algorithm. Formally, this is equivalent to the framework returning region segmentation $\mathcal{S}_{f(\sigma)}(G)$, where function $f : [0, 1] \rightarrow [0, 1]$ is defined as $f(\sigma') = \arg \max_\sigma \frac{N_\sigma}{N_G} \leq \sigma'$.

From this multi-scale region segmentation hierarchy, we are able to obtain two region segmentations, a coarser one for which we can guarantee, with a high probability, that it contains a large known fraction of the boundary segments from relevant parts, and a finer segmentation that, with high probability, contains all boundary segments from all relevant parts in the scene. We use these two segmentations to make our cycle detection approach more efficient.

The image in Figure A.3 was generated from a set of 143 images in which detected ground-truth consistent cycles corresponding to the projection of actual object parts were identified as positive cases. This image corresponds to a matrix of values, whose value at position $(i, j)$ corresponds to the fraction of images for which at least $j\%$ of the boundary of all their ground-truth cycles are accounted for by their segmentation at scale $i$. For example, we see that for scale value 20 (i.e., region segmentations with only
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

20% of the total regions in the oversegmentations), 50% of the ground-truth boundaries are accounted for in 97.71% of the images. And that for scale value 79, 100% of the ground-truth boundaries are accounted for in 90.54% of the cases.

From these values empirically estimated over a set of images of a similar nature to the ones on which we will run our experiments, we deduce that, with high probability, a coarser segmentation of an input image at scale 20 will account for at least 50% of the boundary of every ground-truth consistent cycle in the image. And also, with rather high probability, a finer segmentation of the image at scale 79 will account for the entire boundary of all consistent cycles. In the next section, we explain how we use these two segmentations to make our consistent cycle search algorithm more efficient.

A.2 Consistent Cycle Detection

In this section, we describe efficiency improvements to the consistency cycle search algorithm that take advantage of the properties provided by the new image region segmenta-
tion approach. The changes to the algorithm include the path initialization step, a weight property for the region boundary graph edges, and an edge-weight-based condition for path extension. We also describe the classifier with better generalization properties and corresponding features that we used for path and cycle consistency detection to replace our previous Nearest Neighbor Data Description approach. (See Section 4.3.)

A.2.1 Search Algorithm

We modified the consistent cycle search algorithm in order to improve its efficiency, taking advantage of the properties of the new image region segmentation approach. Let \( \{\ell_1, \ldots, \ell_R\} \) be the length in pixels of the projected boundaries \( B = \{b_1, \ldots, b_R\} \) from all \( R \) visible surfaces of the 3-D parts in a scene. Recall that the region segmentation approach described in Section A.1 outputs a pair of region segmentations at two different scales, with the Region Boundary Graph (RBG) \( G^c = (V^c, E^c) \) yielded by the coarser segmentation being a subgraph of the RBG \( G^f = (V^f, E^f) \) yielded by the finer one. And these segmentations are such that, with high probability:

1. the edges in \( E^f \) include entirely each of the boundaries in \( B \), and

2. the edges in \( E^c \) account for at least length \( \rho \ell_i \) of each boundary \( b_i \), with \( 0.5 < \rho \leq 1 \).

The RBG of the finer segmentation is the one used for the search. The algorithm was modified to make sure that the search spends most of its time extending paths with edges from the RBG of the coarser (and therefore with less available edge choices) segmentation, and minimizes the number of extensions by edges that are exclusively from the finer segmentation (with many more available edge choices). We’ll show that by adding a weight property to the edges and setting these weights in a certain way, we can guarantee that each relevant cycle can be generated, starting from some edge, by subsequent edge extensions such that the fraction of the boundary length of each path
extension that is accounted by edges from the coarser segmentation is at least $\rho \ell$, where $\ell$ is the path boundary length.

Let $\ell(e)$ be the length, in pixels, of the boundary segment corresponding to an edge $e$. Let $p$ be a path, and let $\ell^c$ and $\ell^{c'}$ be the length, in pixels, of the boundary segments corresponding to edges in $p$ that belong to $E^c$ and $E^{f} \setminus E^c$, respectively. It is easy to prove that if we set the weights of each edge $e \in E^c$ to $\ell(e)$ and the weight of each edge $e' \in (E^{f} \setminus E^c)$ to $\frac{\rho}{\ell(e')}\ell(e')$, then: the fraction of the path formed by edges in $E^c$ is at least $\rho$ (i.e., $\frac{\ell^c}{\ell^c + \ell^{c'} \setminus f} \geq \rho$) if and only if the weight of the path is non-negative.

