

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Analyzing 3D Objects in 2D Images

Permalink

<https://escholarship.org/uc/item/6vf2h7b2>

Author

Hejrati, seyed mohammadmohsen

Publication Date

2015

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Analyzing 3D Objects in 2D Images

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Mohsen Hejrati

Dissertation Committee:
Professor Deva Ramanan, Chair
Professor Charless Fowlkes
Professor Aditi Majumder

Dedicated to my parents.

TABLE OF CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	xiii
ACKNOWLEDGEMENTS	xiv
CURRICULUM VITAE	xiv
ABSTRACT	i
CHAPTER	
I. Introduction	1
1.1 3D Recognition with 2D Alignment	4
1.2 3D Recognition with 3D Synthesis	6
1.3 3D Categorization	7
II. 3D Recognition with 2D Alignment	9
2.1 Introduction	9
2.2 Related Work	13
2.3 2D Shape and Appearance	14
2.3.1 Inference	17
2.3.2 Learning	17
2.4 3D Shape and Viewpoint	23
2.5 Experiments	25
2.5.1 Datasets	25
2.5.2 Implementation	25
2.5.3 Evaluation	26
2.5.4 Viewpoint Classification	26
2.5.5 Baselines	27
2.5.6 Diagnostics	28
2.5.7 3D Shape	29

2.6	Conclusion	33
III.	3D Recognition with 3D Synthesis	34
3.1	Introduction	34
3.2	Related Work	36
3.3	Synthesis model	37
3.4	Template model	41
3.5	Inference	43
3.6	Learning	44
3.7	Results	47
3.7.1	Evaluation	47
3.7.2	Baselines	47
3.7.3	Implementation	48
3.7.4	Synthesis strategies	48
3.7.5	Interactive synthesis	49
3.7.6	Anytime recognition/reconstruction	49
3.7.7	Box benchmark results	50
3.7.8	Car benchmark results	50
3.7.9	Diagnostic analysis	51
3.8	Conclusions	56
IV.	3D Categorization	57
4.1	Introduction	57
4.2	Image-based rendering	60
4.3	Approaches	62
4.3.1	Pose-Agnostic	64
4.3.2	Pose-Normalized	64
4.3.3	Pose-Retargeted	65
4.3.4	Pose-Synthesis	66
4.3.5	Theoretical Analysis	68
4.4	Experimental Results	69
4.4.1	Categorization	71
4.4.2	Landmark localization	79
4.5	Conclusion	82
	BIBLIOGRAPHY	83

LIST OF FIGURES

Figure

1.1	A large body of work is focused on classifying an image into one of many labels (<i>a</i>), object detection methods aim to provide location of objects in the image (<i>b</i>). Many methods build spatial understanding of images by reasoning about surfaces in the images (<i>c</i>), while other are based on aligning geometric shapes in order to reconstruct indoor scenes or buildings (<i>d</i>). Understanding 3D structures from 2D images and creating rich 3D representation and reconstruction is a defining challenge in machine vision (<i>e</i>).	2
1.2	A large body of work focus on 3D reconstruction from multiple images. In this thesis we concentrate on single images. Some methods reconstruct 3D structure of single images by analyzing small image patches or pixels. Some approaches are based on fitting geometric shapes to the whole scene. We focus on 3D recognition, reconstruction and categorization of objects.	3
1.3	We focus on 3D object recognition, reconstruction and categorization in single 2D images. Chapter 2 and 3 present two novel approaches for 3D object recognition and reconstruction and Chapter 4 investigates various representation methods for 3D object categorization.	4
1.4	Chapter 2 introduces a two-stage model for detecting and analyzing the 3D shape of objects in unconstrained images. In the first stage, our model reason about 2D appearance and shape using variants of deformable part models (DPMs). Our 2D model localizes even fully-occluded landmarks, shown as hollow circles and dashed lines in (top-middle). We feed this output to our second stage, which directly reasons about 3D shape and camera viewpoint. We show the reconstructed 3D model on (top-right). The bottom 3 viewpoints.	5

1.5	In Chapter 3 we propose an analysis by synthesis approach for 3D object recognition. We describe a method for synthesizing a large set of discriminative templates, each associated with a candidate 3D reconstruction of an object, then the interpretation that best agrees with the test image is selected.	6
1.6	In Chapter 4 we examine how to use the geometric-reasoning engines proposed in Chapter 2 and 3 for <i>categorical recognition</i> . We evaluate 3D shape categorization of cuboidal objects (left). Such objects share similar shape, so conventional folk wisdom might advocate the use of shape-invariant (or pose-normalized) representations for recognition (top) that are attractive because they (1) factor out shape (which seems uninformative when classifying objects with similar shape) and (2) can generalize to novel shapes not encountered in training data. We show that this approach is not optimal. We demonstrate that pose-synthesis (bottom), a simple approach of augmenting training data with geometrically perturbed training samples, is a surprisingly effective strategy that allows for state-of-the-art categorization and automatic 3D alignment.	7
2.1	Overview	10
2.2	A two-stage models for detecting and analyzing the 3D shape of objects in unconstrained images is proposed. In the first stage, our models reason about 2D appearance and shape using variants of deformable part models (DPMs). We use global mixtures of trees with local mixtures of gradient-based part templates (top-left). Global mixtures capture constraints on visibility and shape (headlights are only visible in certain views at certain locations), while local mixtures capture constraints on appearance (headlights look different in different views). Our 2D models localize even fully-occluded landmarks, shown as hollow circles and dashed lines in (top-middle). We feed this output to our second stage, which directly reasons about 3D shape and camera viewpoint. We show the reconstructed 3D model and associated ground-plane (assuming its parallel to the car body) on (top-right). The bottom row shows 3D reconstructions from four novel viewpoints.	12
2.3	We report histograms of viewpoint label errors for the dataset of [1]. We compare to the reported performance of [1] and [22]. Our model reduces the median error (right) by a factor of 2.	27

2.4	We compare our model with various view-based baselines in (a), and examine various components of our model through a diagnostic analysis in (b). We refer the reader to the text for a detailed analysis, but our model outperforms many state-of-the-art view-based baselines based on trees, stars, and latent parts. We also find that modeling the effects of shape due to global changes in 3D viewpoint is crucial for both detection and landmark localization.	28
2.5	Sample results of our system on real images with heavy clutter and occlusion. We show pairs of images corresponding to detections that matched to ground-truth annotations. The top image (in the pair) shows the output of our tree model, and the bottom shows our 3D shape reconstruction, following the notational conventions of Figure 2.2. Our system estimates 3D shapes of multiple cars under heavy clutter and occlusions, even in cases where more than 50% of a car is occluded. Our morphable 3D model adapts to the shape of the car, producing different reconstructions for SUVs and sedans (row 2, columns 2-3). Recall that our tree model explicitly reasons about changes in visibility due to self-occlusions versus occlusions from other objects, manifested as local mixture templates. This allow our 3D reconstructions to model occlusions due to other objects (e.g., the rear of the car in row 2, column 3). In some cases, the estimated 3D shape is misaligned due to extreme shape variation of the car instance (e.g., the folding doors on the lower-right).	29
2.6	Sample results of our system on real images with heavy clutter and occlusion. We show pairs of image corresponding to a detection that matched to a ground-truth annotation. The top image (in the pair) shows the output of our tree model, and the bottom shows our 3D shape reconstruction, following the notational conventions of Figure 2.5. Our system estimates 3D shapes of multiple cars under heavy clutter and occlusions, even in cases where more than 50% of a car is occluded.	31
2.7	Sample results of our system on real images with heavy clutter and occlusion. We show pairs of image corresponding to a detection that matched to a ground-truth annotation. The top image (in the pair) shows the output of our tree model, and the bottom shows our 3D shape reconstruction, following the notational conventions of Figure 2.5. Our system estimates 3D shapes of multiple cars under heavy clutter and occlusions, even in cases where more than 50% of a car is occluded.	32
3.1	Overview	35

3.2	We describe a method for synthesizing a large set of discriminative templates, each associated with a candidate 3D reconstruction of an object (in this case, cars). Our model makes use of a generative 3D shape model to synthesize a large collection of 2D landmarks, which in turn specify rules for composing 2D templates out of a common pool of parts.	36
3.3	We use basis shapes to model different types of cars, like sedans in the first column and SUVs in the fourth. Since the [55] assumes orthographic camera, another major aspect of learned shape basis is modeling perspective effect. For example the second and third column are modeling back and front view perspective effect.	38
3.4	We learn local part mixtures by clustering the relative 3D position of keypoint i and its connected neighbors in the underlying 3D mesh. We show keypoint cluster means μ_k^i (above), along with their associated part templates β_i^k (below). Each synthesized 3D pose (and associated template) is constructed by adding together shifted copies of local part templates, which in turn allows for efficient run-time search.	41
3.5	We search through a large collection of templates (with shared parts) by first caching part responses, and then looking up response values to score each template.	43
3.6	A visualization of our interactive, morphable interface for exploring 3D shapes and their associated templates. We display the corresponding shape coefficients α as colored bars.	49

- 3.7 Detection (**left**) and reconstruction accuracy (**right**) versus running time of our method and baselines, including DPMs [18], supervised-tree models and multi-view star models (introduced in Chapter 2). Points correspond to different (constant-time) baselines, while curves correspond to our models. Because our models can process a variable number of synthesized templates, we sweep over $K \in \{ 20, 50, 100, 500, 1000, 4000 \}$ templates to generate the curves. Note that Exemplars are limited by the number of training images. Exemplars always dominate Parametric Synthesis (for a given K), suggesting our parametric model is failing to capture important shape statistics. We examine this further in Figure 3.8. Our box (**top**) detection and reconstruction results (43% and 48%) nearly double the best previously-reported performance from Xiao et al.[62] (24% and 38%), while being 10X faster. Our car (**bottom**) results approach the state-of-the-art tree models proposed in Chapter 2, but directly report 3D shape while being 5X faster. One can also use cascade models and or context to reduce the number of evaluated synthesized templates, thus spend less time for detection. 52
- 3.8 We plot the performance of various synthesis approaches as a function of the amount of training images. **Exemp** enumerates the set of shapes encountered in the training set of images. **+Exemp Synth** uses the learned local templates β from **Exemp** and instantiates new shapes obtained from keypoint annotations not in the training set. This improves performance by up to 2%. **+Retrain** discriminatively retrains β given this synthesized set of shapes, further improving performance by up to 5%. Hence it is crucial to discriminatively-tune the synthesized set. Our synthesis models outperform state-of-the-art methods [62] with orders-of-magnitude less training data. 53
- 3.9 Recognition + reconstructions from our method. Odd rows show the test image and recognized + reconstructed object overlaid on it. Even rows illustrate the associated template that triggered the detection. Our method can recognize objects from various viewpoints, shapes and is robust to heavy occlusion. Because every synthesized template has a 3D shape, recognition is inherently reconstruction. On the top right, we show results for images with multiple cars. Our box results show accurate reconstructions across various viewpoints, aspect ratios, and even perspective effects. However, some images are genuinely ambiguous, like the Rubik’s Cube (bottom-right) or the shape is very extreme and our synthesis engine never synthesized that shape, like the container on the last row. 54

3.10	We show an example detection for which the reconstruction problem is fundamentally ill-posed (in our HOG feature space). Our brute-force strategy for enumerating all reconstructions can readily return multiple high-scoring interpretations, addressing a classic limitation of “inverse rendering” approaches.	55
4.1	Overview	58
4.2	We examine 3D shape categorization of cuboidal objects (left). Such objects share similar shape, so conventional folk wisdom might advocate the use of shape-invariant (or pose-normalized) representations for recognition (top) that are attractive because they (1) factor out shape (which seems uninformative when classifying objects with similar shape) and (2) can generalize to novel shapes not encountered in training data. We show that this approach is not optimal. One reason is that current methods produce small errors in geometric alignment, which can result in large fluctuations in the pose-normalized appearance. However, even with ground-truth alignment, pose-normalization is still not optimal. We demonstrate that pose-synthesis (bottom), a simple approach of augmenting training data with geometrically perturbed training samples, is a surprisingly effective strategy that allows for state-of-the-art categorization and automatic 3D alignment.	59
4.3	Proposed image-based rendering engine takes an image I , a set of 2D landmarks P and a set of target 2D landmark locations T as input, then it renders the cube in target view by warping each surface using a homography warp.	61
4.4	The image-based rendering engine works in three steps: (a) extracting background from the image. (b) rendering object in different poses using homography warps on object surfaces. (c) pasting the warped object on the background and filling the the holes using interpolation.	62
4.5	We visualize various strategies for achieving geometric-invariance in recognition. Please refer to the text for a detailed description of each strategy.	63

4.6	Given an image of an object, we show novel synthesized views generated by Pose-Synthesis (left) and the normalized view used by Pose-Normalized (right). Synthesized views look fairly realistic (because we can explicitly control and limit the degree of view synthesis), while the normalized views often have pixelation artifacts. The artifacts can arise from extrapolation of heavily-foreshortened cube faces (middle row) or small mistakes in the predicted 2D landmarks (bottom row).	66
4.7	Our dataset of cuboidal object categories, re-purposed from the SUN Primitive database [62]. This dataset includes variation in shapes (aspect ratios), viewpoint, backgrounds and clutter.	70
4.8	The center red column designates a test image, while the first six correspond to the NN-matches using our various representations. On the right , we visualize both the automatically-generated and ground-truth normalized image. Note how erroneous alignment create significant fluctuations in pose-normalized image, causing wrong NN-matches. Provided ground-truth alignment, pose-retargeting performs the best, while pose-synthesis performs best using automatic alignment.	71
4.9	Confusion matrices for various representation using CNN+SVM. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.	74
4.10	Confusion matrices for various representation using CNN+NN. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.	75
4.11	Confusion matrices for various representation using HOG+SVM. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.	76
4.12	Confusion matrices for various representation using HOG+NN. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.	77
4.13	Accuracy of all methods for differing amounts of real training images.	78

4.14	Analysis of various synthesis strategies. We show that synthesis based on uninformed pose prior does not perform very well. However, imposing weak prior through training data perturbation is very effective. The curves show that categorization accuracy decreases as the perturbation interval increases.	79
4.15	Pose estimation accuracy of Pose-Synthesis and Pose-Agnostic (where we assume all training images have been annotated with landmarks), compared to the method proposed in Chapter 3. Our models outperform prior art with as little as 5 training images per class.	80
4.16	Samples of our landmark localization results. We compare the approach in Chapter 3 of this thesis with pose-agnostic and pose-synthesis approaches.	81
4.17	Landmark localization results using pose-synthesis approach using differing amount of training data.	81

LIST OF TABLES

Table

4.1	Categorization accuracy of various approaches. Chance performance on this 9-class task is roughly 11%. The top-four methods are fully automatic, making use of predicted landmarks estimated using the method proposed in Chapter 3 when needed. The bottom-two make use of ground-truth (GT) landmarks. Pose-Synthesis, although simple, consistently outperforms Pose-Normalized and pose-retargeting representation.	72
4.2	Background synthesis effect in pose-synthesis approach. Similar to 4.1 the table shows categorization accuracy of various approaches. Chance performance on this 9-class task is roughly 11%. One hypothesis is that background provides useful contextual information for classification, thus pose-retargeted and pose-normalized representations are not using it. Here we show that the performance of pose-synthesis without background synthesis is even slightly better than pose-synthesis with background synthesis.	73

ACKNOWLEDGEMENTS

I feel extremely lucky for the opportunities I had that allowed me to write this thesis. I thank Deva for being a great adviser and mentor. He fueled and guided my curiosity and supported me along the journey.

I thank Charless Fowlkess, Aditi Majumder and Alex Ihler for their insight, feedback and support. I thank Ramesh Jain for his mentorship and advice. I thank Mehrdad Shahshahani and Amir Daneshgar for introducing me to research.

I thank Ali Farhadi, Hamed Pirsiavash, Amin Sadeghi, Mohammad Rastegari and Babak Saleh for sharing ideas with me. I thank Sam Halman, Golnaz Ghiasi, Bailey Kong, Yi Yang, Xiangxin Zhu, Chaitanya Desai, James Supancic, Gregory Rogez, Ral Daz, Dennis Park, Phuc Nguyen, Peiyun Hu, Shu Kong, Maryam Khademi, Julian Yarkony, Carl Vondrik and Mehdi Abbaspour for being awesome friends and labmates. I thank Sepehr Akhavan, Soheil Akhavan, Mahdi Khoshchehreh and Matias Giorgio for good times in Irvine.

I thank Vicarious for its generous support.

This thesis wouldn't be possible without my family's endless love and support.

CURRICULUM VITAE

Mohsen Hejrati

- 2010 B.S. Mathematical Sciences, Sharif University of Technology, Tehran, Iran
- 2012 M.S. Computer Science, University of California, Irvine, CA, USA
- 2015 Ph.D. Computer Science, University of California, Irvine, CA, USA

ABSTRACT

Analyzing 3D Objects in 2D Images

by

Mohsen Hejrati

Ph.D. In Computer Science

University of California, Irvine, 2015

Professor Deva Ramanan, Chair

Robots are mechanically capable of doing many tasks, carrying loads, precisely manipulating objects, picking and packing or collaborating with humans. However, they require accurate 3D perception of objects and surrounding environment to do these tasks autonomously. Traditional methods build 3D representation of the scene using structure from motion techniques or depth sensors, while more recent approaches use statistical models to learn geometry and appearance of 3D objects and scenes. This thesis investigates approaches to represent, learn and analyze 3D objects in natural images. We first propose two new methods for 3D object recognition and pose estimation in single 2D images. Second, we study various geometric representations for the novel task of primitive 3D shape categorization.

We propose two novel approaches for recognizing 3D objects: (1) *Aligning a 3D model* to detected 2D landmarks, where we propose a novel method based on deformable-part models to propose candidate detections and 2D estimates of shape, then these estimates are refined by using an explicit 3D model of shape and viewpoint. (2) An *analysis by synthesis* approach where a forward synthesis model constructs possible geometric interpretations of the world, and then selects the interpretation that

best agrees with the measured visual evidence. We show state of the art performance for detection and pose estimation on two challenging 3D object recognition datasets of cars and cuboids.

3D object recognition methods focus on modeling 3D shape of the objects, however, many objects may have similar 3D shape (washing machines, cabinets and microwave are all cuboidal), thus recognizing them require reasoning about appearance and geometry at the same time. The natural approach for recognition might extract pose-normalized appearance features. Though such approaches are extraordinarily common in the literature, in this thesis we demonstrate that they are *not optimal*. Instead, we introduce methods based on pose-synthesis, a somewhat simple approach of augmenting training data with geometrically perturbed training samples. We demonstrate that synthesis is a surprisingly simple but effective strategy that allows for state-of-the-art categorization and automatic 3D alignment.

CHAPTER I

Introduction

“ There are things known and there are things unknown, and in between are the doors of *perception*. ”

Aldous Huxley

Perception is crucial for intelligence. Visual perception is inevitable to build intelligent agents that are capable of understanding and interacting with the world as we humans do. A large body of work is focused on classifying an image into one of many labels, for instance [13, 61, 46, 40]. Object detection methods aim to provide location of objects, pedestrians, faces, etc in the image [15, 12, 68, 30]. Many methods build spatial understanding of images by reasoning about surfaces (horizontal, vertical, etc.) in the images [28, 26, 23], while other are use geometric alignment in order to reconstruct indoor scenes or buildings [27, 25]. A grand challenge in machine vision is the task of understanding 3D structures from 2D images and creating rich 3D representation and reconstruction of the images (Figure 1.1).