Proposition A.2.1 states that for each cycle with non-negative weight and for a given direction of traversal $d$, there exists a generator edge $e$ in the cycle such that the cycle can be generated from $e$ by successive path extensions in direction $d$, where each extended path has a non-negative weight. Since we are interested in finding cycles whose boundary fraction along edges in $E^c$ is at least $\rho$, these results guarantee that all relevant cycles can be found even if we discard paths that have a negative weight (i.e., paths in which the boundary fraction of edges in the coarse segmentation is below $\rho$).

**Proposition A.2.1.** Let $(w_0, ..., w_{n-1})$ be the weights of a sequence of graph edges forming a cycle in that order for some direction of traversal of the cycle, and let $S = \sum_{i=0}^{n-1} w_i$. If $S \geq 0$, then $(\exists j) (\forall k) \sum_{i=j}^{j+k} w_i \mod n >= 0$.

**Proof.** By contradiction, suppose that $S \geq 0$ and $(\forall j) (\exists k) \sum_{i=j}^{j+k} w_i \mod n < 0$. For each $j$, let $k_j = \min \{ k \in \mathbb{N}_0 : \sum_{i=j}^{j+k} w_i \mod n < 0 \}$. Consider the sequence recursively defined by:

\[
\begin{align*}
a_0 &= 0 \\
a_{i+1} &= a_i + k_{a_i} + 1
\end{align*}
\]

From the definition of $a_i$ and $k_j$, we know that

\[
\sum_{j=a_i}^{a_{i+1}-1} w_i \mod n < 0.
\] (A.1)
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

Now, because $n$ is finite, $\exists i_1, i_2 : i_1 < i_2$ and $a_{i_1} \equiv a_{i_2} \pmod{n}$. Let $T = \sum_{i=i_1}^{a_{i_2} - 1} w_i \mod n$.

Since $i_1 < i_2$ and $a_{i_1} \equiv a_{i_2} \pmod{n}$, it is easy to see that $\exists t \in \mathbb{N}_{>0}$ s.t. $T = t.S$. And because $t > 0$ and $S \geq 0$, it follows that $T \geq 0$.

Now, $T$ can be also rewritten as $T = \sum_{i=i_1}^{a_{i_2} - 1} \sum_{j=a_{i_1}}^{a_{i_2} - 1} w_j \mod n$. Then, using Inequality A.1, we infer that $T < 0$, which is a contradiction. \hfill \Box

The previous results state that if we start the cycle generation from a set of initial paths formed by all edges with a positive weight in both possible directions of edge traversal, and we keep extending these paths at their end while their weights are non-negative, we are guaranteed to generate all relevant cycles. Since the direction of traversal of an edge matters, the consistent cycle search is actually performed on a directed RBG $\tilde{G}^f$ obtained from $G^f$ by replacing each undirected edge $e = (v_1, v_2) \in E^f$ between a pair of nodes by two directed edges $\hat{e}_1 = (v_1, v_2) \in \hat{E}^f$ and $\hat{e}_2 = (v_2, v_1) \in \hat{E}^f$ in opposing directions between the same two nodes. Edge ranks for the path initialization edges were set according to their weight, with increasing ranks assigned in descending order of weight. The remaining edges were assigned a rank of $\infty$.

In our previous formulation of the search algorithm, edge ranks were used to condition the extension of a path in order to set a unique valid order of exploration of a cycle to avoid repeated generations of the same cycle. Paths were allowed to be extended only by edges with a higher rank than that of the initial edge in the path. In our new formulation, for a cycle to be fully traversed by the algorithm, its exploration must start at a generator edge. Since the rank of such an edge may not be the lowest among all cycle edges, we need to modify the way in which rank values are used to constrain path extensions in order to avoid repeated generations of the same cycle.

In order to ensure that all relevant cycles are fully explored, and at the same time the repeated exploration and generation of the same cycle is restricted, the algorithm’s structure was modified and a dictionary was introduced to keep the set of paths from each initial edge that were abandoned because of the path achieving a negative weight.
Figure A.4 shows the new version of the algorithm. The search is realized sequentially, starting at each initial path, one at a time, in the order of their ranks. While the queue is not empty, a path $p$ is removed from the queue, and if it is not consistent, it is discarded. If $p$ is a cycle, it is added to the output list of consistent cycles; otherwise, the path’s possible extensions by each edge $e$ are considered. If the rank of $e$ is greater than that of $p$, then the extended path $p \parallel e$ is added to the queue if its weight is non-negative, or otherwise it is added to the dictionary of abandoned paths. If the rank of $e$ is less than that of $p$, $p$ is extended by each abandoned path $p'$ starting at $e$, and each extended path $p \parallel p'$ is added to the queue if its weight is non-negative, or otherwise it is added to the dictionary of abandoned paths.

The reasoning behind the need for maintaining a list of abandoned paths originating at each initial edge is the following. Consider a path $p$ originating at an edge $s$ that is currently being extended by the algorithm, and assume that an edge $e$ of lower rank than $s$ is reached. Since $e$ has a lower rank, all the consistent cycles with a non-negative weight that have $e$ as a generator have already been generated in a previous iteration. There is no need to extend $p$ along all paths that originate at $e$ that lead to those cycles. However, it is possible that there is a consistent cycle with total non-negative weight that includes $p$ and a path $p'$ originating at $e$ which was abandoned because of its negative weight. For this reason, all extensions of $p$ by such paths $p'$ need to be explored.

### A.2.2 Consistency Classification

We replaced the Neighbor Data Description approach (see Section 4.3) used for the path and cycle consistency classification step in the consistency cycle detection algorithm by a classifier with better generalization properties, namely, a Randomized Decision Forest classifier, on a corresponding set of new features. Randomized Decision Forest (RDF) \cite{18} has proven to be a fast and effective multi-class classifier for many computer vision tasks, including object recognition and body part segmentation \cite{157, 32}. An RDF consists of
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

1: **Input:** edge sets $\hat{E}^c$ and $\hat{E}^f$
2: **Output:** a set $R$ containing the detected consistent cycles
3: $R \leftarrow \emptyset \{\text{Consistent Cycles}\}$
4: $Q \leftarrow \emptyset \{\text{Open paths}\}$
5: $U \leftarrow \emptyset \{\text{Abandoned paths}\}$
6: **for all** $e \in \hat{E}^c$, in ascending rank order **do**
7: \[ Q \leftarrow Q \cup \{e\} \]
8: **while** $Q \neq \emptyset$ **do**
9: \[ p \leftarrow \text{pop}(Q) \]
10: **if** $p$ is consistent **then**
11: \[ R \leftarrow R \cup \{p\} \]
12: **else**
13: **for all** edge $e$ incident to the end of $p$ **do**
14: \[ \text{if } \text{rank}(p) < \text{rank}(e) \text{ then} \]
15: \[ \text{if } \text{weight}(p \parallel e) \geq 0 \text{ then} \]
16: \[ Q \leftarrow Q \cup \{p \parallel e\} \]
17: \[ \text{else} \]
18: \[ U \leftarrow U \cup \{p \parallel e\} \]
19: **end if**
20: **else**
21: **for all** $p' \in U$ with initial edge $e$ **do**
22: \[ \text{if } \text{weight}(p \parallel p') \geq 0 \text{ then} \]
23: \[ Q \leftarrow Q \cup \{p \parallel p'\} \]
24: \[ \text{else} \]
25: \[ U \leftarrow U \cup \{p \parallel p'\} \]
26: **end if**
27: **end for**
28: **end if**
29: **end for**
30: **end if**
31: **end if**
32: **end if**
33: **end while**
34: **end for**

Figure A.4: Consistent Cycle Search Algorithm

a collection of Randomized Decision Trees (RDT). An RDT is a binary decision tree in which each internal (i.e., non-leaf) node has an associated split binary function, which when applied to an input, returns a decision if the input is to be routed to the node's
right or left child. Each leaf node in an RDT is associated with a discrete probability
distribution over the set C of classes of the classifier.

The classification process in an RDT for a given input $X$ starts by first processing
$X$ in the tree’s root node by applying the root’s associated split function and recursively
continuing the process of the input on the child node corresponding to the split function’s
result. When the process reaches a leaf node, the probability distribution associated with
the leaf node is returned as the output. The split function at each non-leaf node returns a
binary value based on a particular feature computed on the input. The class probability
distribution output of the RDF classifier, for a given input $X$, is computed by averaging
the probability distributions output by each of its RDTs for input $X$.