3D computer vision has been studied for decades. A plethora of work focus on 3D reconstruction from multiple images by finding consistent interpretation across images. On the other hand, many single-image approaches are based on learning structure in natural images or objects. Also, the advancement of depth sensors paved the way for more recent approaches that are more focused on 3D representation and recognition instead of reconstruction.

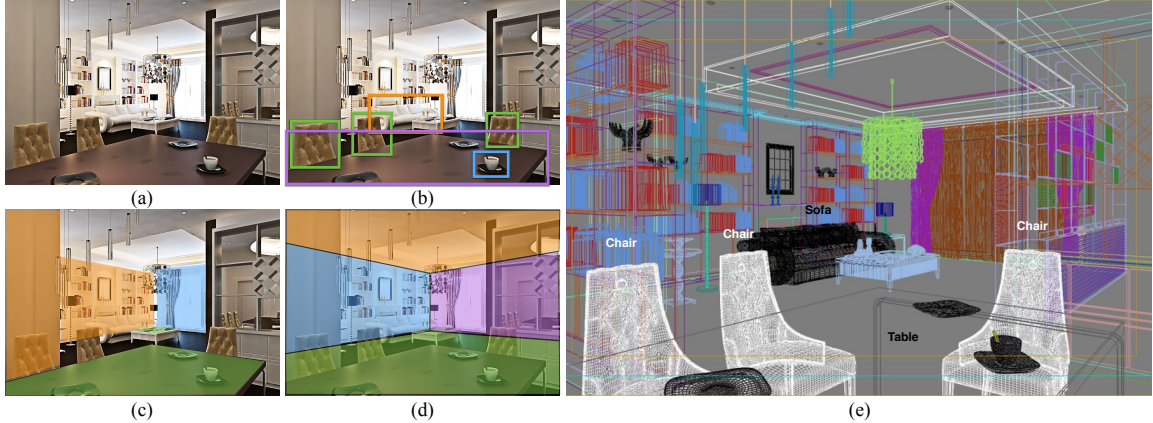


Figure 1.1: A large body of work is focused on classifying an image into one of many labels (a), object detection methods aim to provide location of objects in the image (b). Many methods build spatial understanding of images by reasoning about surfaces in the images (c), while other are based on aligning geometric shapes in order to reconstruct indoor scenes or buildings (d). Understanding 3D structures from 2D images and creating rich 3D representation and reconstruction is a defining challenge in machine vision (e).

In this thesis, we concentrate on single images. A category of single image 3D reconstruction methods work by analyzing small image patches and then form a coherent global reconstruction. On the other hand, some approaches fit geometric shapes to the whole scene to estimate coarse 3D structure of the scene. We focus on 3D recognition, reconstruction and categorization of objects.

In this thesis we study 3D object recognition, reconstruction and categorization in single 2D images. In Chapter 2 and 3 we propose two novel approaches for 3D object recognition and reconstruction and in Chapter 4 we investigate various representation methods for 3D object categorization.

Various approaches for 3D object recognition and reconstruction from single images exist in the literature. Historical methods based on *geometric indexing* work by aligning 3D models to image data. They typically detect a sparse set of local features and treat alignment as a feature correspondence problem. [19, 37, 22]. *View-based modeling* is another popular approach which partition a 3D object into view-specific

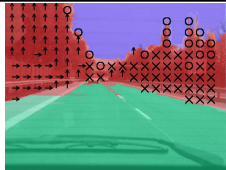

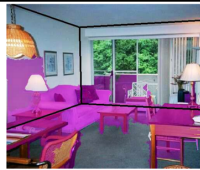
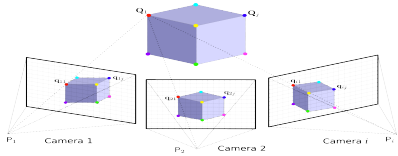
	Patch	Object	Scene
Single Image			
Multiple Images			

Figure 1.2: A large body of work focus on 3D reconstruction from multiple images. In this thesis we concentrate on single images. Some methods reconstruct 3D structure of single images by analyzing small image patches or pixels. Some approaches are based on fitting geometric shapes to the whole scene. We focus on 3D recognition, reconstruction and categorization of objects.

2D sub-categories and a model is then trained separately for each of these categories. [41, 24, 18, 68, 54]. We will discuss the related work more deeply inside each chapter.

In Chapter 2 we propose a two-stage model for 3D recognition which is first **localizes 2D landmarks** by reasoning about 2D shape and appearance and the **aligns** a 3D model to these 2D landmarks using non-rigid structure from motion techniques.

In Chapter 3 we introduce a new approach for 3D recognition based on an **analysis by synthesis** strategy. A synthesis model constructs possible geometric interpretations of the world and then selects the interpretation that best agrees with the measured visual evidence. One benefit of such an approach is that recognition is inherently (re)constructive.

In Chapter 4 we address the question of how to use the geometric-reasoning engines studied in Chapter 2 and 3 for **categorical recognition**, focusing on cuboidal object categories. We evaluate both categorization and 3D shape estimation using a variety of representations capturing both appearance and geometry.

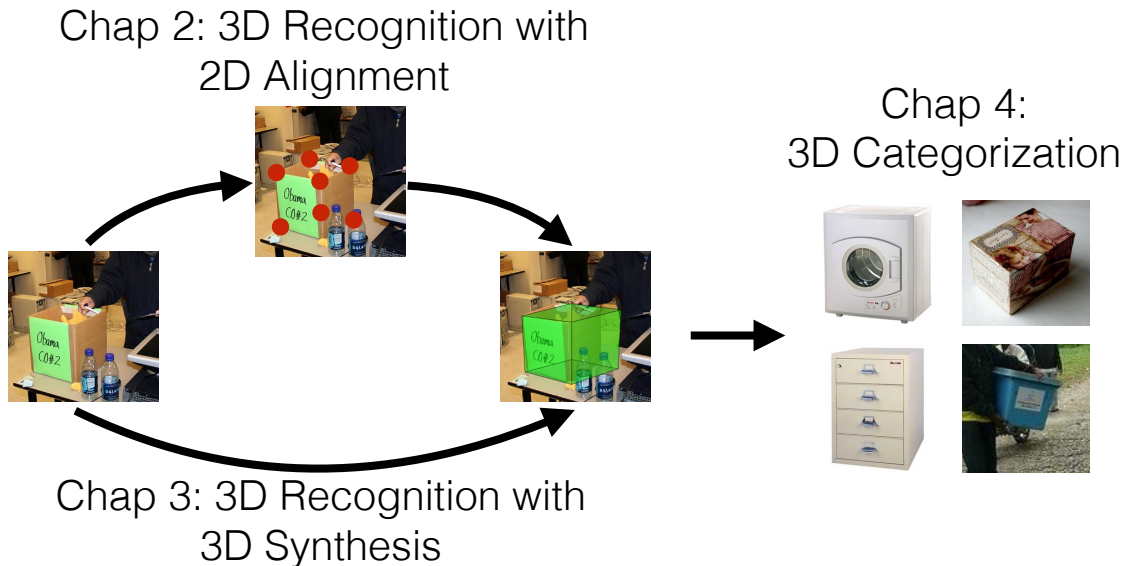


Figure 1.3: We focus on 3D object recognition, reconstruction and categorization in single 2D images. Chapter 2 and 3 present two novel approaches for 3D object recognition and reconstruction and Chapter 4 investigates various representation methods for 3D object categorization.

1.1 3D Recognition with 2D Alignment

In Chapter 2 we propose a novel approach for 3D object recognition based on aligning a 3D morphable model to 2D landmarks. Contemporary recognition methods tend to build statistical models of 2D appearance, consisting of classifiers trained with large training sets using engineered appearance features. Successful examples include face detectors [60], pedestrian detectors [11], and general object-category detectors [18]. Such methods are usually limited to coarse 2D output, such as bounding-boxes. We develop a model that detects objects, estimates camera viewpoint, and recovers 3D landmarks configurations and their visibility with state-of-the-art accuracy. It does so by reasoning about appearance, 3D shape, and camera viewpoint through the use of 2D structured, relational classifiers and 3D geometric subspace models.

While deformable models and pictorial structures [18, 63, 21] are known to successfully model articulation, 3D viewpoint is still not well understood. The typical

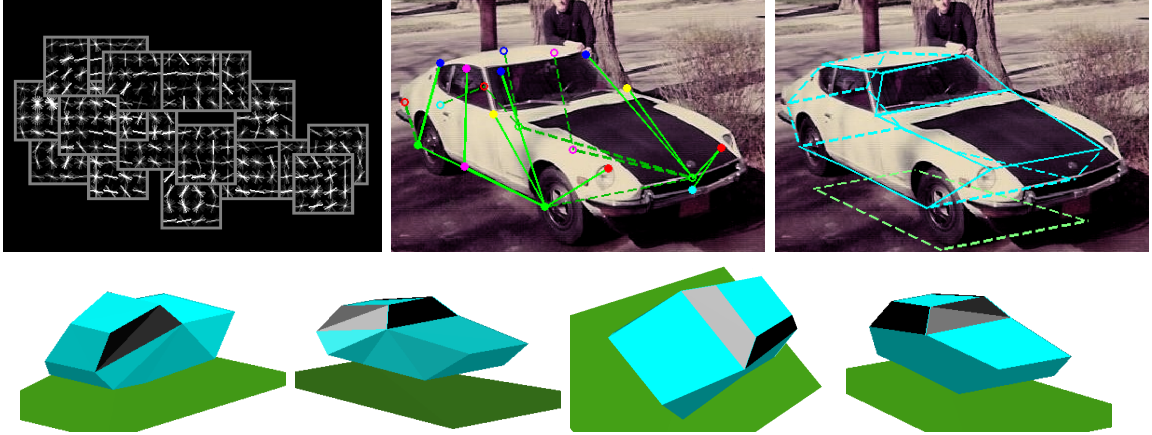


Figure 1.4: Chapter 2 introduces a two-stage model for detecting and analyzing the 3D shape of objects in unconstrained images. In the first stage, our model reasons about 2D appearance and shape using variants of deformable part models (DPMs). Our 2D model localizes even fully-occluded landmarks, shown as hollow circles and dashed lines in (**top-middle**). We feed this output to our second stage, which directly reasons about 3D shape and camera viewpoint. We show the reconstructed 3D model on (**top-right**). The **bottom** 3 viewpoints.

solution is to “discretize” viewpoint and build multiple view-based models tuned for each view (frontal, side, 3/4...). We introduce a two-stage approach that first reasons about 2D shape and appearance variation, and then reasons explicitly about 3D shape and viewpoint given 2D correspondences from the first stage.

2D shape and appearance: Our first stage models 2D shape and appearance using a variant of deformable part models (DPMs) designed to produce reliable 2D landmark correspondences. Our approach differs from traditional view-based models in that it is *compositional*; it “cuts and pastes” together different sets of local view-based templates to model a large set of global viewpoints. We use global mixtures of trees with local mixtures of “part” or landmark templates. Global mixtures capture constraints on visibility and shape (headlights are only visible in certain views at certain locations), while local mixtures capture constraints on appearance (headlights look different in different views).

3D shape and viewpoint: Our second layer processes the 2D output of our

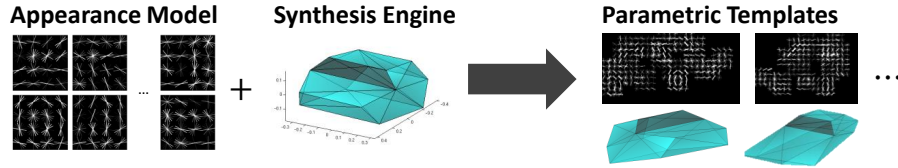


Figure 1.5: In Chapter 3 we propose an analysis by synthesis approach for 3D object recognition. We describe a method for synthesizing a large set of discriminative templates, each associated with a candidate 3D reconstruction of an object, then the interpretation that best agrees with the test image is selected.

first stage, incorporating global shape constraints arising from 3D shape variation and viewpoint. To capture viewpoint constraints, we model landmarks as weak-perspective projections of a 3D object. To capture within-class variation, we model the 3D shape of any object instance as a linear combination of 3D basis shapes. We use tools from nonrigid structure-from-motion (SFM) to both learn and enforce such models using 2D correspondences.

1.2 3D Recognition with 3D Synthesis

In contrast to Chapter 2 where a two stage model is used to recognize and reconstruct 3D objects, in Chapter 3, we describe a single model that simultaneously detects instances of general object categories, and reports a detailed 3D reconstruction of each instance. *Analysis by synthesis* strategy works by synthesizing possible geometric interpretations of the world, and then selecting the one that matches best with the measured visual evidence.

“Inverse rendering” approach to computer vision is wildly challenging for two primary reasons. (1) It is difficult to build accurate generative models that capture the full complexity of the visual world. (2) Even given such a model, inverting it is difficult because the problem is fundamentally ill-posed (different reconstructions may generate similar images) and full of local minima. Our approach addresses both difficulties. (1) Instead of generating pixel values, we synthesize visual templates

defined on invariant (HOG) features. (2) We describe a “brute-force” approach to inference that efficiently searches through a large number of candidate reconstructions, returning the optimal one (or multiple likely candidates, if desired).

1.3 3D Categorization

Chapter 4 focuses on the question of how to use the geometric-reasoning engines studied in Chapter 2 and 3 for *categorical recognition*. We evaluate both categorization and 3D shape estimation using a variety of representations capturing both appearance and geometry.

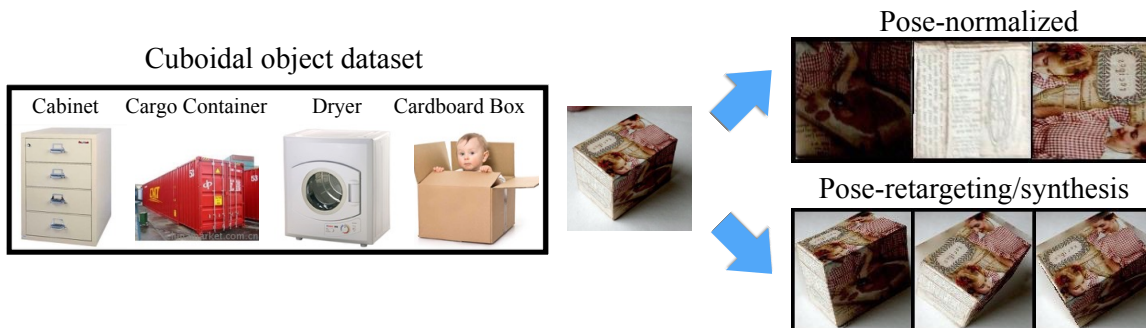


Figure 1.6: In Chapter 4 we examine how to use the geometric-reasoning engines proposed in Chapter 2 and 3 for *categorical recognition*. We evaluate 3D shape categorization of cuboidal objects (**left**). Such objects share similar shape, so conventional folk wisdom might advocate the use of shape-invariant (or pose-normalized) representations for recognition (**top**) that are attractive because they (1) factor out shape (which seems uninformative when classifying objects with similar shape) and (2) can generalize to novel shapes not encountered in training data. We show that this approach is not optimal. We demonstrate that pose-synthesis (**bottom**), a simple approach of augmenting training data with geometrically perturbed training samples, is a surprisingly effective strategy that allows for state-of-the-art categorization and automatic 3D alignment.

The most natural approach would use the estimated alignment to extract pose-normalized appearance features. For a cuboidal object, one might represent the appearance of each cuboidal face in a fronto-parallel view (Figure 1.6). Many state-of-

the-art systems for recognition (such as faces [52, 35], cars [36], animal species [8, 17], or general attributes [65]) similarly normalize landmarks/keypoints into a canonical coordinate frame during training and or testing. For example, the vast majority of face recognition systems work by detecting landmarks, warping the image such that landmarks are aligned into a canonical frontal view, and classifying the warped (pose-normalized) appearance [66, 31]. Importantly, normalization allows one to (1) factor out “nuisance” variables such as viewpoint and aspect/shape during recognition, and (2) generalize to poses not seen in training data.

We demonstrate that pose-normalization is *not* the optimal strategy for dealing with appearance variation due to pose. One explanation maybe the inaccuracy of current systems for pose estimation - small misalignments in the predicted pose may cause large errors in the pose-normalized appearance. We show that *even with ground-truth alignment on test images*, pose-normalization is still not optimal. In short, pose-normalization (a) removes geometric cues that maybe helpful for recognition (washing machines may have differing aspect ratio from microwaves) and (b) artificially re-weights foreshortened regions of the objects. To address these limitations, we describe an approach that warps (or *retargets*) training examples to the shape and viewpoint of a particular detected instance, and performs recognition using this retargeted training set.

We demonstrate that pose-retargeting is the optimal approach given ground-truth alignment, but falls short given the accuracy of current systems that estimate cuboidal alignments. To address this limitation, we evaluate another approach that *pre-synthesizes* a large set of possible target poses. The synthesized set is used to train a practical system that jointly performs categorization and 3D alignment, at a level of accuracy that surpasses the current state-of-the-art. Importantly, synthesis also allows our system to generalize to unseen viewpoints and shapes not seen in the training set *without* requiring pose-normalization.

CHAPTER II

3D Recognition with 2D

Alignment

“ We can only see a short distance ahead, but we can see plenty there that needs to be done. ”

Alan Turing

2.1 Introduction

In this chapter, we propose a novel approach for 3D object recognition based on aligning a 3D morphable model to 2D landmarks. Classic approaches based on 3D geometric models [4] could sometimes exhibit brittle behavior on cluttered, “in-the-wild” images. Contemporary recognition methods tend to build statistical models of 2D appearance, consisting of classifiers trained with large training sets using engineered appearance features. Successful examples include face detectors [60], pedestrian detectors [11], and general object-category detectors [18]. Such methods seem to work well even in cluttered scenes, but are usually limited to coarse 2D output, such as bounding-boxes.

We attempt to combine the two approaches, with a focus on statistical, 3D geometric models of objects. Specifically, we focus on the practical application of detecting and analyzing cars in cluttered, unconstrained images. We refer the reader to our

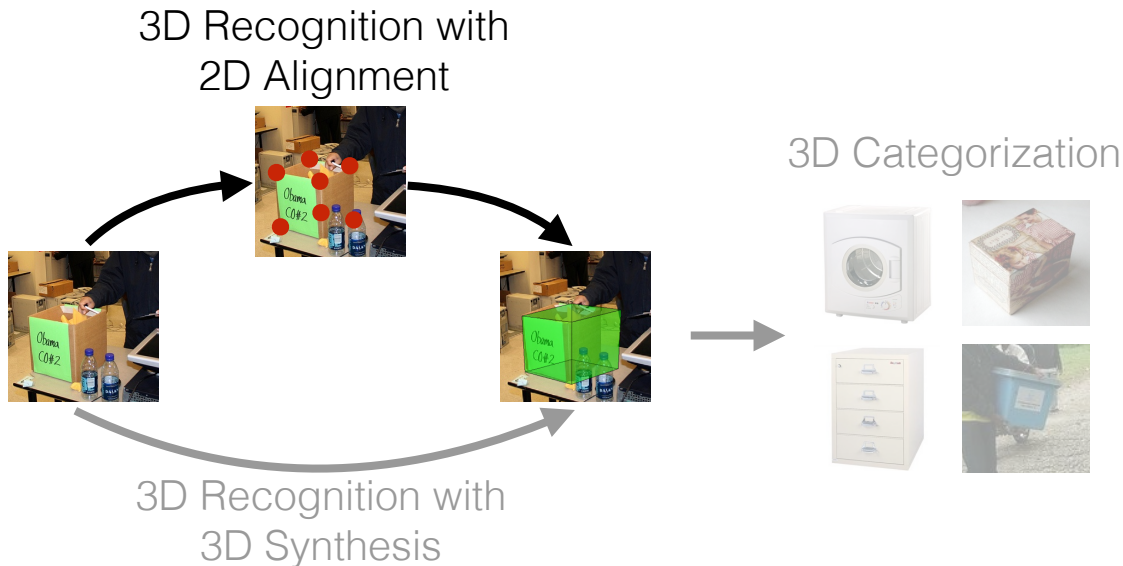


Figure 2.1: Overview

results (Figure 2.5) for a sampling of cluttered images that we consider. We develop a model that detects cars, estimates camera viewpoint, and recovers 3D landmarks configurations and their visibility with state-of-the-art accuracy. It does so by reasoning about appearance, 3D shape, and camera viewpoint through the use of 2D structured, relational classifiers and 3D geometric subspace models.