### A.2.3 RDF Training

An RDF classifier is trained by independently training each of its RDTs. An RDT is
trained one node at a time, starting from its root node. Each node is trained using
a certain input training set. Initially, the entire training set is the input set used for
training the root node. Given an input training set $T$ for a node $n$, the node is trained
by randomly sampling a sufficiently large number of times the parameter space of split
functions. For every sampled split function $f$, each input $x \in T$ is assigned to a partition
of $T$ into two sets, $T^f_0$ and $T^f_1$, based on the boolean value of $f(x)$.

Let $f$ be the sampled split function that achieves the maximum Information Gain
(IG) over the split of all elements in the node’s input set $T$. (Information gain,
$IG(T, f) = H(T) - \sum_{i=0,1} \frac{|T^f_i|}{|T|} H(T^f_i)$, is the expected reduction in entropy caused by
partitioning the input training set $T$ according to $f$, and entropy,
$H(T) = -\sum_{x \in \{0,1\}} p_T(x) \log(p_T(x))$, characterizes the purity of the input training set $T$.)
If IG is too small or if the node is at a maximum preselected height, then the node is
set to be a leaf node and the class probability distribution associated with that leaf node
is set to the proportion of each class present in the input set $T$. On the other hand, if
IG is sufficiently high, the split function $f$ is associated to the node $n$, and the node is assigned two children nodes, each being recursively trained using the input sets $T_0^f$ and $T_1^f$ yielded by $f$.

### A.2.4 Path and Cycle Features

In this section, we describe the binary features that were used as split functions by the RDF for path and cycle consistency classification.

As before, the features used to decide on the consistency of a path or cycle contour are computed on an initial abstraction of the contour achieved by its polyline approximation (by means of the Ramer-Douglas-Peuker’s method) at the contour’s proper scale. This ensures a one-to-one correspondence between features computed on abstractly equivalent contours regardless of their differences in detail. See Figure A.5 for example. In our new formulation, a contour’s proper scale is computed as the diameter of the maximum inscribed disk inside the contour (instead of the “diameter” of the contour computed as twice the root mean square distance of the cycle boundary to its centroid, used in Chapter 4). Using this scale definition, the largest number of segments achieved by a polyline approximation of a consistent path or cycle in our implementation was 19 segments.

In order to compute the features used for classification, the polyline is subsampled...
by evenly dividing each polyline segment into a fixed number $Q$ of subsegments. (In our implementation, we used $Q = 10$.) Thus, the polyline becomes a sequence of $Q \times S + 1$ points, where $S$ is the number of segments in the polyline. Three classes of binary features were computed on this polyline representation, attempting to capture the 2-D shape of the underlying path or cycle in a translation-, rotation-, and scale-invariant manner.

Let $i_0, i_1,$ and $i_2$ be the indices of three of the points sampled along a polyline, and let $\tau \in \mathbb{R}$. The first class of features, with parameters $\Theta = \langle i_0, i_1, \tau \rangle$, corresponds to the ratio of the distance $b$ between the polygonal points with indices $i_0$ and $i_1$ to the arc-length $l$ (along the polyline) between such points, in comparison to the threshold value $\tau$, as shown in Figure A.6 (a):

$$f^{\text{chord}}_{\Theta} = \frac{b}{l} \leq \tau.$$

The second class of features, parameterized by $\Theta = \langle i_0, i_1, \tau \rangle$, is proportional to the ratio of the unsigned distance $h$ from the line segment between points with indices $i_0$ and $i_1$ and its furthest point along the polygonal between such points to the arc-length $l$ between them along the polyline, in comparison to the threshold value $\tau$. See Figure A.6 (b):

$$f^{\text{maxDist}}_{\Theta} = \frac{2h}{l} \leq \tau.$$

The third class of features, with parameters $\Theta = \langle i_0, i_1, i_2, \tau \rangle$, corresponds to the signed angle magnitude $\Delta$ between the two line segments defined by the polyline points with indices $(i_0, i_1)$ and $(i_1, i_2)$, in comparison to the threshold value $\tau$, as shown in Figure A.6 (c):

$$f^{\text{angle}}_{\Theta} = \Delta \leq \tau.$$

The feature space sizes are $T \times (Q \times S + 1)^2$ for the first and second classes, and $T \times (Q \times S + 1)^3$ for the third class, for polylines with $S$ segments and constraining threshold $\tau$ to $T$ discrete values. During the training of each tree node, the spaces of all three feature classes were sampled. The number of samples from each class of features was set to the square root of the corresponding feature space size, as suggested by Brieman [18].

We trained the classifiers using synthetically generated contours in the same way that we did for our previous formulation. Initially, a large number of contours were
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

\[ f^\text{chord} = \frac{b}{l} \leq \tau, \theta = (i_0, i_1, \tau) \]

\[ f^\text{maxDist} = \frac{2h}{l} \leq \tau, \theta = (i_0, i_1, \tau) \]

\[ f^\text{angle} = \alpha \leq \tau, \theta = (i_0, i_1, i_2, \tau) \]

Figure A.6: Example of the three types of polyline features used for cycle and path consistency classification. For this example, \( Q = 3 \).

synthetically generated as noisy versions of shapes from the 2-D model vocabulary by randomly sampling their dimensions, bending, tapering, and shearing parameters, where various levels of noise were used. This set of contours was split into two sets. One set of contours were used as positive training cases. The other set was used to generate negative training cases, each consisting of the concatenation of three subcontours extracted from three contours randomly selected from the set, after they were randomly scaled and rotated.

As in our previous formulation, different sets of classifiers were trained and used for open and closed contours (i.e., path and cycles). And each of those two types of contours had a different classifier for each possible polyline segment count from 3 to 19. Each RDF classifier contained 12 trees, and each tree had a maximum height of 20. Features for positive training cases were computed on polylines of subcontours from synthetically generated positive training contours, computed at the proper model’s scale. Features for negative training cases were computed on polylines of subcontours from synthetically generated negative training contours, computed at various random scales. The path and cycle classifiers were trained on 33.6 million and 2.3 million cases, respectively. Test data was also independently generated (11.2 million path cases and 770K cycle cases).
and used to select, based on each classifier’s receiver operating characteristic (ROC), the consistency probability thresholds achieving a maximum false negative rate of 0.01.

The performance of the trained classifiers on the test data is shown in Figures A.7 and A.8 for path and cycles, respectively, for the lengths of their polyline approximations in row-major order, starting at the left of the top row at length 3. We see that the performance of path consistency classification for cycles with a polyline approximation of length 3 is somewhat poor compared with longer polylines. This is expected, since such a short polyline only carries local information insufficient to make a judgment on the consistency of the entire cycle it belongs to. Also, due to the diminishing number of training examples of consistent paths for polylines with 16 or more segments, a degradation in performance is observed for those cases.

In the case of the consistent cycles classifiers, the performance of cycles with polyline approximations of length 3 is also poor. This is because no shapes with a triangular abstraction were included in the set of model shapes used to generate the training examples. We also observe that the performance is almost perfect for cycles with shorter polyline approximations of length at least 4, but it degrades as the polyline approximations become longer, which is to be expected as the complexity of the underlying cycles increases. Performance for cycles approximated by polylines of lengths 16 onwards decreases for the same reason mentioned above of a diminishing number of examples for such cases.

A.3 Non-Maximum Suppression

A step was introduced to do non-maximum suppression of competing consistent cycle hypotheses. When two or more consistent cycles have a high spatial overlap (i.e., they attempt to explain very much the same image boundaries), only the cycle with maximum likelihood of being an actual 2-D part boundary should be preserved, and the others discarded. We applied logistic regression to various image features related to the cycle’s
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

contour, in order to estimate such likelihood. The employed features were designed to be
good discriminators of the image properties that set an actual 2-D part boundary apart
from other overlapping closed image contours. Such image properties include contour
saliency, good internal region continuation, and contour complexity.
Figure A.8: Performance of cycle consistency classifiers for each polyline length.

A.3.1 Approach

The goal of the non-maximum suppression step is to select a winning cycle hypothesis when two or more are competing to explain the same data. This is not a competition between each cycle hypothesis and all others, but between all cycle hypotheses within a
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

Given subset of spatially overlapping cycles. We formulate this as a classification problem in which we want to determine, for each pair of overlapping cycles \((c_1, c_2)\), the probability \(p_{\geq}(c_1, c_2)\) of \(c_1\) being more likely an actual 2-D part boundary than \(c_2\). We model this probability of cycle ordering using logistic regression applied to the feature vector difference \(F = F^{c_1} - F^{c_2}\), where \(F^{c_i}\) is the feature vector of cycle hypothesis \(c_i, i \in \{1, 2\}\), over a specific set of features.