While deformable models and pictorial structures [18, 63, 21] are known to successfully model articulation, 3D viewpoint is still not well understood. The typical solution is to “discretize” viewpoint and build multiple view-based models tuned for each view (frontal, side, 3/4...). One advantage of such a “brute-force” approach is that it is computationally efficient, at least for a small number of views. Fine-grained 3D shape estimation may still be difficult with such a strategy. On the other hand, it is difficult to build models that reason directly in 3D because the “inverse-rendering” problem is hard to solve. We introduce a two-stage approach that first reasons about 2D shape and appearance variation, and then reasons explicitly about 3D shape and viewpoint given 2D correspondences from the first stage. We show that “inverse-rendering” *is* feasible by way of 2D correspondences.

2D shape and appearance: Our first stage models 2D shape and appearance using a variant of deformable part models (DPMs) designed to produce reliable 2D landmark correspondences. Our approach differs from traditional view-based models in that it is *compositional*; it “cuts and pastes” together different sets of local view-based templates to model a large set of global viewpoints. We use global mixtures of trees with local mixtures of “part” or landmark templates. Global mixtures capture constraints on visibility and shape (headlights are only visible in certain views at certain locations), while local mixtures capture constraints on appearance (headlights look different in different views). We use this model to efficiently generate candidate 2D detections that are refined by our second 3D stage. One salient aspect of our 2D model is that it reports 2D locations of all landmarks including occluded ones, each augmented with a visibility flag.

3D shape and viewpoint: Our second layer processes the 2D output of our first stage, incorporating global shape constraints arising from 3D shape variation and viewpoint. To capture viewpoint constraints, we model landmarks as weak-perspective projections of a 3D object. To capture within-class variation, we model the 3D shape of any object instance as a linear combination of 3D basis shapes. We use tools from nonrigid structure-from-motion (SFM) to both learn and enforce such models using 2D correspondences. Crucially, we make use of occlusion reports generated by our local view-based templates to estimate morphable 3D shape and camera viewpoint. Such morphable models are typically learned by applying subspace methods (such as PCA or an SVD) directly to 3D landmarks or mesh vertices [5]. We show that similar methods can also be applied to 2D projections under an affine camera model that can view “occluded” landmarks. Because our tree models report the 2D location of such occluded parts, we are able to learn and estimate morphable 3D shape using off-the-shelf tools from nonrigid structure-from-motion (SFM) [57]. Because directly evaluating 3D output is difficult (due to lack of ground-truth), we

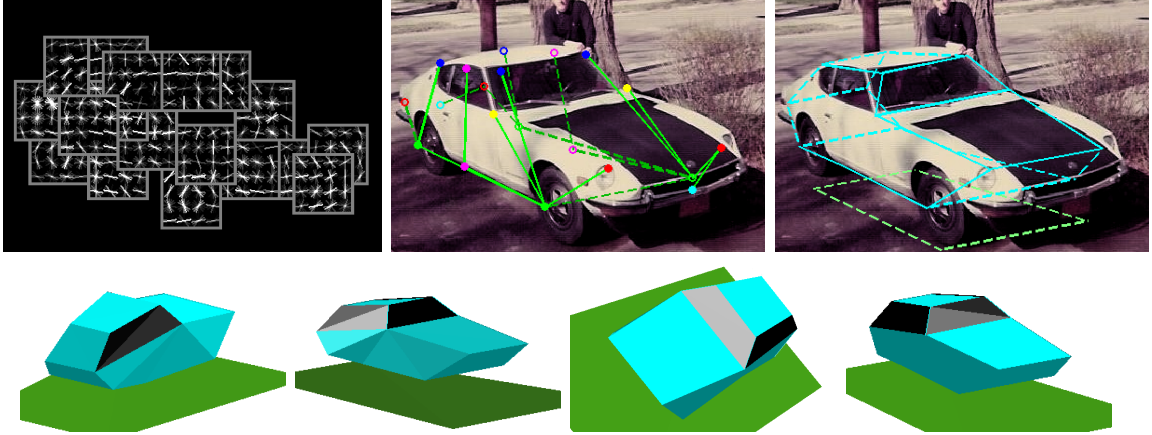


Figure 2.2: A two-stage models for detecting and analyzing the 3D shape of objects in unconstrained images is proposed. In the first stage, our models reason about 2D appearance and shape using variants of deformable part models (DPMs). We use global mixtures of trees with local mixtures of gradient-based part templates (**top-left**). Global mixtures capture constraints on visibility and shape (headlights are only visible in certain views at certain locations), while local mixtures capture constraints on appearance (headlights look different in different views). Our 2D models localize even fully-occluded landmarks, shown as hollow circles and dashed lines in (**top-middle**). We feed this output to our second stage, which directly reasons about 3D shape and camera viewpoint. We show the reconstructed 3D model and associated ground-plane (assuming its parallel to the car body) on (**top-right**). The **bottom** row shows 3D reconstructions from four novel viewpoints.

qualitatively show that our approach produces accurate 2D landmark localization and visibility labels, particularly-so for occluded landmarks.

Because our model is very efficient, we are still able to train it discriminatively in a structured prediction framework using massive training data sets (hundreds of positives and hundreds of millions of negative examples). This allows us to outperform even state-of-the-art methods in detection accuracy [18], while obtaining accurate 3D estimates “for free”.

Partial occlusions: Most approaches for occlusion “zero-out” responses of occluded regions within a visual template [59, 20]. Our model differs in that we explicitly search for visual evidence consistent with an occlusion (say, T-junctions), and perhaps more importantly, look for occlusions that are consistent with relational constraints

between local templates.

2.2 Related Work

We focus most on recognition methods that deal explicitly with 3D viewpoint variation.

Voting-based methods: One approach to detection and viewpoint classification is based on bottom-up geometric voting, using a Hough transform or geometric hashing. Images are first processed to obtain a set of local feature detections. Each detection can then vote for both an object location and viewpoint. Examples include [22] and implicit shape models [1, 53]. Our approach differs in that we require no initial feature detection stage, and instead we reason about all possible geometric configurations and occlusion states.

View-based models: A popular approach is to partition an object category into view-specific sub-categories, and train a detector for each. Early successful approaches included multi-view face detection [48, 33]. Recent approaches based on view-based deformable part models include [41, 24, 18]. Our model differs in that we use a single representation that directly generates multiple views. Finally, one can share local parts across views [68, 54].

Aspect-based models: One can augment view-based models to share local parts across views [54, 43, 67]. This typically requires reasoning about topological changes in viewpoint; certain parts or features can only be visible in certain view due to self-occlusion. One classic representation for encoding such visibility constraints is an aspect graph [7]. [68] model such topological constraints with global mixtures with varying tree structures. Our model is similar to such approaches, except that we use a decomposable notion of aspect; we simultaneously reason about global and semi-local changes in visibility using local part mixtures with global co-occurrence constraints.

3D models: One can also directly reason about local features and their geometric arrangement in a 3D coordinate system [47, 50, 70]. Though such models are three-dimensional in terms of their underlying representation, run-time inference usually proceeds in a bottom-up manner, where detected features vote for object locations. To handle non-Gaussian observation models, [39] evaluate randomly sampled model estimates within a RANSAC search. Our approach is closely related to the recent work of [44], which also uses a deformable part model (DPM) to capture viewpoint variation in cars. Though they learn spatial constraints in a 3D coordinate frame, their model at run-time is equivalent to a view-based model, where each view is modeled with a star-structured DPM. Our model differs in that we directly reason about the location of fully-occluded landmarks, we model an exponential number of viewpoints by using a compositional representation, and we produce continuous 3D shapes and camera viewpoints associated with each detection using only 2D training data.

Finally, we represent the space of 3D models of an object category using a set of basis shapes, similar to the morphable models of [5]. To estimate such models from 2D data, we adapt methods designed for tracking morphable shapes to 3D object category recognition [57, 55].

2.3 2D Shape and Appearance

We first describe our 2D model of shape and appearance. We write it as a scoring function with linear parameters. Given training data of images and ground-truth landmark locations (in 2D), we show how to learn parameters in a linear classification framework.

Our model can be seen as an extension of the flexible mixtures-of-part model [63], which itself augments a deformable part model (DPM) [18] to reason about local mixtures. Our model differs its encoding of occlusion states using local mixtures,

as well as the introduction of global mixtures that enforce occlusions and spatial geometry consistent with changes in 3D viewpoint. We take care to design our model so as to allow for efficient dynamic-programming algorithms for inference.

Let I be an image, $p_i = (x, y)$ be the pixel location for part i and $t_i \in \{1..T\}$ be the local mixture component of part i . As an example, part i may correspond to a front-left headlight, and t_i can correspond to different appearances of a headlight in frontal, side, or three-quarter views. A notable aspect of our model is that we estimate landmark locations for all parts in all views, even when they are fully occluded. We will show that local mixture variables perform surprisingly well at modeling complex appearances arising from occlusions.

Let $i \in V$ where V is the set of all landmarks. We consider different relational graphs $G_m = (V, E_m)$ where E_m connects pairs of landmarks constrained to have consistent locations and local mixtures in global mixture m . We can loosely think of m as a “global viewpoint”, though it will be latently estimated from the data. We use the lack of subscript to denote the set of variables obtained by iterating over that subscript; e.g., $p = \{p_i : i \in V\}$. Given an image, we score a collection of landmark locations and mixture variables

$$S(I, p, t, m) = \sum_{i \in V} \left[\alpha_i^{t_i} \cdot \phi(I, p_i) \right] + \sum_{ij \in E_m} \left[\beta_{ijm}^{t_i, t_j} \cdot \psi(p_i - p_j) + \gamma_{ijm}^{t_i, t_j} \right] \quad (2.1)$$

Local model: The first term scores the appearance evidence for placing a template $\alpha_i^{t_i}$ for part i , tuned for mixture t_i , at location p_i . We write $\phi(I, p_i)$ for the feature vector (e.g., HOG descriptor [11]) extracted from pixel location p_i in image I . Note that we define a template even for mixtures t_i corresponding to fully-occluded states. One may argue that no image evidence should be scored during an occlusion; we take the view that the learning algorithm can decide for itself. It may choose to

learn a template of all zeros (essentially ignoring image evidence) or it may find gradient features statistically correlated with occlusions (such as t-junctions). Unlike the remaining terms in our scoring function, the local appearance model is *not* dependent on the global mixture/viewpoint. We show that this independence allows our model to compose together different local mixtures to model a single global viewpoint.

Relational model: The second term scores relational constraints between pairs of parts. We write $\psi(p_i - p_j) = \begin{bmatrix} dx & dx^2 & dy & dy^2 \end{bmatrix}$, a vector of relative offsets between part i and part j . We can interpret $\beta_{ijm}^{t_i, t_j}$ as the parameters of a spring specifying the relative rest location and quadratic spring penalty for deviating from that rest location. Notably, this spring depends on part i and j , the local mixture components of part i and j , and the global mixture m . This dependency captures many natural constraints due to self-occlusion; for example, if a car’s left-front wheel lies to the right of the other front wheel (in image space), then it is likely self-occluded. Hence it is crucial that local appearance and geometry depend on each other. The last term $\gamma_{ijm}^{t_i, t_j}$ defines a co-occurrence score associated with instancing local mixture t_i and t_j , and global mixture m . This encodes the constraint that, if the left front headlight is occluded due to self occlusion, the left front wheel is also likely occluded.

Global model: We define different graphs $G_m = (V, E_m)$ corresponding to different global mixtures. We can loosely think of the global variable m as capturing a coarse, quantized viewpoint. To ensure tractability, we force all edge structures to be tree-structured. Intuitively, different relational structures may help because occluded landmarks tend to be localized with less reliability. One may expect occluded/unreliable parts should have fewer connections (lower degrees in G_m) than reliable parts. Even for a fixed global mixture m , our model can generate an exponentially-large set of appearances $|V|^T$, where T is the number of local mixture types. We show such a model outperforms a naive view-based model in our experiments.

2.3.1 Inference

Inference corresponds to maximizing (2.1) with respect to landmark locations p , local mixtures t , and global mixtures m :

$$S^*(I) = \max_m [\max_{p,t} S(I, p, t, m)] \quad (2.2)$$

We optimize the above equation by enumerating all global mixtures m , and for each global mixture, finding the optimal combination of landmark locations p and local mixtures t by dynamic programming (DP). To see that the inner maximization can be optimized by DP, let us define $z_i = (p_i, t_i)$ to denote both the discrete pixel position and discrete mixture type of part i . We can rewrite the score from (2.1) for a fixed image I and global mixture m with edge structure E as:

$$S(z) = \sum_{i \in V} \phi_i(z_i) + \sum_{ij \in E} \psi_{ij}(z_i, z_j), \quad (\text{for a fixed } I \text{ and } m) \quad (2.3)$$

$$\text{where } \phi_i(z_i) = \alpha_i^{t_i} \cdot \phi(I, p_i) \quad \text{and} \quad \psi_{ij}(z_i, z_j) = \beta_{ijm}^{t_i, t_j} \cdot \psi(p_i - p_j) + \gamma_{ijm}^{t_i, t_j}$$

From this perspective, it is clear that our model (conditioned on I and m) is a discrete, pairwise Markov random field (MRF). When $G = (V, E)$ is tree-structured, one can compute $\max_z S(z)$ with dynamic programming [63].

2.3.2 Learning

We assume we are given training data consisting of image-landmark triplet $\{I_n, p_{in}, o_{in}\}$, where landmarks are augmented with an additional discrete visibility flag o_{in} . With a slight abuse of notation, we use n to denote an instance of a training image. We use $o_{in} \in \{0, 1, 2\}$ to denote visible, self-occlusion, and other-occlusion respectively, where other occlusion corresponds to a landmark that is occluded by another object (or the

image border). We now show how to augment this training set with local mixtures labels t_{in} , global mixtures labels m_n , and global edge structures E_m . Essentially, we infer such mixture labels using probabilistic algorithms for generating local/global clusters of 2D landmark configurations. We then use this inferred mixture labels to train the linear parameters of the scoring function (2.1) using supervised, max-margin methods.

Learning local mixtures: We use the clustering algorithm described in [14, 6] to learn local part mixtures. [63] obtain local mixture labels by clustering the relative location of a landmark (part, in their language) relative to its parent in the relational graph G . We wish to learn local mixtures in a manner agnostic to graph structure (since that will be viewpoint dependent). Inspired by the poselet framework [6], we construct a “local-geometric-context” vector for each part, and obtain landmark mixture labels by grouping landmark instances with similar local geometry. Specifically, for each landmark i and image n , we construct a K -element vector g_{in} that defines the 2D relative location of a landmark with respect to the other K landmarks in instance n , normalized for the size of that training instance. We construct sets of features $\text{Set}_{ij} = \{g_{in} : n \in 1..N \text{ and } o_{in} = j\}$ corresponding to each part i and occlusion state j .

We separately cluster each set of vectors using K -means, and then interpret cluster membership as mixture label t_{in} . This means that, for landmark i , a third of its T local mixtures will model visible instances in the training set, a third will model self-occlusions, and a third will capture other-occlusions.

Learning relational structure: Given landmark positions p_{in} and local mixture labels t_{in} , we simultaneously learn global mixtures m_n and edge structure E_m with a probabilistic model of $z_{in} = (p_{in}, t_{in})$. We find the global mixtures and edge structure that maximizes the probability of the observed $\{z_{in}\}$ labels. Probabilistically speaking, our spatial spring model is equivalent to a Gaussian model (who’s mean and

covariance correspond to the rest location and rigidity), making estimation relatively straightforward.

We first describe the special case of a single global mixture, for which the most-likely tree E can be obtained by maximizing the mutual information of the labels using the Chow-Liu algorithm. In our case, we find the maximum-weight spanning tree in a fully connected graph whose edges are labeled with the mutual information (MI) between $z_i = (p_i, t_i)$ and $z_j = (p_j, t_j)$. Hence *both* spatial consistency and mixture consistency are used when learning the relational structure. Given observations of z from labeled training data, recall that Chow-Liu finds the tree that maximizes the likelihood of the training data with the following:

$$P(z) = P(z_1) \prod_{ij \in E} P(z_i | z_j) = \left[\prod_{i \in V} P(z_i) \right] \prod_{ij \in E} \frac{P(z_i, z_j)}{P(z_i)P(z_j)} \quad (2.4)$$

The second parameterization is convenient because it is symmetric. We can find the tree structure E that maximizes the log likelihood of a set of observations by computing

$$\max_E \sum_{ij \in E} MI(z_i, z_j) \quad \text{and} \quad MI(z_i, z_j) = \sum_{z_i, z_j} P(z_i, z_j) \log \frac{P(z_i, z_j)}{P(z_i)P(z_j)} \quad (2.5)$$

The summation is over the empirical distribution encountered in a training set. The maximization is equivalent to finding the maximum weight spanning tree in a completely connected graph, where weights are given by the mutual information between two variables. Let us write out the mutual information between our joint location/mixture variables z_i, z_j . Their joint can always be written as:

$$P(z_i, z_j) = P(p_i, p_j | t_i, t_j) P(t_i, t_j) \quad (2.6)$$

We will assume the first term is a Gaussian whose mean and covariance depend on the discrete values of t_i, t_j , and the second term is a discrete table of co-occurrence priors.

Since $\sum_{p_i, p_j} P(p_i, p_j | t_i, t_j) = 1$, We can rewrite the $MI(t_i, t_j)$:

$$\begin{aligned}
MI(t_i, t_j) &= \sum_{t_i, t_j} P(t_i, t_j) \log \frac{P(t_i, t_j)}{P(t_i)P(t_j)} \\
&= \sum_{t_i, t_j} P(t_i, t_j) \log \frac{P(t_i, t_j)}{P(t_i)P(t_j)} \sum_{p_i, p_j} P(p_i, p_j | t_i, t_j) \\
&= \sum_{t_i, t_j} \sum_{p_i, p_j} P(p_i, p_j | t_i, t_j) P(t_i, t_j) \log \frac{P(t_i, t_j)}{P(t_i)P(t_j)}
\end{aligned} \tag{2.7}$$

Plugging in (2.6) and (2.7) into (2.5):

$$MI(z_i, z_j) = MI(t_i, t_j) + \sum_{t_i, t_j} P(t_i, t_j) MI(p_i, p_j | t_i, t_j) \tag{2.8}$$

$MI(t_i, t_j)$ can be directly computed from the empirical joint frequency of mixture labels in the training set. $MI(p_i, p_j | t_i, t_j)$ is the mutual information of the Gaussian random variables for the location of landmarks i and j given a fixed pair of discrete mixture types t_i, t_j ; this again is readily obtained by computing the determinant of the sample covariance of landmark i and j , estimated from the training data. Hence *both* spatial consistency and mixture consistency are used when learning our relational structure.

$$MI(p_i, p_j | t_i, t_j) = \frac{1}{2} \log \frac{|\Sigma_{ii|t_i}| |\Sigma_{jj|t_j}|}{|\Sigma_{ij|t_i t_j}|}$$

where $\Sigma_{ii|t_i} = E[p_i p_i^T | t_i]$ and $\Sigma_{ij|t_i, t_j} = E[(p_i, p_j)(p_i, p_j)^T | t_i, t_j]$.

Learning structure and global mixtures: To simultaneously learn global mixture labels m_n and edge structures associated with each mixture E_m , we use an EM algorithm for learning mixtures of trees, following Meila and Jordan. We

iterate between inferring distributions over tree mixture assignments (the E-step) and estimating the tree structure (the M-step). Notably, the M-step makes use of the Chow-Liu algorithm. Formally speaking, the global mixture model can be written as

$$P(z) = \sum_m P(m)P(z|m) \quad \text{where} \quad P(z|m) = \left[\prod_{i \in V} P(z_i) \right] \prod_{ij \in E_m} \frac{P(z_i, z_j)}{P(z_i)P(z_j)} \quad (2.9)$$

One can write the expected complete log-likelihood of the observed labels $\{z\}$, where θ are the model parameters (Gaussian spatial models, local mixture co-occurrences and global mixture priors) to be maximized and the global mixture assignment variables $\{m_n\}$ are the hidden variables to be marginalized:

$$L(q, \theta) = E_{q(m)}[\log P(z, m|\theta)] \quad (2.10)$$

The EM algorithm performs coordinate ascent on the expected complete log-likelihood from (2.10), iterating the following steps:

1. E step $q(m_n) = P(m_n|z_n, \theta) \quad \forall n$
2. M step $(\theta, \{E_m\}) = \operatorname{argmax}_{\theta, \{E_m\}} E_{q(m)}[\log P(z, m|\theta)]$

Step 1 is performed by computing the likelihood of each data example z_n under each tree, multiplying by the prior of that tree mixture, and normalizing over all trees. Step 2 computes model parameters, including springs, local mixture co-occurrences, global mixture priors, as well as the most likely tree structure for each mixture. Model parameters are computed by weighted sample estimates of means, variances,

and frequency counts, where weights are given by the posterior probability of an example belonging to a particular tree mixture. The tree structures are computed using the Chow-Liu algorithm using weighted mutual information estimates, where weights (again) are posterior probability of an example belonging to a particular tree mixture.

Learning parameters: The previous steps produces local/global mixture labels and edge structures. Treating these as “ground-truth”, we now define a supervised max-margin framework for learning model parameters. To do so, let us write the landmark position labels p_n , local mixtures labels t_n , and global mixture label m_n collectively as y_n .

Given a training set of positive images with labels $\{I_n, y_n\}$ and negative images not containing the object of interest, we define a structured prediction objective function similar to one proposed in [63]. The scoring function in (2.1) is linear in the parameters $w = \{\alpha, \beta, \gamma\}$, and therefore can be expressed as $S(I_n, y_n) = w \cdot \Phi(I_n, y_n)$. We learn a model of the form:

$$\begin{aligned} \operatorname{argmin}_{w, \xi_i \geq 0} \quad & \frac{1}{2} w^T \cdot w + C \sum_n \xi_n & (2.11) \\ \text{s.t.} \quad & \forall n \in \text{positive images} \quad w \cdot \Phi(I_n, y_n) \geq 1 - \xi_n \\ & \forall n \in \text{negative images}, \forall y \quad w \cdot \Phi(I_n, y) \leq -1 + \xi_n \end{aligned}$$

The above constraint states that positive examples should score better than 1 (the margin), while negative examples, for all configurations of part positions and mixtures, should score less than -1. We collect negative examples from images that does not contain any cars. This form of learning problem is known as a structural SVM, and there exist many well-tuned solvers such as the cutting plane solver of SVMStruct in [32] and the stochastic gradient descent solver in [18]. We use the dual

coordinate-descent QP solver of [63]. We show an example of a learned model and its learned tree structure in Figure 2.2.

2.4 3D Shape and Viewpoint

The previous section describes our 2D model of appearance and shape. We use it to propose detections with associated landmarks positions p^* . In this section, we describe a 3D shape and viewpoint model for refining p^* .

Consider 2D views of a single rigid object; 2D landmark positions must obey epipolar geometry constraints. In our case, we must account for within-class shape variation as well (e.g., sedans look different than station wagons). To do so, we make two simplifying assumptions: (1) We assume depth variation of our objects are small compared to the distance from the camera, which corresponds to a weak-perspective camera model. (2) We assume the 3D landmarks of all object instances can be written as linear combinations of a few basis shapes. Let us write the set of detected landmark positions as p^* as a $2 \times K$ matrix where $K = |V|$. We now describe a procedure for refining p^* to be consistent with these two assumptions:

$$\min_{R, \alpha} \|p^* - R \sum_i \alpha_i B_i\|^2 \quad \text{where } p \in \mathbb{R}^{2 \times K}, R \in \mathbb{R}^{2 \times 3}, RR^T = Id, B_i \in \mathbb{R}^{3 \times K} \quad (2.12)$$

Here, R is an orthonormal camera projection matrix and B_i is the i^{th} basis shape, and Id is the identity matrix. We factor out camera translations by working with mean-centered points p^* and let α directly model weak-perspective scalings.

Inference: Given 2D landmark locations p^* and a known set of 3D basis shapes B^i , inference corresponds to minimizing (2.12). For a single basis shape ($n_B = 1$), this problem is equivalent to the well-known “extrinsic orientation” problem of registering a 3D point cloud to a 2D point cloud with known correspondence [29]. Because the

squared error is linear in a_i and R , we solve for the coefficients and rotation with an iterative least-squares algorithm. We enforce the orthonormality of R with a nonlinear optimization, initialized by the least-squares solution [29]. This means that we can associate each detection with shape basis coefficients α_i (which allows us to reconstruct the 3D shape) and camera viewpoint R . One

could combine the reprojection error of (2.12) with our original scoring function from (2.1) into a single objective that jointly searches over all 2D and 3D unknowns. However inference would be exponential in K . We find a two-layer inference algorithm to be computationally efficient but still effective.

Learning: The above inference algorithm requires the morphable 3D basis B_i at test-time. One can estimate such a basis given training data with labeled 2D landmark positions by casting this as nonrigid structure from motion (SFM) problem. Stack all 2D landmarks from N training images into a $2N \times K$ matrix. In the noise-free case, this matrix is rank $3n_B$ (where n_B is the number of basis shapes), since each row can be written as a linear combination of the 3D coordinates of n_B basis shapes. This means that one can use rank constraints to learn a 3D morphable basis. We use the publicly-available nonrigid SFM code [55]. By slightly modifying it to estimate “motion” given a known “structure”, we can also use it to perform the previous projection step during inference.

Occlusion: A well-known limitation of SFM methods is their restricted success under heavy occlusion. Notably, our 2D appearance model provides location estimates for occluded landmarks. Many SFM methods (including [55]) can deal with limited occlusion through the use of low-rank constraints; essentially, one can still estimate low-rank approximations of matrices with some missing entries.

We can use this property to learn models from partially-labeled training sets. Recall that our learning formulation requires all landmarks (including occluded ones) to be labeled in training data. Manually labeling the positions of occluded landmarks

can be ambiguous. Instead, we use the estimated shape basis and camera viewpoints to infer/correct the locations of occluded landmarks.

2.5 Experiments

2.5.1 Datasets

In this chapter, we focus on car detection and 3D landmark estimation in cluttered, real-world datasets with severe occlusions. We labeled a subset of 500 images from the PASCAL VOC 2011 dataset [16] with locations and visibility states of 20 car landmarks. Our dataset contains 723 car instances. 36% of landmarks are not visible due to self-occlusion, while 21% of landmarks are not visible due to occlusion by another object (or truncation due to the image border). Hence *over half* our landmarks are occluded, making our dataset considerably more difficult than those typically used for landmark localization or 3D viewpoint estimation. We evenly split the images into a train/test set. We also compare results on a more standard viewpoint dataset from [1], which consists of 200 relatively “clean” cars from the PASCAL VOC 2007 dataset, marked with 40 discrete viewpoint class labels.

2.5.2 Implementation

We modify the publicly-available code of [63] and [55] to learn our models, setting the number of local mixtures $T = 9$, the number of global mixtures $M = 50$, and the number of basis shapes $n_B = 5$. We found results relatively robust to these settings. Learning our 2D deformable model takes roughly 4 hours, while learning our 3D shape model takes less than a minute. Our model is defined at a canonical scale, so we search over an image pyramid to find detections at multiple scales. Total run-time for a test image (including both 2D and 3D processing over all scales) is 10 seconds.

2.5.3 Evaluation

Given an image, our algorithm produces multiple detections, each with 3D landmark locations, visibility flags, and camera viewpoints. We qualitatively visualize such output in Figure 2.5.

To evaluate our output, we assume test images are marked with ground-truth cars, each annotated with ground-truth 2D landmarks and visibility flags. We measure the performance of our algorithm on four tasks. We evaluate **object detection (AP)** using the PASCAL criteria of Average Precision [16], defining a detection to be correct if its bounding box overlaps the ground truth by 50% or more. We evaluate **2D landmark localization (LP)** by counting the fraction of predicted landmarks that lie within $.5x$ pixels of the ground-truth, where x is the diameter of the associated ground-truth wheel. We evaluate **landmark visibility prediction (VP)** by counting the number of landmarks whose predicted visibility state matches the ground-truth, where landmarks may be “visible”, “self-occluded”, or “other-occluded”. Our 3D shape model refines only LP and VP, so AP is determined solely by our 2D (mixtures of trees) model. To avoid conflating the evaluation measures, we evaluate LP and VP assuming bounding-box correspondences between candidates and ground-truth instances are provided. Finally to evaluate **viewpoint classification (VC)**, we compare predicted camera viewpoints with ground-truth viewpoints on the standard benchmark of [1] which has fine viewpoint labels for 200 images from PASCAL 2007. To evaluate landmark localization/visibility and viewpoint, we assume correspondences between candidate detections and ground-truth instances are given.

2.5.4 Viewpoint Classification

We first present results for viewpoint classification in Figure 2.3 on the benchmark of [1]. Given a test instance, we run our detector, estimate the camera rotation R ,

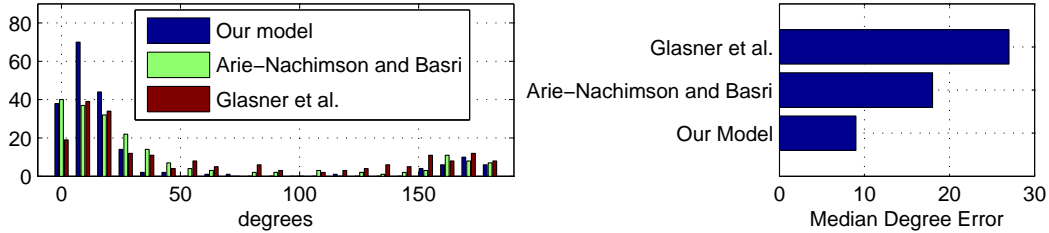


Figure 2.3: We report histograms of viewpoint label errors for the dataset of [1]. We compare to the reported performance of [1] and [22]. Our model reduces the median error (**right**) by a factor of 2.

and report the reconstructed 2D landmarks generated using the estimated R . Then we produce a quantized viewpoint label by matching the reconstructions to landmark locations for a reference image (provided in the dataset). We found this approach more reliable than directly matching 3D rotation matrices (for which metric distances are hard to define). We produce a median error of 9 degrees, a factor of 2 improvement over state-of-the-art. This suggests our model does accurately capture viewpoints. We next turn to a detailed analysis on our new cluttered dataset.

2.5.5 Baselines

We compare the performance of our overall system to several existing approaches for multi-view detection in Figure 2.4(a). We first compare to widely-used latent deformable part model (DPM) of [18], trained on the exact same data as our model.

A supervised DPM (MV-star) considerably improves performance from 63 to 74% AP, where supervision is provided for (view-specific) root mixtures and part locations. This latter model is equivalent in structure to a state-of-the-art model for car detection and viewpoint estimation [44], which trains a DPM using supervision provided by a 3D CAD model. By allowing for tree-structured relations in each view-specific global mixture (MV-tree), we see a small drop in AP = 72.3%. This model is equivalent to a state-of-the-art model for view-based face detection and pose estimation [68].

Our final model is similar in term of detection performance (AP = 72.5%), but

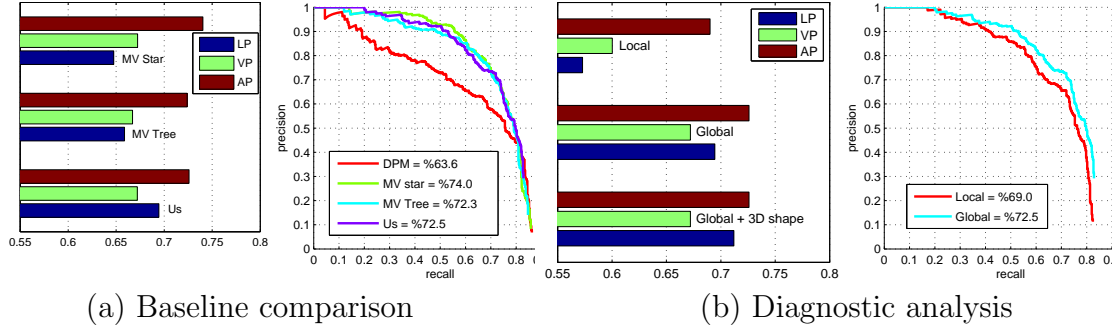


Figure 2.4: We compare our model with various view-based baselines in (a), and examine various components of our model through a diagnostic analysis in (b). We refer the reader to the text for a detailed analysis, but our model outperforms many state-of-the-art view-based baselines based on trees, stars, and latent parts. We also find that modeling the effects of shape due to global changes in 3D viewpoint is crucial for both detection and landmark localization.

does noticeably better than both view-based models for landmark prediction. We correctly localize landmarks 69.5% of time, while MV-tree and MV-star score 65.7% and 64.7%, respectively. We produce landmark visibility (VP) estimates from our multi-view baselines by predicting a fixed set of visibility labels conditioned on the view-based mixture. We should note that accurate landmark localization is crucial for estimating the 3D shape of the detected instance. We attribute our improvement to the fact that our model can model a large number of global viewpoints by composing together different local view-based templates.

2.5.6 Diagnostics

We compare various aspects of our model in Figure 2.4(b). “Local” refers to a single tree model with local mixtures only, while “Global” refers to our global mixtures of trees. We see a small improvement in terms of AP, from 69% for “Local” to 72.5% for “Global”. However, in terms of landmark prediction, “Global” strongly outperforms “Local”, 69.4% to 57.2%. We use these predicted landmarks to estimate 3D shape below.

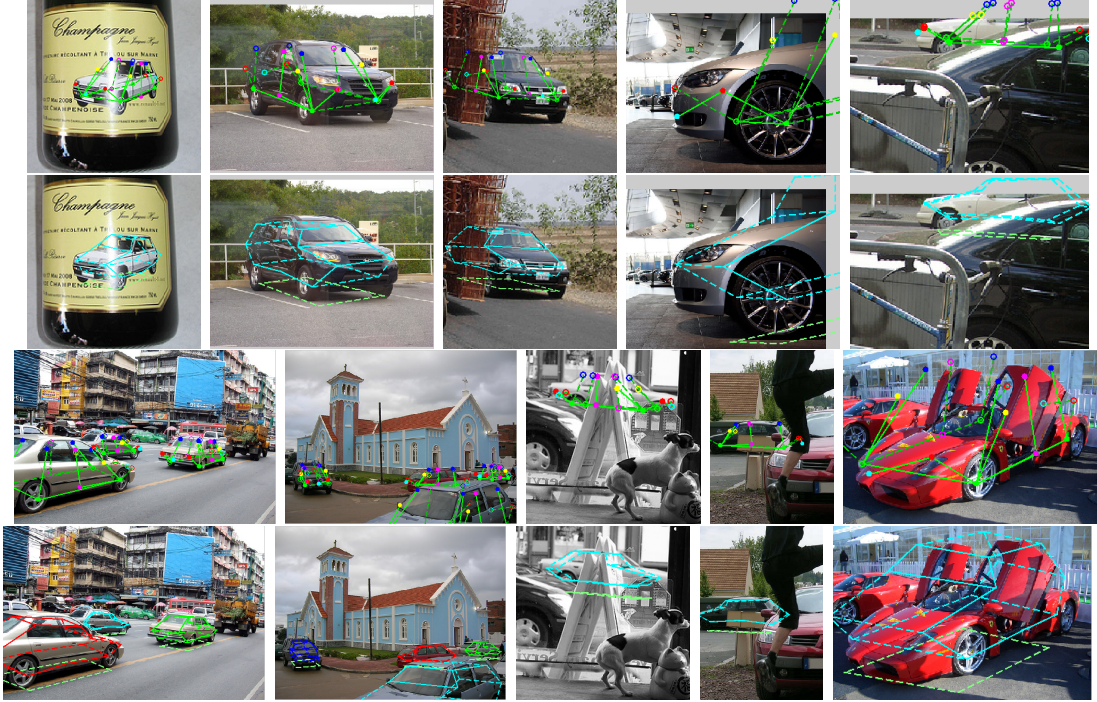


Figure 2.5: Sample results of our system on real images with heavy clutter and occlusion. We show pairs of images corresponding to detections that matched to ground-truth annotations. The top image (in the pair) shows the output of our tree model, and the bottom shows our 3D shape reconstruction, following the notational conventions of Figure 2.2. Our system estimates 3D shapes of multiple cars under heavy clutter and occlusions, even in cases where more than 50% of a car is occluded. Our morphable 3D model adapts to the shape of the car, producing different reconstructions for SUVs and sedans (row 2, columns 2-3). Recall that our tree model explicitly reasons about changes in visibility due to self-occlusions versus occlusions from other objects, manifested as local mixture templates. This allow our 3D reconstructions to model occlusions due to other objects (e.g., the rear of the car in row 2, column 3). In some cases, the estimated 3D shape is misaligned due to extreme shape variation of the car instance (e.g., the folding doors on the lower-right).

2.5.7 3D Shape

Our 3D shape model reports back a z depth value for each landmark (x, y) position. Unfortunately, depth is hard to evaluate without ground-truth 3D annotations. Instead, we evaluate the improvement in re-projected VP and LP due to our 3D shape model; we see a small 2% improvement in LP accuracy, from 69.4% to 71.2%. We

further analyze this by looking at the improvement in localization accuracy of ground-truth landmarks that are visible (73.3 to 74.8%), self-occluded (70.5 to 72.5%), and other-occluded (22.5 to 23.4%). We see the largest improvement for occluded parts, which makes intuitive sense. Local templates corresponding to occluded mixtures will be less accurate, and so will benefit more from a 3D shape model. Our 3D model accurately estimates self-occlusion, but cannot reason directly about occlusions due to other objects in the scene.