A.3.2 Model Features and Training

We selected simple features that can be efficiently computed from image data and that are informative at deciding which one of two competing 2-D cycle hypotheses is more likely to correspond to a true 2-D part boundary. Expected properties of an actual 2-D boundary include contour saliency, good internal continuation of the region enclosed by the boundary, and a low complexity of the boundary. Here are the details on the set of features that we employed in our approach:

- Contour saliency: As indicators of saliency, we used 1) the mean gradient magnitude along the cycle hypothesis, and 2) the minimum gradient magnitude over all fragments that results from splitting the cycle’s contour into fragments of equal length, with this feature computed for a set of fragment lengths. The lengths in this set are obtained from a fixed set of fractions of the image size.

- Good internal region continuation: The mean gradient magnitude along region boundary fragments that are inside the region enclosed by the cycle hypothesis and that are incident to the cycle’s contour was used as a measure of internal region continuation.

- Shape complexity: We use various features to assess boundary shape complexity based on the following shape properties:
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

Complexity of Shape Approximation: We estimate this from: 1) the number of polyline segments in the cycle’s polyline approximations computed at various multiples of the cycle’s proper scale. (Simpler boundary shapes are expected to have a polyline approximation with less segments than more complex ones.); from 2) the average polyline segment length of the approximating polyline at the cycle’s proper scale, normalized by the value of the cycle’s proper scale; and from 3) the ratio between the perimeter of the cycle and the arc-length of the approximating polyline at the cycle’s proper scale.

Sinuosity: This is estimated from: 1) the average number of changes in “curvature sign” along the cycle, approximated by the average number of changes in angle sign between consecutive polyline segments in the cycle’s polyline approximation computed at a scale significantly smaller than the cycle’s proper scale; from 2) the mean absolute angle between consecutive polygonal segments in the same polyline approximation; and from 3) the number of high curvature\(^1\) points along the cycle’s contour.

Divergence from Simple Shape: This is approximated by the mean distance from the cycle’s contour points to their corresponding segment in the cycle’s polyline approximation computed at the cycle’s proper scale. This distance is normalized by the cycle’s proper scale.

The model parameters were learned from features computed on a set of 130 manually-labeled training images, where cycle detection was initially applied to each training image and groups of detected consistent cycles competing to explain the same image data were identified. In each group, the cycles that best explained the image contours were identified as positive cases for the group. Formally, let \( C \) be the set of consistent cycles in the

\(^1\)In our implementation, we considered the contour to have high curvature at a given contour pixel \( p \) when the cosine of the angle between the contour normal vectors (computed on a Gaussian smoothed version of the cycle’s contour) at the two contour pixels adjacent to \( p \) have a value less than 0.85.
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

Let $W$ be the learned model’s weight vector such that $p_{\geq}(c_1, c_2) = \logit^{-1}(W(F_{c_1} - F_{c_2}))$. Non-maximum suppression over a set $H$ of competing cycle hypothesis is performed at runtime by selecting cycle \( \hat{c} = \arg \max_{c \in H} WF^c \), which is guaranteed to maximize $p_{\geq}(c, c')$ over all $c' \in H$. (Because $\logit^{-1}$ is a monotonically increasing function and $\forall c : F^c \geq 0$, we know that $p_{\geq}(c, c')$ achieves its maximum value when $W(F^c - F^{c'})$ is maximum which, in turn, is achieved when $WF^c$ is maximum.)

Figure A.9 shows the performance of the trained classifier using 10-fold cross-validation on the annotated data. (Cycles from a same group were all assigned to the same fold.) As shown by the ROC, the obtained classifier does a good job at picking the cycle that best explains image data within each competing group.

Figure A.9: Non-maximum suppression performance.
A.4 2-D Abstraction

Improvements to the 2-D abstraction step include using a Point Distribution Model generated from projections of actual 3-D volumes, softening the abstraction decisions by simultaneously keeping several abstraction hypotheses per cycle, and adding a hypothesis false positive filtering step that takes advantage of the newly available 2-D abstraction info.