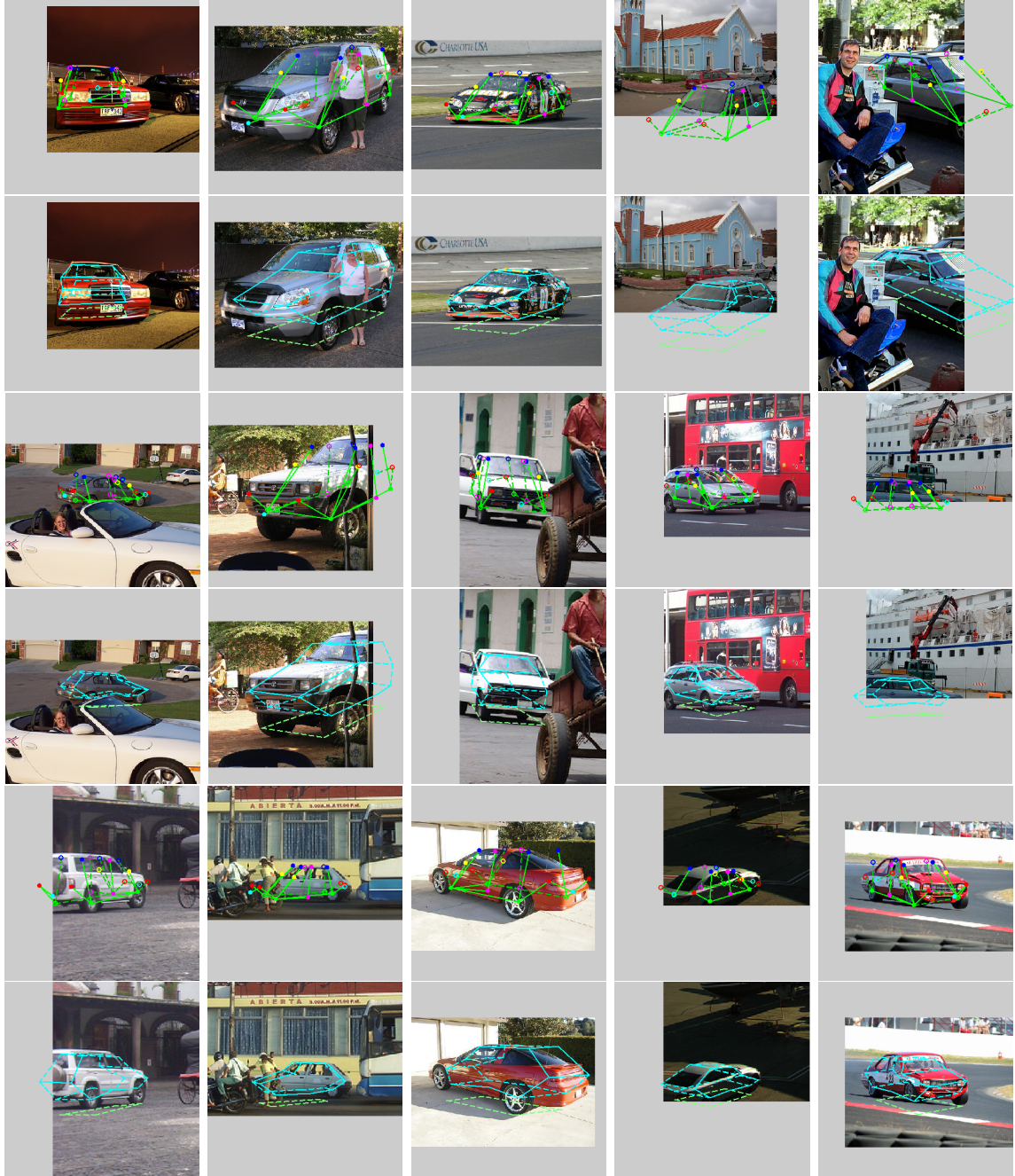


Figure 2.6: Sample results of our system on real images with heavy clutter and occlusion. We show pairs of image corresponding to a detection that matched to a ground-truth annotation. The top image (in the pair) shows the output of our tree model, and the bottom shows our 3D shape reconstruction, following the notational conventions of Figure 2.5. Our system estimates 3D shapes of multiple cars under heavy clutter and occlusions, even in cases where more than 50% of a car is occluded.

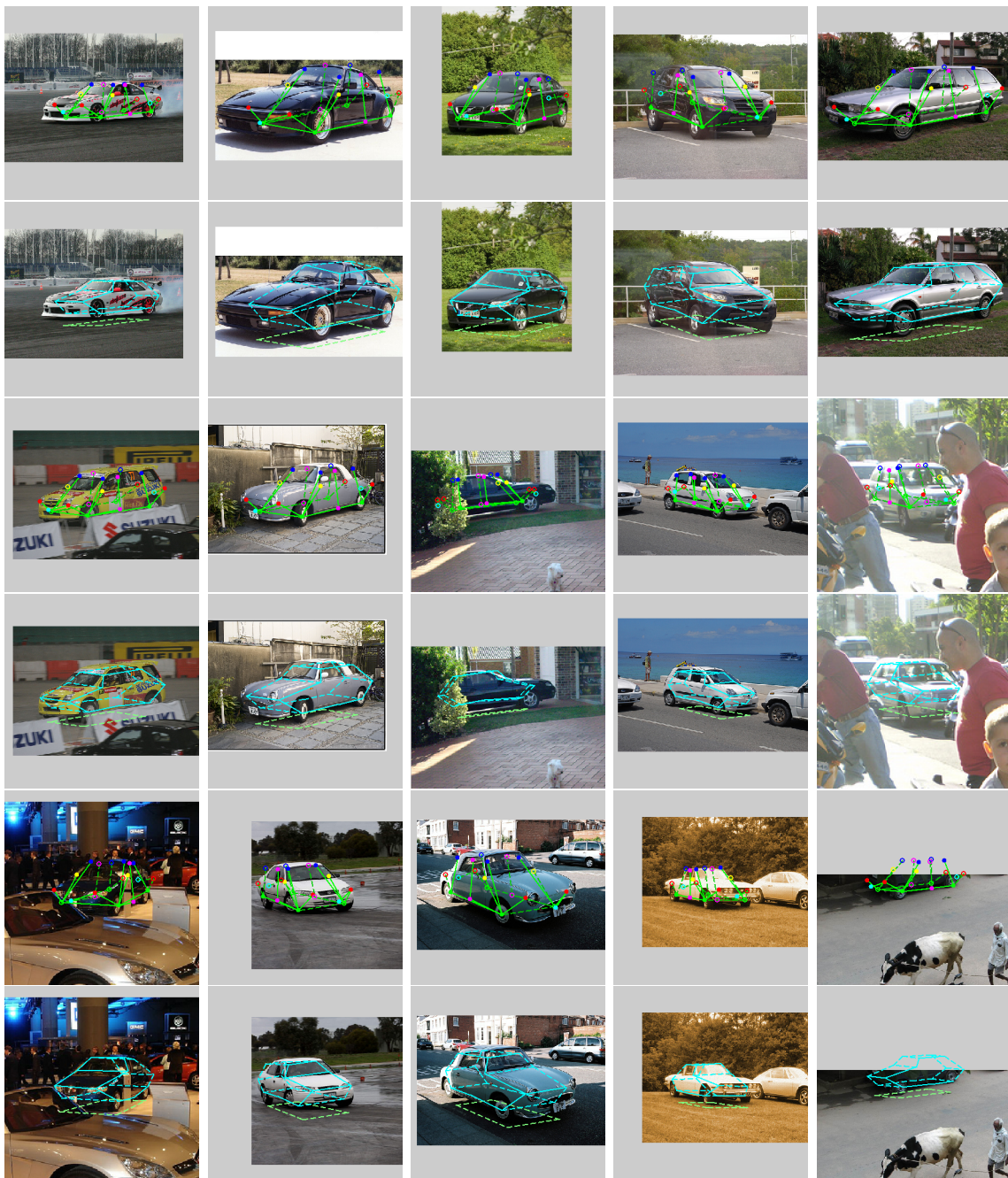


Figure 2.7: Sample results of our system on real images with heavy clutter and occlusion. We show pairs of image corresponding to a detection that matched to a ground-truth annotation. The top image (in the pair) shows the output of our tree model, and the bottom shows our 3D shape reconstruction, following the notational conventions of Figure 2.5. Our system estimates 3D shapes of multiple cars under heavy clutter and occlusions, even in cases where more than 50% of a car is occluded.

2.6 Conclusion

We have described a geometric model for detecting and estimating the 3D shape of objects in heavily cluttered, occluded, real-world images. Our model differs from typical multi-view approaches by reasoning about local changes in landmark appearance and global changes in visibility and shape, through the aid of a morphable 3D model. While our model is similar to prior work in terms of detection performance, it produces significantly better estimates of 2D/3D landmarks and camera positions, and quantifiably improves localization of occluded landmarks. Though we have focused on the application of analyzing cars, we believe our method could apply to other geometrically-constrained objects.

CHAPTER III

3D Recognition with 3D Synthesis

“ Evolve solutions; when you find a good one, don’t stop. ”

David Eagleman

3.1 Introduction

In Chapter 2, we described a two stage model to recognize and reconstruct 3D objects. In this chapter we describe a single model that simultaneously detects instances of general object categories, and reports a detailed 3D reconstruction of each instance. The proposed approach is based on an *analysis by synthesis* strategy. A forward synthesis model constructs possible geometric interpretations of the world, and then selects the interpretation that best agrees with the measured visual evidence.

The forward model synthesizes visual templates defined on invariant (HOG) features. These visual templates are discriminatively trained to be accurate for inverse estimation. We introduce an efficient “brute-force” approach to inference that searches through a large number of candidate reconstructions, returning the optimal one. One benefit of such an approach is that recognition is inherently (re)constructive. We show state of the art performance for detection and reconstruction on two challenging 3D object recognition datasets of cars and cuboids.

Challenges: Though attractive, an “inverse rendering” approach to computer vision is wildly challenging for two primary reasons. (1) It is difficult to build accurate

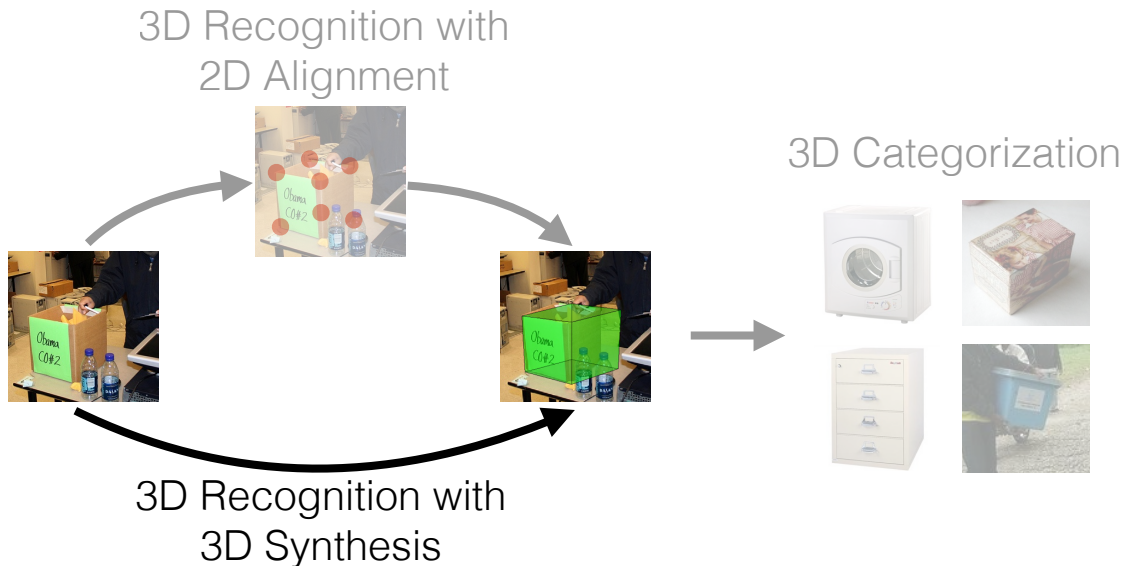


Figure 3.1: Overview

generative models that capture the full complexity of the visual world. (2) Even given such a model, inverting it is difficult because the problem is fundamentally ill-posed (different reconstructions may generate similar images) and full of local minima (implying local search will fail).

Our approach: Our approach addresses both difficulties. (1) Instead of generating pixel values, we use forward models to synthesize visual templates defined on invariant (HOG) features. These visual templates are discriminatively trained to be accurate for inverse estimation. (2) We describe a “brute-force” approach to inference that efficiently searches through a large number of candidate reconstructions, returning the optimal one (or multiple likely candidates, if desired).

Latent-variable object models: Our model is related to approaches that recognize objects with latent variable models, such as the state-of-the-art deformable part model (DPM) [18]. [69] point out that DPMs implicitly synthesize a set of deformed templates by searching over possible latent values. The deformation set is limited to obey sparse 2D spring constraints, making the search amenable to dynamic programming. In contrast, we *explicitly* synthesize a massive set of templates by enumerating

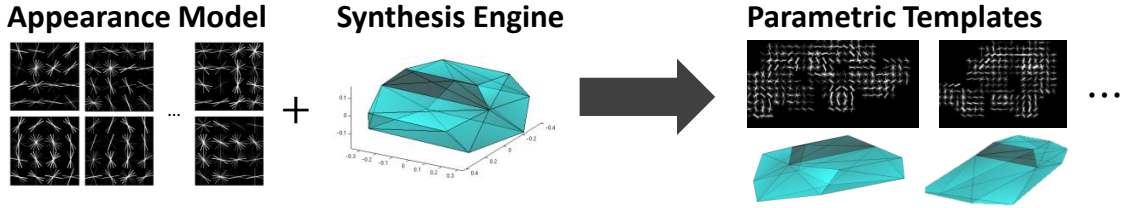


Figure 3.2: We describe a method for synthesizing a large set of discriminative templates, each associated with a candidate 3D reconstruction of an object (in this case, cars). Our model makes use of a generative 3D shape model to synthesize a large collection of 2D landmarks, which in turn specify rules for composing 2D templates out of a common pool of parts.

over latent parameters in an arbitrarily-complex forward model (that explicitly constructs 3D objects and cameras). We perform a brute-force search through these (re)constructions. Surprisingly, by making use of part indexing, our search can be even faster than a DPM.

Our model produces state-of-the-art benchmark performance for detection and reconstruction of cuboids in indoor images [62] and cars from the PASCAL dataset (introduced in Chapter 2). We also present a diagnostic analysis that shows that, in some cases, our synthesis model is close to optimal (given our feature space).

We introduce our shape synthesis model in section 3.3 and specify the forward rendering process for generating 2D templates in section 3.4. We then describe our algorithms for inference in section 3.5 and learning in section 3.6. We conclude with experimental results in section 3.7.

3.2 Related Work

3D categorical models: Many approaches represent object categories using local features and their geometric arrangement in a 3D coordinate system [47, 50, 70]. Most related to us are approaches based on view-based part models [41, 24, 44, 18]. In particular, [44] learn view-based car models making use of a geometric CAD model

to generate synthetic training images. Instead of synthesizing images, we synthesize feature templates (an easier task). We synthesize templates that are detailed enough to perform 3D reconstruction, while this may be difficult for view-based approaches (since many views may be needed).

Geometric indexing: Historically, many model-based recognition systems proceeded by aligning 3D models to image data. Typically, a sparse set of local features are initially detected, after which alignment is treated as a feature correspondence problem. Efficient correspondence search is implemented through affine/projective invariant indexing [19], geometric hashing [37], or simple enumeration [22].

Hough transforms use sparse feature detections to vote in a shape parameter space [2], resulting in 2D implicit shape models [1, 53]. Our part indexing scheme is similar in spirit, except that we use a dense set of part responses to cast votes for a discrete set of candidate 3D reconstructions.

Model synthesis: Synthetic parametric models of object-categories is an active area of research in the graphics community. Approaches include procedural grammars [42], morphable basis shapes [5], and component/part-based models [34].

Similar to Chapter 2, we make use of morphable models to represent categorical shape variation. We show that such basis models can be learned from 2D annotations using techniques adapted from nonrigid structure from motion (SFM) [57, 55]. The proposed models differ from past work in that we use a geometric model to synthesize a large set of exemplars, which are then used for “brute-force” matching. From this perspective, our approach is similar to work that relies on a non-parametric model of object shape [3].

3.3 Synthesis model

Let’s begin by defining a parametric model for constructing 3D shapes from a particular object category (such as cars). We will then sample from this parametric

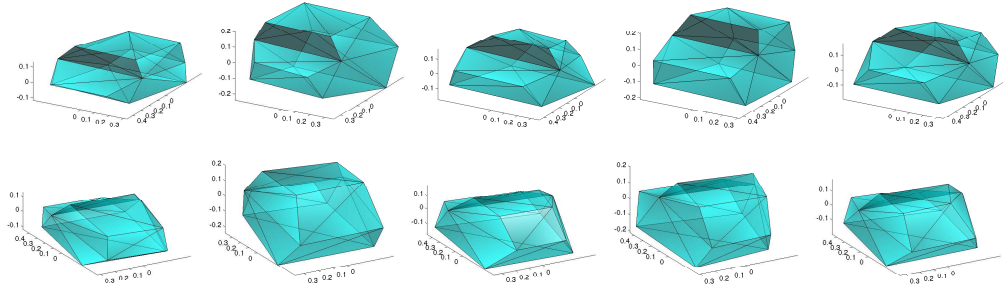


Figure 3.3: We use basis shapes to model different types of cars, like sedans in the first column and SUVs in the fourth. Since the [55] assumes orthographic camera, another major aspect of learned shape basis is modeling perspective effect. For example the second and third column are modeling back and front view perspective effect.

family to define a large set of candidate 3D reconstructions. Our 3D shape model should capture nonrigid shape variation within an object category (sedans and SUVs look different) and viewpoint variation.

Similar to Chapter 2, we make use of morphable basis models [5] that model any 3D shape instance as a linear combination of 3D basis shapes. Our shape model should also encode changes in appearance due to geometric variation (wheels look different when foreshortened). To do so, we learn separate local templates for landmarks conditioned on their 3D geometry (learning separate templates for frontal vs. foreshortened wheels).

Shape parameters: We represent the 3D shape of an object with a set of N 3D keypoints, represented as $B \in \mathbb{R}^{3 \times N}$. Given a set of n_B basis shapes $\{B_j\}$ and coefficients α , we synthesize a 3D shape B as follows:

$$B = B_0 + \sum_{j>0}^{n_B} \alpha_j B_j, \quad \text{where} \quad B, B_j \in \mathbb{R}^{3 \times N} \quad (3.1)$$

where B_0 is the mean 3D shape. We visualize our shape basis in Figure 3.3.

Camera parameters: We transform B into camera coordinates by rotating by

R and translating by t

$$P = t + RB, \quad t \in \mathbb{R}^3, R \in \mathbb{R}^{3 \times 3} \quad (3.2)$$

When augmented with camera intrinsic parameters (the focal length), the set of camera parameters are (t, R, f) . We now summarize our parameters with a vector θ :

$$\begin{aligned} \theta &= \begin{bmatrix} \textit{Shape} & \textit{Camera} \end{bmatrix} \\ \textit{Shape} &= \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_k \end{bmatrix} \\ \textit{Camera} &= \begin{bmatrix} t & R & f \end{bmatrix} \end{aligned} \quad (3.3)$$

Forward projection: Given a parameter vector, we generate a set of 2D keypoints, scales, and local mixtures with the following:

$$\textit{Render}(\theta) = \{(z_i, m_i) : i = 1 \dots N\} \quad (3.4)$$

$$z_i = (x_i, y_i, \sigma_i) = \left(f \frac{p_x^i}{p_z^i}, f \frac{p_y^i}{p_z^i}, \frac{f}{p_z^i} \right) \quad (3.5)$$

where (3.5) is a standard perspective projection model, and p^i is the i^{th} column of matrix P . We have assumed unity-scaled pixels factors for simplicity (though they can easily be added).

Appearance synthesis: To capture changes in appearance caused by geometry (frontal and foreshortened wheels look different), we associate each keypoint with a discrete mixture m_i . We will use mixture-dependent local templates $\beta_i^{m_i}$ to capture such appearance variability. We now describe a simple approach for synthesizing m_i conditioned on P (the view-dependent 3D geometry). Let us define $rel_i(P)$ to be a

vector of relative 3D landmark locations:

$$rel_i(P) = \{p_j - p_i : j \in N(i)\}. \quad (3.6)$$

where $N(i)$ is the set of keypoints connected to i under the 3D mesh model. We use the 3 other keypoints with highest spatial correlation to i . Offline, we extract the set of $\{rel_i\}$ from the set of synthesized shapes, and cluster them using k-means. Given this clustering, we now can synthesize mixture labels m_i by finding the closest geometric mean:

$$m_i = k^* \quad \text{where} \quad k^* = \underset{k \in M}{\operatorname{argmin}} \ ||rel_i(P) - \mu_k^i||^2 \quad \text{and} \quad rel = \{p_j - p_i : j \in N(i)\}. \quad (3.7)$$

where μ_k^i is defined as the average relative location of neighboring points $N(i)$ for cluster k .

We show 3D geometric means μ_k^i and their associated appearance-specific visual templates β_i^k in Figure 3.4.

Parameter quantization: We explore various strategies for producing a set of parameters θ . One option is to use the set of parameters encountered in a set of training images. Alternatively, we can synthesize a set of parameters θ with a grid search over a bounded range of parameters (where bounds on the camera rotation matrix is defined in terms of elevation and azimuth Euler angles). In either case, we clamp camera translations to be 0 ($t = 0$) to ensure translation-invariance. We do search over focal lengths f to model perspective effects during synthesis. This produces a massive set of thousands or millions of parameters vectors, produced by enumerating over a training set or a grid search. In our results, we experiment with

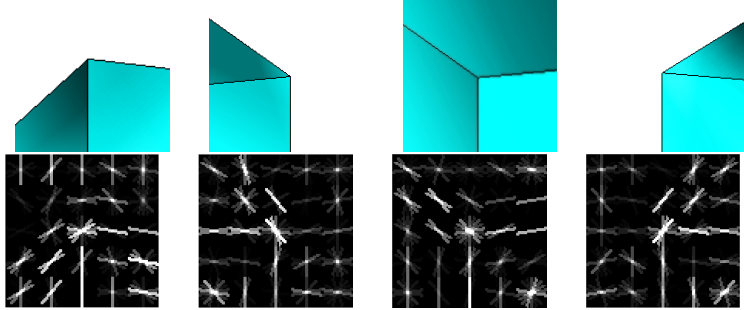


Figure 3.4: We learn local part mixtures by clustering the relative 3D position of keypoint i and its connected neighbors in the underlying 3D mesh. We show keypoint cluster means μ_k^i (**above**), along with their associated part templates β_i^k (**below**). Each synthesized 3D pose (and associated template) is constructed by adding together shifted copies of local part templates, which in turn allows for efficient run-time search.

various quantized subsets. We wish to quantize together parameters that yield similar 2D projections. Specifically, we construct a vector of 2D (x_i, y_i) keypoint positions for each discrete θ , and cluster this set with K -means. We denote the final set of K -quantized parameter vectors as

$$\Omega_K = \{\theta_1 \dots \theta_K\} \quad (3.8)$$

3.4 Template model

Given a parameter vector θ and image I , we describe a method for scoring a visual template w_θ :

$$S(I, \theta) = \sum_{u \in U} w_\theta[u] \cdot I[u] \quad (3.9)$$

where $I[u]$ is an image feature extracted from a pixel location and scale $u = (x, y, \sigma)$ in image I . We write U for the set of all possible discrete pixel locations and scales

enumerated in a feature pyramid. In practice, w_θ is a single-scale template with local spatial support. For notational simplicity, we assume that templates are zero-padded (across space and scales).

To efficiently represent our family of templates, we construct each template w_θ by adding together local keypoint templates shifted to lie at locations given by $Render(\theta)$ (3.5). We write $\beta_i^{m_i}$ for the (zero-padded) local visual template, or “part”, associated with keypoint i , when tuned for mixture m_i :

$$S(I, \theta) = \sum_{i=1}^N \sum_{u \in U} \beta_i^{m_i}[u] \cdot I[u + z_i] \quad (3.10)$$

where we drop the dependence of the rendered keypoint location $z_i = z_i(\theta)$ and mixture $m_i = m_i(\theta)$ on parameter θ . If keypoint i is occluded given the viewpoint specified by θ , then the associated m_i acts as an occlusion-specific mixture. In such cases, the learned template $\beta_i^{m_i}$ may be set to all zeros, or it may capture image features characteristic of occlusions (such as t-junctions).

Let us define a dummy indexing variable $u' = u + z_i$ and switch the order of summations in the above equation. This allows us to write the global template w_θ from (3.9) as a superposition of shifted keypoint templates:

$$w_\theta[u] = \sum_{i=1}^N \beta_i^{m_i}[u - z_i] \quad (3.11)$$

where we have assumed keypoint templates β are zero-padded outside of their default spatial extent.

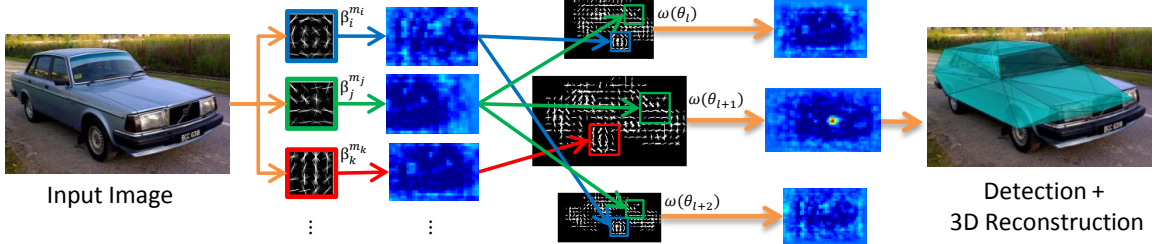


Figure 3.5: We search through a large collection of templates (with shared parts) by first caching part responses, and then looking up response values to score each template.

3.5 Inference

Inference corresponds to computing

$$\max_{\theta \in \Omega_K} S(I, \theta) \quad (3.12)$$

To simplify notation, we assume that the translation-invariant set of parameters $\theta \in \Omega_K$ are augmented with camera translations at run-time. This allows the above maximization to perform a scanning-window search over image translations and scales.

To efficiently search over scores for all $\theta \in \Omega_K$ given an image I , we first pre-compute a response map of keypoint template responses for each location u :

$$R_i^{m_i}[u] = \sum_{u' \in U} \beta_i^{m_i}[u'] \cdot I[u' + u] \quad (3.13)$$

We pre-compute the above response map for each keypoint i and mixture m_i by convolving the feature pyramid I with the part template $\beta_i^{m_i}$.

We now can define the score associated with a particular object parameter by

looking up values in the cached response maps:

$$S(I, \theta) = \sum_{i=1}^N R_i^{m_i}[z_i(\theta)] \quad (3.14)$$

The final inference algorithm, visualized in Figure 3.5, is as follows.

1. Offline, enumerate parameters $\theta \in \Omega_K$ and cache the associated set of rendered keypoints $Render(\theta)$.
2. Online, given an image, compute the response map for all N parts and M mixtures (3.13).
3. Evaluate $S(I, \theta)$ for each $\theta \in \Omega_K$ with N (≈ 10) table lookups (3.14).
4. Return parameters θ above a detection threshold, along with their associated reconstructions B (3.1).

3.6 Learning

Our models require two sets of parameters; those associated with shape synthesis θ , and those associated with local keypoint templates $\beta_i^{m_i}$. We learn both using training images annotated with 2D keypoint locations.

Synthesis parameters: In some cases, one can use a graphics engine or CAD models to directly synthesize a set of 3D parameter vectors θ . We can also infer such 3D parameters from 2D keypoint annotations so as to minimize 2D reprojection error. Similar to Chapter 2, we employ nonrigid structure from motion (SFM) [55] to learn a 3D basis. Stack all 2D keypoints from N training images into a $2N \times K$ matrix. In the noise-free case, this matrix is rank $3n_B$ (where n_B is the number of basis shapes), since each row can be written as a linear combination of the 3D coordinates of n_B basis shapes. This means that one can use rank constraints to learn a 3D morphable

basis. We learn local appearance clusters μ by clustering view-dependent 3D shapes obtained from the set of $\{\theta_i\}$.

Template parameters: We will learn templates that are discriminatively tuned for accurate detection and reconstruction on single-view training images. Assume we are given supervised training data including positives $\{I_i, \theta_i\}$ and negatives $\{I_i\}$. Oftentimes supervision is more naturally specified in terms of 2D keypoint annotations rather than 3D shapes. In such a scenario, we use the nonrigid SFM procedure from the previous paragraph to estimate shape parameters θ given 2D keypoint annotations. Combining (3.9) with (3.11), we can explicitly denote the score as linear in keypoint templates $\beta = \{\beta_i^{m_i}\}$:

$$S(I, \theta) = \beta \cdot \Phi(I, \theta) \tag{3.15}$$

We will learn templates that minimize the following training objective function:

$$\min_{\beta} \frac{1}{2} \|\beta\|^2 + C \sum_i \xi_i \tag{3.16}$$

$$\text{s.t. } \forall i \in \text{pos}, \quad \beta \cdot \Phi(I_i, \theta_i) \geq 1 - \xi_i \tag{3.17}$$

$$\forall i \in \text{neg}, \forall \theta \in \Omega_K, \quad \beta \cdot \Phi(I_i, \theta) \leq -1 + \xi_i \tag{3.18}$$

The above constraint states that positive examples should score better than 1 (the margin), while negative examples, for all configurations of keypoints positions and mixtures defined by Ω_K , should score less than -1. Violations of these constraints are penalized through slack terms. We find margin violations on negative images (not containing the object) by running the efficient inference algorithm from section 3.5 to find detections that score above -1. This form of learning is known as a structural

SVM, and there exist many well-tuned solvers such as SVMStruct [32] and stochastic gradient descent [18]. We use a stochastic dual coordinate-descent implementation based on [63].

Scalability: Training time scales roughly linearly with K (the number of synthesized shapes). This holds true because β is independent of K , while hard-negative mining scales linearly with K since each shape must be enumerated. For large K , we found that one could speed up training times by stochastically sub-sampling shapes during hard-negative mining without sacrificing accuracy. We sub-sampled a fixed number (50) regardless of K , making training time practically independent of K .

Our model on average takes a few hours to train on a commodity PC using full train set of 1196 positives and 1375 negatives. We show an example of a learned model in Figure 3.2.

Recognition vs reconstruction: The above constraints naturally corresponds to detection accuracy. One could augment them to ensure that, for a positive example I_i , the true shape θ_i outscores incorrect shapes $\theta \neq \theta_i$ by some amount. This corresponds to a structured prediction task that explicitly trains parameters so as to generate accurate shapes. We found that these additional constraints did not improve performance given a large enough negative set (we use a generic set of 1000 outdoor images).

Data scarcity: Interestingly, the above formulation learns accurate models even with a small number of positives that are dwarfed by the the number of templates $|pos| \ll |\Omega_K|$. It may seem strange that *we are learning templates for shapes that have never been seen* - but this is precisely the benefit of synthesis! We learn good templates so long as there exist enough positives to train the local parts $\beta_i^{m_i}$. Given this fact, one might be tempted to simply train the local parts independently, but the above structured formulation takes advantage of contextual interactions between all parts, defined by the *entire* set of parameters Ω_K . Because “hard negative” margin

violations are produced by searching across all templates in Ω_K , the learning algorithm above will tend to produce a strong set of models $\{w_\theta : \theta \in \Omega_K\}$.

3.7 Results

Datasets: We evaluate the proposed model using two object detection datasets. The SUN primitive dataset [62] that contains 785 images with 1269 annotated cuboids. The UCI-Car dataset that is proposed in the Chapter 2 and contains 500 images from PASCAL VOC2011 containing 723 cars with detailed landmark annotation. For both datasets, we use the same train-test split provided by the curators for training and evaluation.

3.7.1 Evaluation

Our models report back object detections with associated 3D reconstructions. Because annotating images with 3D shapes is cumbersome, we evaluate our reconstructions by evaluating 2D landmark re-projection error. This allows us to use standard benchmarks and compare to past work. For object detection we use now-standard average precision measure introduced in PASCAL. For landmark localization, we follow [62] and plot landmark accuracy for various levels of object-detection recall. A predicted landmark is defined to correct if it lies within t pixels of the ground-truth location, where $t = 15\%$ of the square root of the area of the ground-truth box.

3.7.2 Baselines

We compare to previously published results as well as the approach introduced in Chapter 2 for both datasets. In particular, we use state-of-the DPMS as baseline for object detection [18], and supervised tree-based part models proposed in Chapter 2 as baselines for landmark prediction. In Chapter 2 we used a two-stage inference procedure for reconstruction, where detections from a 2D tree-based part model are

refined to produce 3D reconstructions, in contrast, the proposed model performs both detection and reconstruction in a *single* stage.

3.7.3 Implementation

For our car models, we set the number of basis shapes $n_b = 5$. We learn a model with $N = 20$ 3D landmarks, each modeled with $|M| = 9$ local mixtures. For our cuboid models, we manually define a 3D parametric cuboid model of varying aspect ratios. Our model consists of $N = 17$ 3D landmarks (consisting of cube corners and midpoints), with $|M| = 12$ local mixtures.

3.7.4 Synthesis strategies

We explored numerous strategies for constructing a set of 3D shape parameters $\{\theta_i\}$. First, our *Exemplar* model uses the shapes encountered in the training set of annotated images, augmented with synthetic camera translations.

Exemplar Synthesis augments this set with additional exemplar shapes. We implement this strategy by learning a model with a subset of training images, but using the larger (full) set of keypoint annotations. This mimics scenarios where we have access to a limited amount of image data, but a larger set of keypoint annotations.

Parametric Synthesis constructs a shape set by discretely enumerating θ_i over bounded parameter ranges. Finally, *Oracle Synthesis* uses shapes extracted from annotated test-data. We use this upper bound on performance (given the the “perfect” synthesis strategy) for additional analysis. Note that exemplar-synthesis is a valid, implementable strategy (that even outperforms parametric synthesis, given enough shape exemplars) while oracle synthesis is an upper-bound that is used purely for analysis.

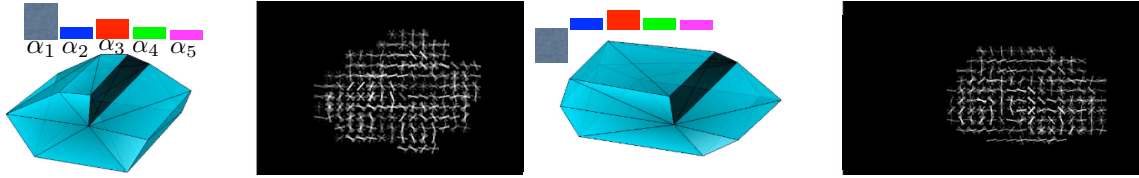


Figure 3.6: A visualization of our interactive, morphable interface for exploring 3D shapes and their associated templates. We display the corresponding shape coefficients α as colored bars.

3.7.5 Interactive synthesis

We have implemented an interface for interactive synthesis (Figure 3.6). A common tool for visualizing morphable models is an interface where a user can dynamically toggle/slide shape coefficients, and view the resulting model. We have constructed such an interface, and can use it to visualize our family of 3D shapes, camera viewpoints, and associated HOG templates. We find it to be an intuitive user experience for “understanding” the modeling capacity of our representation.

3.7.6 Anytime recognition/reconstruction

Our models have a free parameter K , the number of enumerated shapes. Both performance and run-time computation increase with K . When comparing to baselines with fixed run-time costs, we plot performance as a function of run-time, measured in terms of seconds per image. All methods are run on the same physical system (a 12-core Intel 3.5 Ghz processor). Recall that we obtain shape sets for smaller K by clustering a larger set of shapes. Our plots reveal that a simple *re-ordering* of shapes in a coarse-to-fine fashion (with hierarchical clustering) can be used for *any-time* analysis. For example, after enumerating the first $K = 20$ coarse shapes, one can still obtain 65% car landmark reconstruction accuracy (which in turns improves as more shapes are enumerated).

3.7.7 Box benchmark results

Figure 3.7 plots performance for box detection and localization. Exemplars almost *double* the best previously-reported numbers in [62], in terms of detection (43% vs 24%) and landmark reconstruction (48% vs 38%).

Interestingly, the tree model proposed in Chapter 2 outperforms [62], perhaps due to its modeling of local part mixtures. The proposed models even surpass the two-stage model of Chapter 2, while directly reporting 3D reconstructions and while being 10X faster. Exemplar and Parametric Synthesis perform similarly for low numbers of templates, but Exemplars do better with more templates, particularly with respect to reconstruction accuracy. Moreover, both methods still fall short of the upper-bound given by Oracle Synthesis. These results suggests that our parametric model is not capturing true shape statistics. For example, people may take pictures of certain objects from iconic viewpoints. Such dependencies are not modeled by Parametric Synthesis, but are captured by Exemplars. We later demonstrate (Figure 3.8) that Exemplar Synthesis also captures such dependencies.

3.7.8 Car benchmark results

We find similar trends when evaluating detection and landmark accuracy for cars. The proposed models fall just shy of the tree model of Chapter 2, but directly report 3D reconstructions while being 5X faster. As before, Exemplars dominate Parametric Synthesis for any fixed number of templates. But Parametric Synthesis can potentially outperform Exemplars with additional shapes (because Exemplar is limited to observed training data). Moreover, our upper-bound analysis reveals that both models are close to the upper bound provided by Oracle Synthesis. This suggests that our morphable 3D model is a rather accurate description of car shapes.

3.7.9 Diagnostic analysis

We present qualitative results for both boxes and cars in Figure 3.9. In Figure 3.8, we present further diagnostic analysis of our box model with respect to training data size. When training on small amounts of training data, Exemplar Synthesis noticeably improves performance by 5%. To realize this improvement, it is important to discriminatively-train the full synthetic set of templates. These results suggest that accurate shape statistics are crucial to realize the benefit of synthesis. Indeed, we show that one can produce a state-of-the-art model with as little as 20 training images.