A.4.1 Training Models

We improved the 2-D abstraction of consistent cycles by applying the ASM fitting approach on a Point Distribution Model (PDM) generated from a set of models that more accurately represent the shape of the projected 3-D part surfaces. Instead of using deformations of simple 2-D shapes (e.g., stretching, bending, and tapering of ellipses and parallelograms) to approximate the projection of faces of abstract 3-D shapes, we used the shapes of actual 2-D projections of surface boundaries of the simple 3-D volumes forming our abstract 3-D vocabulary, i.e., cuboids, cylinders, tapered cylinders, ellipsoids, and bent ellipsoids. (See Figure 6.11.) These 2-D model shapes were generated from orthographic projections of models generated by finely sweeping the parameter space of each volume in the vocabulary, including size along each axis, 3-D orientation, bending and tapering parameters, and by discretely sweeping the viewing sphere.

A.4.2 Abstract Model Classes

The 2-D abstractions of detected cycles constitute a set of 2-D part hypotheses for projections of abstract 3-D part surfaces from the scene. The orthographic projection of each visible face in a simple volume falls within a very small set of shape categories. We refer to these categories as 2-D Shape Families. Table A.1 shows the shape family that each projected face of a volume from our vocabulary of simple 3-D models can belong to.
APPENDIX A.  IMPROVEMENTS TO THE CONSISTENT CYCLE DETECTION ALGORITHM

<table>
<thead>
<tr>
<th>Volume</th>
<th>Face</th>
<th>Shape Family</th>
</tr>
</thead>
</table>
| Cylinder     | top / bottom side | ellipse  
              |                        | bent-rectangle         |
| Tapered-Cylinder | top / bottom side | ellipse  
       |                        | bent-tapered-rectangle / annuloid |
| Cuboid       | side      | parallelogram          |
| Ellipsoid    |            | ellipse                 |
| Bent-Ellipsoid |          | bent-ellipse          |

Table A.1: Shape families of projected faces of volumes from our 3-D vocabulary.

These are: ellipse, parallelogram, bent-ellipse, bent-rectangle, bent-tapered-rectangle, and annulus. (In our implementation, we did not generate models for almost-accidental viewpoints in which the projection of the side face of a tapered-cylinder has an annular shape.) The shape family of a 2-D part abstraction thus restricts the possible 3-D face interpretation of the part hypothesis.

When in the presence of contour noise or simply within-class variation due to the hard-decision nature of the 2-D abstraction process of Section 4.3 (in which a unique abstract shape is fitted to a consistent cycle based purely on the 2-D shape of the cycle), the converged 2-D abstract shape may be of a different shape family than that of the actual abstract volume’s face that the cycle corresponds to, if any. Since the 3-D interpretation of a 2-D part abstraction is restricted by the shape family of the abstraction, 2-D abstractions resulting in an incorrect shape family should be avoided. To minimize the number of misclassifications of this type, we relaxed the 2-D abstraction process so that instead of it making hard, unrecoverable abstraction decisions, it actually maintained several simultaneous abstraction hypotheses. We did this by splitting the set of abstract 2-D models from the original PDM into six separate PDMs depending on their shape families and then applying the 2-D abstraction process to each consistent cycle.
Appendix A. Improvements to the Consistent Cycle Detection Algorithm

independently for each one of the six shape families. The fitted contour from each shape family defines a 2-D abstraction hypothesis for the cycle.

A.4.3 False Positive Filtering

Notice that it only makes sense to simultaneously maintain alternative 2-D abstraction hypotheses (of different shape families) for a cycle when there is obvious ambiguity in the cycle’s 2-D shape. However, in many cases, it is easy to check that a cycle’s shape does not belong to one or more shape families. To discard these abstraction hypotheses from a wrong shape family (as well as false positive detected cycles whose shape is actually inconsistent and therefore does not belong to any of the expected shape families), a false positive filtering step was added, in which 2-D abstraction hypotheses are discarded when the cycle’s shape is unlikely to belong to the hypothesized shape family.

To implement this filtering, a separate logistic regression classifier is trained for each one of the six shape families in order to estimate (based on the shapes of the cycle and its 2-D abstraction and from corresponding image features) the probability that the 2-D abstraction hypothesis corresponds to the projection of a scene’s 3-D part surface from the given shape family. False positive filtering is performed by simply discarding cycle-abstraction combinations with a low probability. The parameters of these models were learned on the same image set used to train the non-maximum suppression model.

Motivated by the same principles of contour saliency, good internal region continuation, and low boundary complexity, the features employed in this classification include those used for non-maximum suppression plus the following indicators of a good abstraction fitting:

- Fitting error between the cycle and its 2-D abstraction, normalized by the 2-D hypothesis proper scale.