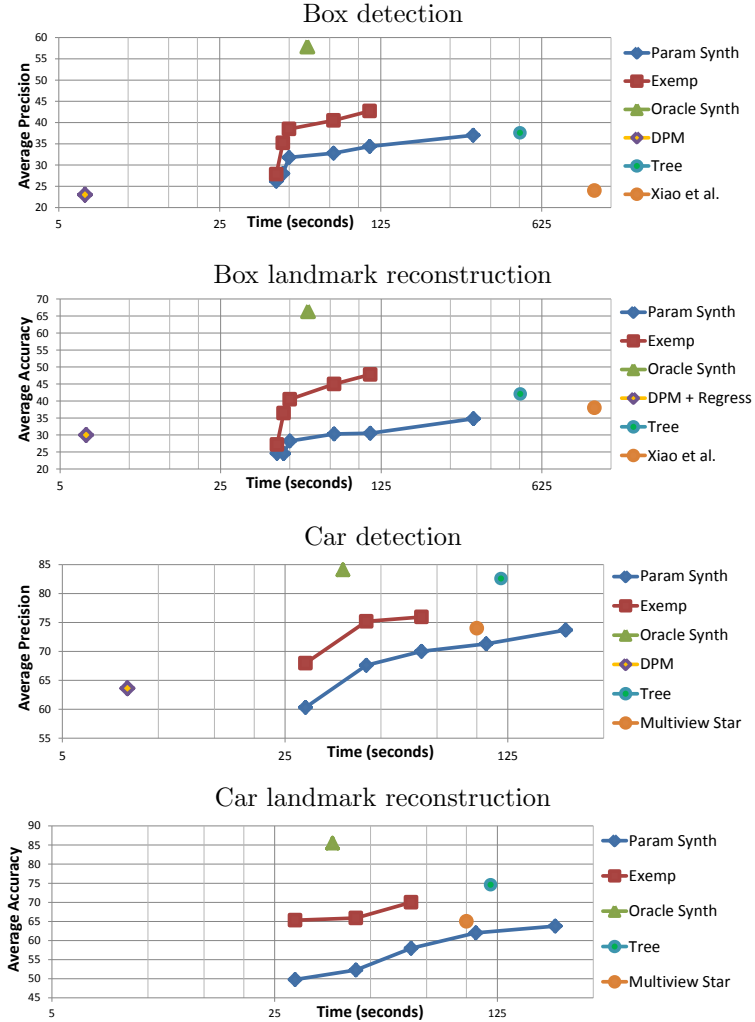


Figure 3.7: Detection (**left**) and reconstruction accuracy (**right**) versus running time of our method and baselines, including DPMs [18], supervised-tree models and multi-view star models (introduced in Chapter 2). Points correspond to different (constant-time) baselines, while curves correspond to our models. Because our models can process a variable number of synthesized templates, we sweep over $K \in \{20, 50, 100, 500, 1000, 4000\}$ templates to generate the curves. Note that Exemplars are limited by the number of training images. Exemplars always dominate Parametric Synthesis (for a given K), suggesting our parametric model is failing to capture important shape statistics. We examine this further in Figure 3.8. Our box (**top**) detection and reconstruction results (43% and 48%) nearly double the best previously-reported performance from Xiao et al.[62] (24% and 38%), while being 10X faster. Our car (**bottom**) results approach the state-of-the-art tree models proposed in Chapter 2, but directly report 3D shape while being 5X faster. One can also use cascade models and or context to reduce the number of evaluated synthesized templates, thus spend less time for detection.

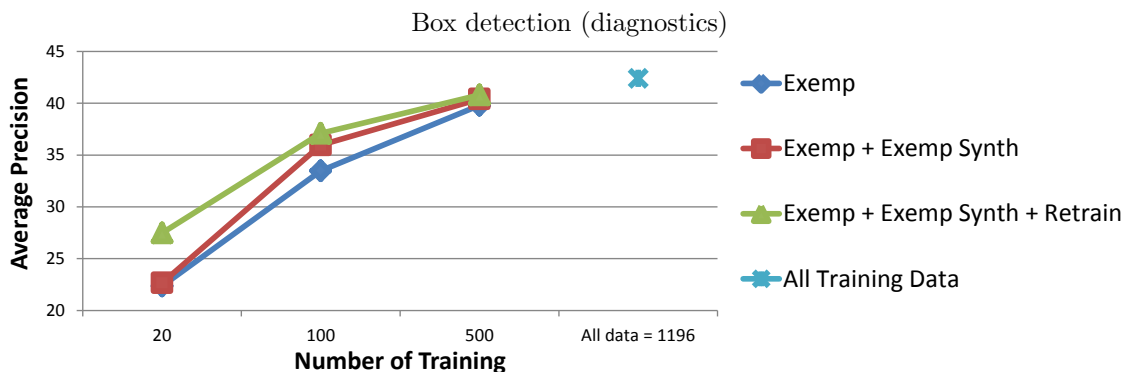


Figure 3.8: We plot the performance of various synthesis approaches as a function of the amount of training images. **Exemp** enumerates the set of shapes encountered in the training set of images. **+Exemp Synth** uses the learned local templates β from **Exemp** and instantiates new shapes obtained from keypoint annotations not in the training set. This improves performance by up to 2%. **+Retrain** discriminatively retrains β given this synthesized set of shapes, further improving performance by up to 5%. Hence it is crucial to discriminatively-tune the synthesized set. Our synthesis models outperform state-of-the-art methods [62] with orders-of-magnitude less training data.

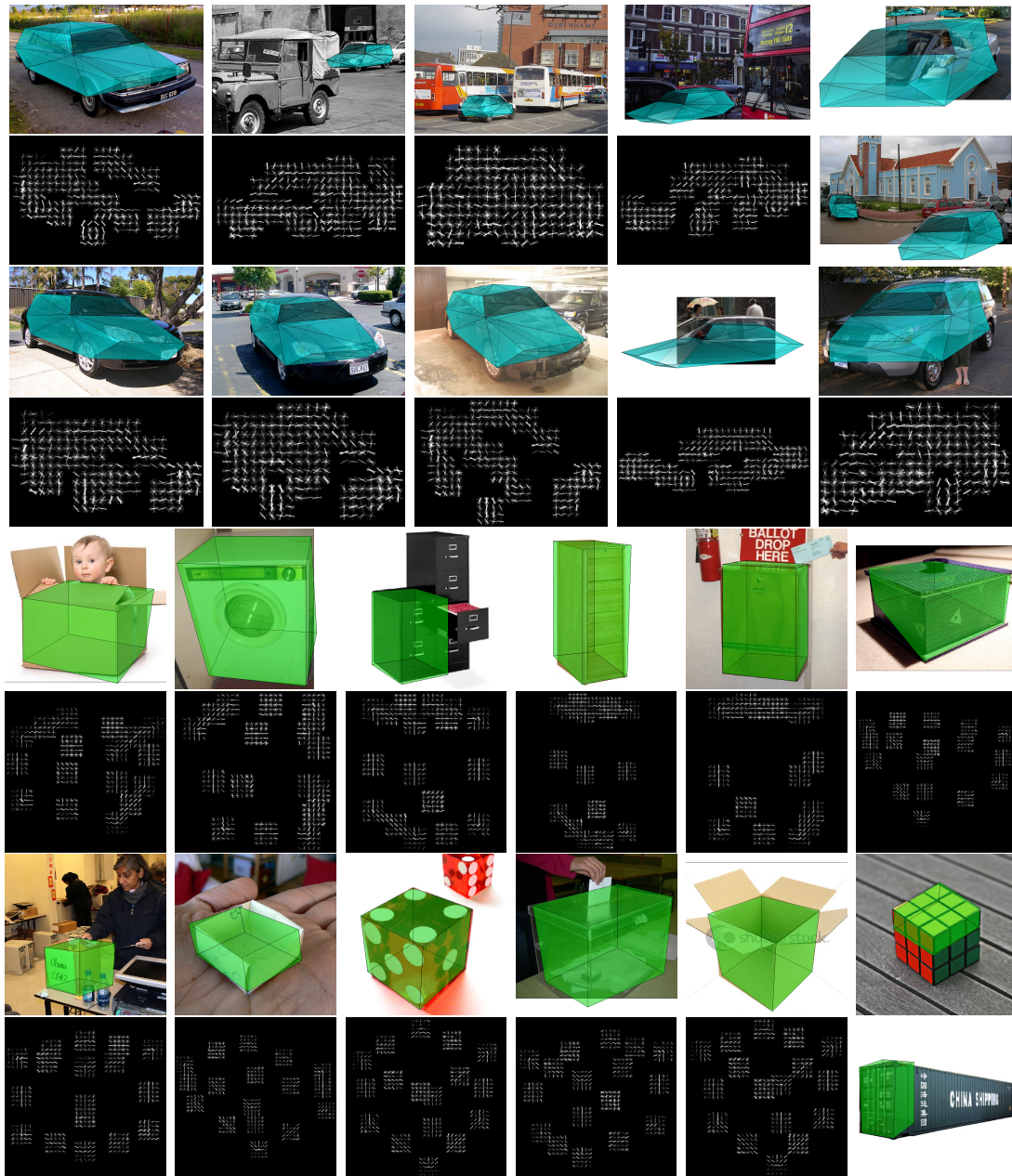


Figure 3.9: Recognition + reconstructions from our method. Odd rows show the test image and recognized + reconstructed object overlaid on it. Even rows illustrate the associated template that triggered the detection. Our method can recognize objects from various viewpoints, shapes and is robust to heavy occlusion. Because every synthesized template has a 3D shape, recognition is inherently reconstruction. On the top right, we show results for images with multiple cars. Our box results show accurate reconstructions across various viewpoints, aspect ratios, and even perspective effects. However, some images are genuinely ambiguous, like the Rubik’s Cube (bottom-right) or the shape is very extreme and our synthesis engine never synthesized that shape, like the container on the last row.

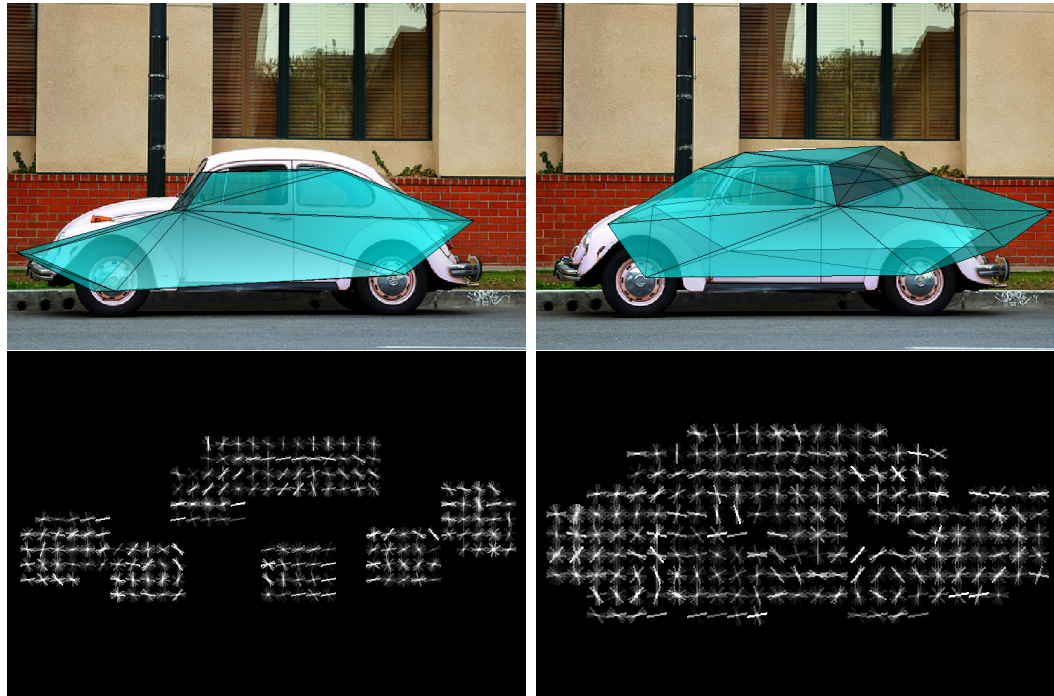


Figure 3.10: We show an example detection for which the reconstruction problem is fundamentally ill-posed (in our HOG feature space). Our brute-force strategy for enumerating all reconstructions can readily return multiple high-scoring interpretations, addressing a classic limitation of “inverse rendering” approaches.

3.8 Conclusions

We have introduced a new approach for recognizing and reconstructing 3D objects in images based on an analysis by synthesis strategy. We make use of forward synthesis models to synthesize a large number of possible geometric interpretations, and efficiently search through this set with indexing schemes. Generative shape models, while common in graphics, have been somewhat absent in recent computer vision techniques based on discriminative classifiers. Our methods discriminatively train a large set of synthetic geometric models, such that they are accurate for both recognition and reconstruction. Constructing this set from an observed collections of exemplar shapes does remarkably well, but one can still improve on these results with accurate shape-driven synthesis.

Our upper-bound analysis suggests that there is much room to improve shape statistics, and such statistics will be crucial for accurate generative synthesis.

CHAPTER IV

3D Categorization

“ *Quotation*, n: The act of repeating erroneously the words of another. ”

Ambrose Bierce

4.1 Introduction

In Chapter 2 and 3 we described how to recognize 3D objects in single 2D images. In this chapter, we address the natural question of how to use these geometric-reasoning engines for *categorical recognition*, focusing on cuboidal object categories such as washing machines, cabinets, etc. We evaluate both categorization and 3D shape estimation using a variety of representations capturing both appearance and geometry. Cuboidal objects are interesting since they share the same basic shape, allowing one to explicitly explore the interplay of geometry and appearance.

Pose-normalization: Perhaps the most natural approach would use the estimated alignment to extract *pose-normalized* appearance features. For a cuboidal object, one might represent the appearance of each cuboidal face in a fronto-parallel view (Figure 4.2). Many state-of-the-art systems for recognition (such as faces [52, 35], cars [36], animal species [8, 17], or general attributes [65]) similarly normalize landmarks/keypoints into a canonical coordinate frame during training and or testing. For example, the vast majority of face recognition systems work by detecting landmarks, warping the image such that landmarks are aligned into a canonical frontal

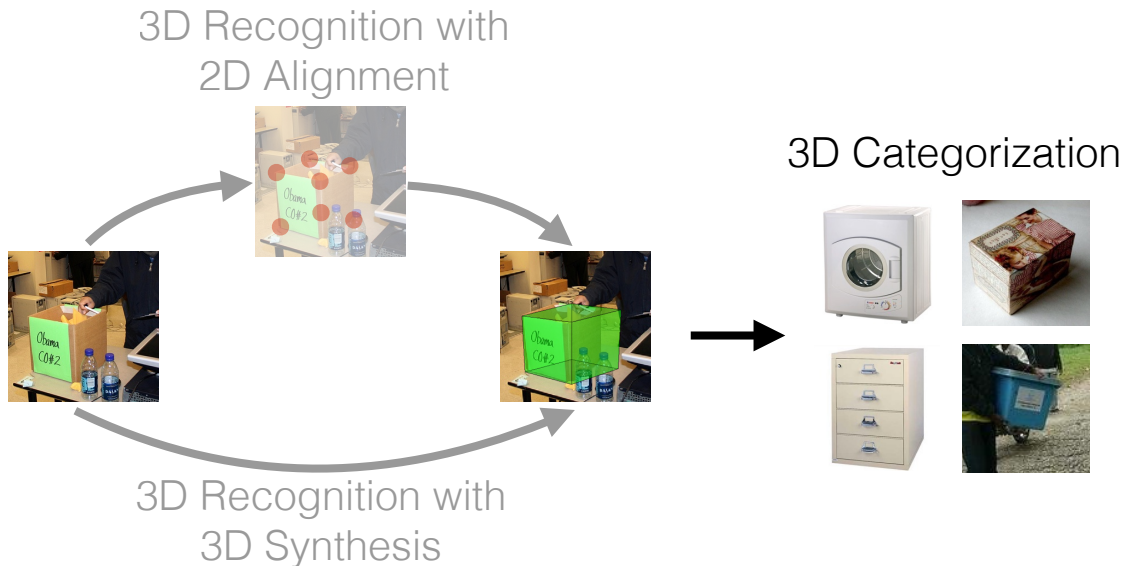


Figure 4.1: Overview

view, and classifying the warped (pose-normalized) appearance [66, 31]. Importantly, normalization allows one to (1) factor out “nuisance” variables such as viewpoint and aspect/shape during recognition, and (2) generalize to poses not seen in training data.

Pose-retargeting: First and foremost, we demonstrate that pose-normalization is *not* the optimal strategy for dealing with appearance variation due to pose. One explanation maybe the inaccuracy of current systems for pose estimation - small misalignments in the predicted pose may cause large errors in the pose-normalized appearance. Surprisingly, we show that *even with ground-truth alignment on test images*, pose-normalization is still not optimal. In short, pose-normalization (a) removes geometric cues that maybe helpful for recognition (washing machines may have differing aspect ratio from microwaves) and (b) artificially re-weights foreshortened regions of the objects. To address these limitations, we describe an approach that warps (or *retargets*) training examples to the shape and viewpoint of a particular detected instance, and performs recognition using this retargeted training set.

Pose-synthesis: We demonstrate that pose-retargeting is the optimal approach given ground-truth alignment, but falls short given the accuracy of current systems

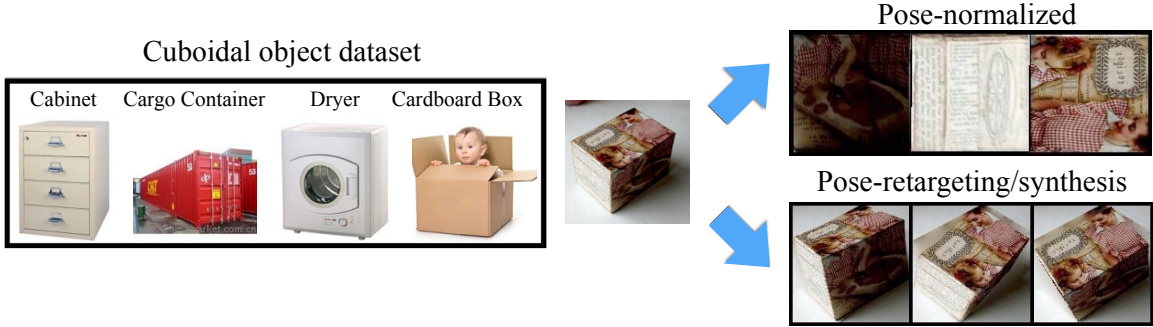


Figure 4.2: We examine 3D shape categorization of cuboidal objects (**left**). Such objects share similar shape, so conventional folk wisdom might advocate the use of shape-invariant (or pose-normalized) representations for recognition (**top**) that are attractive because they (1) factor out shape (which seems uninformative when classifying objects with similar shape) and (2) can generalize to novel shapes not encountered in training data. We show that this approach is not optimal. One reason is that current methods produce small errors in geometric alignment, which can result in large fluctuations in the pose-normalized appearance. However, even with ground-truth alignment, pose-normalization is still not optimal. We demonstrate that pose-synthesis (**bottom**), a simple approach of augmenting training data with geometrically perturbed training samples, is a surprisingly effective strategy that allows for state-of-the-art categorization and automatic 3D alignment.

that estimate cuboidal alignments. To address this limitation, we evaluate another approach that *pre-synthesizes* a large set of possible target poses. The synthesized set is used to train a practical system that jointly performs categorization and 3D alignment, at a level of accuracy that surpasses the current state-of-the-art. We evaluate systems based on exemplar matching and discriminative template-matching. Intuitively, geometric alignment systems (that recognize generic cuboids) must be invariant to variation across cuboidal categories, but our system can exploit the fact that washing machines and microwaves look different. Importantly, synthesis also allows our system to generalize to unseen viewpoints and shapes not seen in the training set *without* requiring pose-normalization.

Data-augmentation: Our proposed approaches are inspired by learning architectures that apply synthetic perturbations to training data. Such “data-augmentation”

appears to be crucial components of state-of-the-art methods like deep learning [45, 38]. However, instead of applying simple perturbations like rotations, we make use of an image-based rendering engine to generating new training images (using piecewise-constant homographies and affine transformations). With a rich enough synthesis engine, the resulting learning algorithm does not need to generalize to unseen test poses (because they can be directly synthesized). Indeed, we show that highly-invariant appearance features based on contemporary CNNs [49] do not outperform traditional gradient orientation histograms [11] when used with large synthetic training sets.