- Ratio between the cycle’s perimeter and the perimeter of its 2-D abstraction.
For various splittings of the cycle’s contour into same-length fragments (for various fragment lengths), the minimum value, over all fragments, of the contour “coverage”\(^2\) by its 2-D abstraction model and, for various splittings of the 2-D abstraction model’s contour, the minimum coverage of the model’s contour by the cycle.

Figure A.10 shows the performance of the learned classifiers on each shape family. The best performance is for the bent-ellipse classifier. Following in order of performance by the classifiers for tapered-rectangles, ellipses, parallelograms, bent-rectangles, and last with a significantly worst performance, bent-tapered-rectangles.

\(^2\)Coverage of a contour fragment by another contour is measured as the number of points in the fragment that are the closest point to some point in the contour.
Figure A.10: Performance of false positive filtering: (a) ellipse; (b) parallelogram; (c) bent-rectangle; (d) tapered-rectangle; (e) bent-ellipse; and (f) bent-tapered-rectangle.
Appendix B

Geometrical Interpretation of Signature Vector Scaling

Let $F = f_1, \ldots, f_K$ be the faces involved in a shape signature vector, and let $A_k$ be the area of face $f_k$, and let $A = \sum_{k=1}^{K} A_k$. Let $T$ be the number of equidistantly sampled points along each face contour, and let $\delta_k$ be the distance between two adjacent sampled points on face $f_k$.

If the points of an ASV are scaled such that points from face $f_k$ are scaled by factor $\xi_k = \frac{\delta_k}{A} \sqrt{KT}$, then the Euclidean distance between two nearby vectors is a plausible upper bound of the ratio of the integral (along face’s contours) of face contour alignment error to total sum of face areas.

Proof. Let $s^a, s^b$ be two signature vectors encoding similar views, and let $\bar{x}_{ik}^h$ be the $i$-th sampled point along the $k$-th face contour of vector $s^h$ before scaling ($h = a, b$). Since the views encoded by $s^a$ and $s^b$ are close, and hence, their corresponding faces have similar shapes, then $\delta_k^a \approx \delta_k^b$ and $A^a \approx A^b$. We can then approximate these by $\delta_k = (\delta_k^a + \delta_k^b)/2$
and $A = (A^a + A^b)/2$. Let $\tilde{s}^a$ and $\tilde{s}^b$ be the scaled versions of the signature vectors. Then

$$||\tilde{s}^a - \tilde{s}^b|| = \sqrt{\sum_{k=1}^{K} \sum_{i=1}^{T} ||\xi^a_k \vec{x}^a_{ik} - \xi^b_k \vec{x}^b_{ik}||^2}$$

$$= KT \sqrt{\frac{1}{K} \sum_{k=1}^{K} \frac{1}{T} \sum_{i=1}^{T} ||\frac{\delta^a_k}{A^a} \vec{x}^a_{ik} - \frac{\delta^b_k}{A^b} \vec{x}^b_{ik}||^2}$$

$$\approx KT \sqrt{\frac{1}{K} \sum_{k=1}^{K} \frac{1}{T} \sum_{i=1}^{T} \frac{\delta^2_k}{A^2} ||\vec{x}^a_{ik} - \vec{x}^b_{ik}||^2}$$

$$> KT \frac{1}{K} \sum_{k=1}^{K} \frac{1}{T} \sum_{i=1}^{T} \frac{\delta_k}{A_k} ||\vec{x}^a_{ik} - \vec{x}^b_{ik}||$$

$$= \sum_{k=1}^{K} \left[ T\delta_k \right] \left[ \frac{1}{T\Pi_k} \sum_{i=1}^{T} ||\vec{x}^a_{ik} - \vec{x}^b_{ik}|| \right] / A,$$

where the inequality between Equations B.1 and B.2 holds because the first is a root mean squared error, and the second is a mean absolute error. Expression $\Pi_k$ approximates the average perimeter of the $k$-th face contour from both views, and expression $\Delta_k$ approximates the average distance between corresponding contour points on the $k$-th face in the two views. Hence, $\Pi_k \times \Delta_k$ approximates the integral of the alignment error along the $k$-th face contour of both views. Equation B.3 thus approximates the ratio of the integral (along face’s contours) of face alignment error to total face area. □
Bibliography


[47] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object cate-