Our contributions: We compare, both theoretically and empirically, different representations for the novel task of categorizing cuboidal objects. We begin with a baseline pose-“agnostic” approach (that trains a categorical classifier agnostic to the pose of training/test data). We compare such a method with pose-normalization, pose-synthesis, and pose-retargeting (which to our knowledge, is novel).

We provide two salient conclusions: (1) The novel problem of categorical cuboid classification provides an interesting testbed for solving a practical task while investigating the role of shape and geometry. To spur future research in this area, we re-purpose an existing dataset [62] (designed for for cuboidal detection and alignment) for the task of cuboidal category recognition by making use of category labels and adding 3D landmark annotations. (2) Pose-retargeting, both at test and train-time (through synthesis), provides a simple approach for dealing with geometric variation that significantly outperforms the common-place technique of pose-normalization.

4.2 Image-based rendering

The core computational engine of all our studied approaches is an image-based renderer that takes an $H \times W$ input image I , a set of N 2D landmarks P , and produces a warped image with a retargeted set of N landmarks T . We write this engine as a function that returns an image:

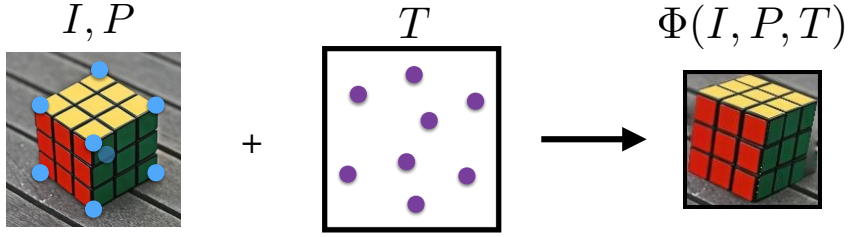


Figure 4.3: Proposed image-based rendering engine takes an image I , a set of 2D landmarks P and a set of target 2D landmark locations T as input, then it renders the cube in target view by warping each surface using a homography warp.

$$\Phi(I, P, T), \quad I \in \mathcal{R}^{H \times W \times 3}, \quad P, T \in \mathcal{R}^{2 \times N} \quad (4.1)$$

Importantly, landmarks and their associated faces are assumed to have a semantic ordering. For example, the first face is the front of the washing machine, while the second is the top, etc. The warped image Φ is synthesized in three stages: foreground synthesis, hidden surface synthesis, and background synthesis.

Foreground: By triangulating the points, one could generate a retargeted image by applying affine warps to each triangle. Instead, we assume that the points will always be corners of a cuboidal object, and so can be connected onto quadrilateral faces instead of triangles. Our rendering engine applies a homography (which can be estimated by the 4 corners of a quadrilateral) to each quadrilateral face.

Hidden-surfaces: If target pose is very different from the input pose, then previously occluded cube faces may now become visible. We assume that objects are symmetric in appearance, and use the texture map from the opposite face as a replacement. One can even synthesize multiple target images by choosing different visible surfaces as a replacement. We found that choosing one replacement performs well.

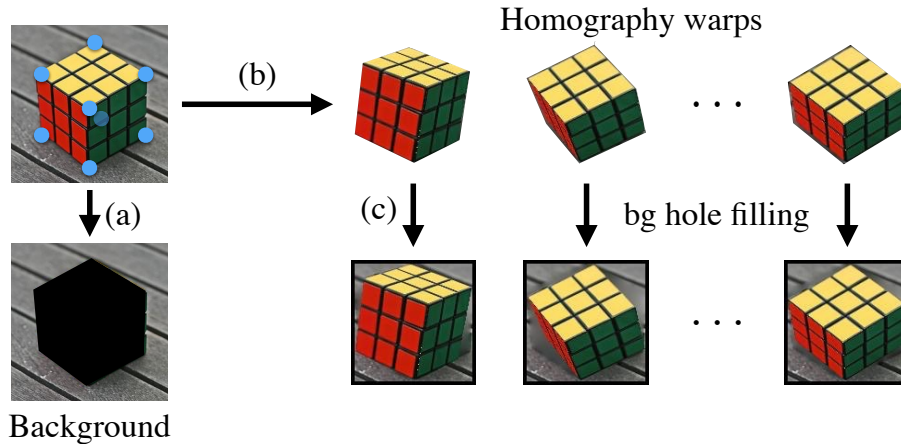


Figure 4.4: The image-based rendering engine works in three steps: (a) extracting background from the image. (b) rendering object in different poses using homography warps on object surfaces. (c) pasting the warped object on the background and filling the the holes using interpolation.

Background: We only warp pixels inside the input object to the target pose, leaving background pixels intact. This will result in holes in the background of the target image. We use standard hole-filling algorithms [10]. Background provides useful contextual cues for object categorization. It is even possible to synthesize images using different background images of the same object category but in our experiment we only used the input image background.

Our synthesis engine is fairly straightforward, similar in complexity to a typical homework assignment in an undergraduate computer vision course! [51]. Nevertheless, it produces startlingly photo-realistic images of cuboidal objects (Figure 4.2). We use it to explore a variety of representations for geometric-invariant recognition.

4.3 Approaches

In this section, we describe a simple mathematical formalism for unifying all the geometric representations that we will consider. Throughout this section, we visualize the function Φ as a warped image, but to simplify notation, we assume that Φ directly extracts a N -dimensional feature vector extracted from the warped image. We will

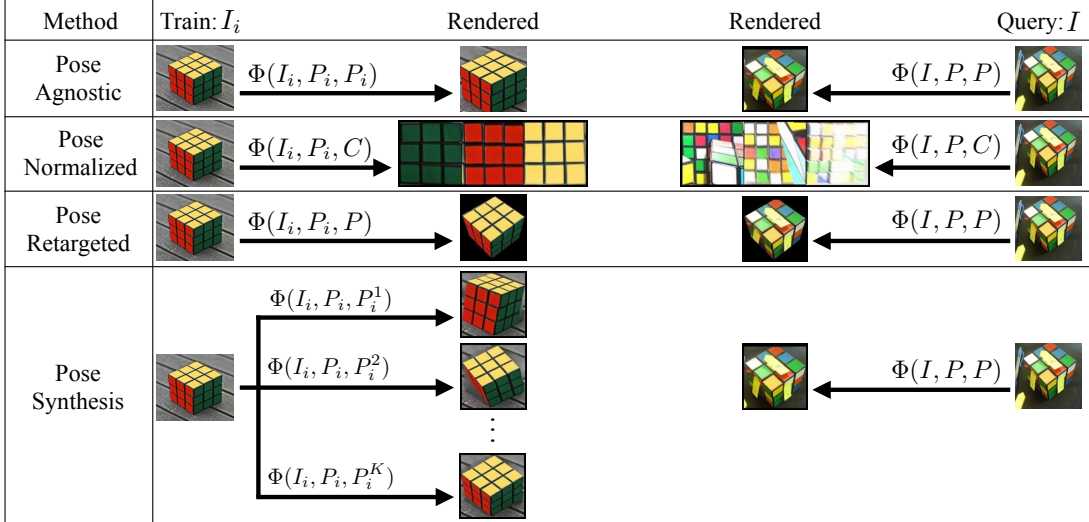


Figure 4.5: We visualize various strategies for achieving geometric-invariance in recognition. Please refer to the text for a detailed description of each strategy.

explore oriented gradient features and deep features. Though these features will be used in a variety of classification engines – including nearest neighbor (NN) matching and SVMs – we write out the mathematical formulation for a simpler NN classifier below. We assume that we have a training set of *real* images where the i^{th} image is associated with a category label y_i and ground-truth landmarks P_i . We also assume that we have a *real* query image at test-time I , with an associated set of test-time landmarks P :

$$\text{Training images: } \{(I_i, y_i, P_i)\} \quad \text{Test image: } (I, P) \quad (4.2)$$

Test-time landmarks are provided by an state-of-the-art method proposed in Chapter 3, though we crucially also consider ground-truth landmarks.

Finally, we also consider representations that do not require 2D landmarks. To denote such methods, we use the notation of $\Phi(I, P, P) = \Phi(I, \cdot, \cdot)$ to specify an identity warp (where in fact, the set of points P need not be specified).

4.3.1 Pose-Agnostic

The simplest approach would be to simply ignore pose as an explicit confounding factor, and match using features extracted from un-warped images:

$$\text{Pose-Agnostic}(I) = y_{i^*} \quad \text{where} \quad i^* = \underset{i}{\operatorname{argmin}} \|\Phi(I_i, P_i, P_i) - \Phi(I, P, P)\|^2 \quad (4.3)$$

$$= \underset{i}{\operatorname{argmin}} \|\Phi(I_i, \cdot, \cdot) - \Phi(I, \cdot, \cdot)\|^2 \quad (4.4)$$

where we use the second line to emphasize the fact the identity warp computed by Φ did not require the knowledge of any landmarks. The first line will be useful for comparison with other approaches that do make use of landmarks. Such agnostic approaches can still be successful if the training set of images spans enough variations in pose. However, such methods fundamentally cannot generalize to novel poses at test-time, unless a highly invariant feature descriptor is used (e.g., bag-of-word representations [64]), in which case discriminability may suffer.

4.3.2 Pose-Normalized

Pose-normalization warps both the training and query images into a canonical configuration of N landmarks C :

$$\text{Pose-Normalized}(I) = y_{i^*} \quad \text{where} \quad i^* = \underset{i}{\operatorname{argmin}} \|\Phi(I_i, P_i, C) - \Phi(I, P, C)\|^2 \quad (4.5)$$

We visualize our canonical configuration of cuboid face landmarks in Figure 4.6.

Pose-Normalized explicitly factors out viewpoint and shape (which is helpful if they serve as nuisance variables for categorization), and trivially generalizes to novel

viewpoints and shapes not seen in the training set. Our experiments will show that normalization performs well when given highly accurate alignments. Small errors in the estimated pose may cause large distortions in the normalized image (Figure 4.17). However, even given ground-truth landmarks, pose-normalization may still artificially re-weight foreshortening portions of the object, sometimes resulting in image distortion due to pixelation artifacts (Figure 4.6).

It is important to note that there are many valid and plausible strategies to formulate a pose-normalization representation. The choice of canonical pose C can impact the performance of the representation. Another strategy is to use multiple canonical poses. We focus on a single canonical pose which unfolds the cube into three squares. Analyzing other pose-normalization strategies is the subject of further research.

4.3.3 Pose-Retargeted

Instead of warping each training image to a canonical landmark configuration C , pose-retargeting warps each training image to the target landmarks P found on a query image I :

$$\mathbf{Pose-Retargeted}(I) = y_{i^*} \quad \text{where} \quad i^* = \underset{i}{\operatorname{argmin}} \|\Phi(I_i, P_i, P) - \Phi(I, P, P)\|^2 \quad (4.6)$$

In some sense, pose-retargeting creates a custom-training set for this particular query image by warping the training set into the viewpoint and shape of the query. This tends to produce less distortions because warps are applied to training images (which tend to be cleaner and more accurately labeled landmarks) rather than a test-image. However, retargeting still requires accurate alignment landmarks at test-time, and more-over, may be slower since it requires generating a custom training set for

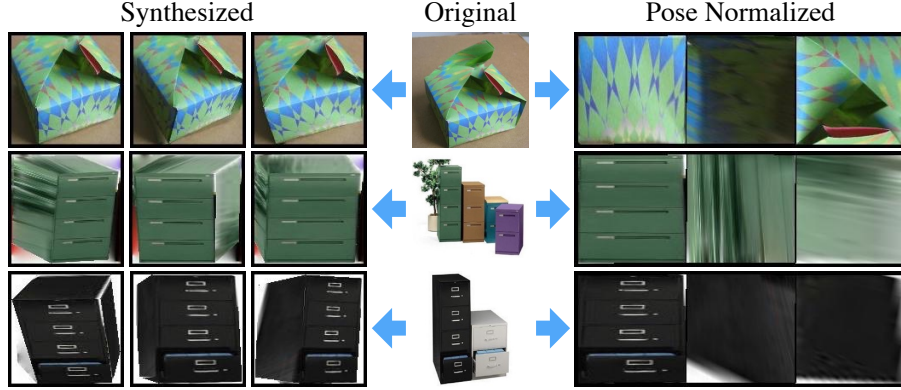


Figure 4.6: Given an image of an object, we show novel synthesized views generated by Pose-Synthesis (**left**) and the normalized view used by Pose-Normalized (**right**). Synthesized views look fairly realistic (because we can explicitly control and limit the degree of view synthesis), while the normalized views often have pixelation artifacts. The artifacts can arise from extrapolation of heavily-foreshortened cube faces (**middle** row) or small mistakes in the predicted 2D landmarks (**bottom** row).

each query.

4.3.4 Pose-Synthesis

Finally, we also consider an alternative that pre-warps all training images to set of possible target shapes and viewpoints. Pose-synthesis representation is based on augmenting the train set with new views of the training images:

$$\text{Pose-Synthesis}(I) = y_{i^*} \quad \text{where} \quad i^* = \underset{i}{\operatorname{argmin}} \min_{P_i^k \in G(P_i)} \|\Phi(I_i, P_i, P_i^k) - \Phi(I, P, P)\|^2 \quad (4.7)$$

$$(i^*, P_i^{k^*}) = \underset{i, P_i^k \in G(P_i)}{\operatorname{argmin}} \|\Phi(I_i, P_i, P_i^k) - \Phi(I, \cdot, \cdot)\|^2 \quad (4.8)$$

where $G(P_i) = \{P_i^1, \dots, P_i^K\}$ generates a set of candidate target landmarks. We refer to this function as a *landmark-synthesis* engine, described further below. We rewrite

the matching function as (4.8) to emphasize that (1) Pose-Synthesis does not require landmarks to be annotated on test images and (2) Pose-Synthesis can also be used for 3D landmark prediction (P_i^{k*}).

Landmark synthesis: Landmark synthesis is used to generate a set of reasonable target landmarks for each training image I_i , to be used by Pose-Synthesis. We assume that each training image is labeled with 3D landmarks (in camera coordinates) and a focal length f . Specifically, 2D landmarks P are assumed to be perspective projection of the 3D points:

$$P_i = \text{Project}(S_i, f_i), \quad \text{where } P_i \in \mathcal{R}^{2 \times N}, S_i \in \mathcal{R}^{3 \times N}, f_i \in \mathcal{R} \quad (4.9)$$

We use nonrigid structure-from-motion [56] to infer 3D landmarks (and affine camera parameters) from 2D annotations. We use these estimates to then infer a perspective camera calibration to produce f_i . Given 3D shape and camera parameters, we generate rotations along the camera x and z axis:

$$G(P_i) = \{\text{Project}(R_k S_i, f_i)_{k=1}^K\}, \quad R_k \in \mathcal{R}^{3 \times 3}, R_k^T R_k = I \quad (4.10)$$

Other synthesis strategies: We explored other synthesis strategies for Pose-Synthesis. For example, one could generate aspect ratio variations in the set of shapes. Moreover, one could extend the notion of data augmentation into the appearance domain as well as shape. For example, we could synthesize hidden cube surfaces or backgrounds by swapping out surfaces and backgrounds from other training examples. Our experiments focus on viewpoint synthesis, but our encouraging results suggest that other synthesis techniques are worth exploring.

4.3.5 Theoretical Analysis

We now provide the theoretical motivation for Pose-Retargeted and Pose-Synthesis. Consider a generative model of an image as

$$Pr(Image, Pose) = Pr(Pose)Pr(Image|Pose) \quad (4.11)$$

$$= Pr(Pose)\mathcal{N}(Image; \Phi(I, C, Pose), \sigma^2 Id) \quad \text{for } \mathcal{N}(x; \mu, \Sigma) \quad (4.12)$$

where I is image of the cuboidal faces of a training image in a canonical view C and Id is the identity matrix.

It is straightforward to show that Pose-Synthesis matches an image using the log-probability of that image under (4.12), (max) marginalizing over an uninformative pose prior:

$$I^*, Pose^* = \underset{I \in train, Pose}{\operatorname{argmax}} Pr(Image, Pose) \quad (4.13)$$

The category label is defined by label of I^* available in training and estimated pose is $Pose^*$. One strategy to impose weak prior on the pose is to search over perturbations of the training data. In our experiments, we uniformly sample $R_k = R_z R_x$ by ranging R_x and R_z over increments of $(-15, 0, 15)$ degrees.

Pose-Retargeted uses the log-probability of $P(Image|Pose)$, conditioning on the known pose.

$$I^* = \underset{I \in train}{\operatorname{argmax}} Pr(Image|Pose) \quad (4.14)$$

Intuitively, both Pose-Retargeted and Pose-Synthesis score an image reconstruction error that searches over candidate poses or conditioned on a known pose.

4.4 Experimental Results

Dataset: We re-purpose the SUN Primitive dataset [62], containing 1269 cuboid objects (annotated with 2D corners) in 785 images. SUN Primitive was proposed to study cuboid detection and 2D alignment. We use a subset of 543 cuboids that have category labels, spanning a set of 9 categories (Figure 4.7). To apply our synthesis algorithms, we generate 3D landmark annotations for this dataset using the method described above. We use a 50-50 split for training and testing. While somewhat small by contemporary standards, this dataset provides a starting point for evaluating this novel problem, while allowing us to compare with previous published systems that were used to benchmark cuboid detection and alignment accuracy. We will release our 3D annotations to spur further research.

Features: We evaluate oriented gradient descriptors (HOG) [11] and state-of-the-art convolutional neural net (CNN) features [38] when defining our final image descriptors Φ .

We resize images to 128x128 pixels before extracting features. For Pose-Normalized, we extract a feature descriptor for each face of the normalized cuboid (concatenating them together to produce Φ). We use standard implementations of HOG and Oxford’s Deep19 CNN model [49], as implemented in the MatConvnet library [58]. We experiment with features extracted from different neural layers, finding the third convolutional layer to perform best.

Classifiers: We describe our representations using a nearest-neighbor (NN) formulation (section 4.3), but the associated feature vectors Φ can be used with any classification system.

We explore SVMs as an alternate state-of-the-art classifier, considering both lin-

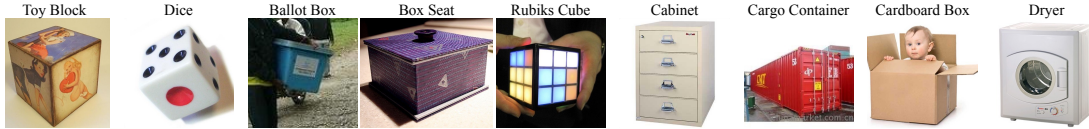


Figure 4.7: Our dataset of cuboidal object categories, re-purposed from the SUN Primitive database [62]. This dataset includes variation in shapes (aspect ratios), viewpoint, backgrounds and clutter.

ear and Gaussian kernels where hyperparameters are selected through 5-fold cross-validation. We make use of the LIBSVM [9] library. When the category model need not generalize across different poses (which is true for Pose-Normalized and Pose-Retargeted), linear classifiers appear to suffice. Pose-Agnostic and Pose-Synthesis must reason across viewpoints, and so Gaussian kernels were vital for good performance.

Landmark prediction: We use a state-of-the-art cuboid landmark detection method which is proposed in Chapter 3 to estimate 2D landmarks at test-time (needed for Pose-Normalized and Pose-Retargeted). Importantly, this system has been shown to produce state-of-the-art alignment results on the SUN Primitive dataset, outperforming prior work such as [62]. We show that some of our simple methods even outperform this body of work, in terms of landmark predictions.

To simplify our analysis, we assume that a ground-truth bounding-box is provided at test-time for all experiments. This is done by running the cuboid landmark detector, then pruning results without sufficient overlap with the ground-truth bounding box. We then take the highest scoring detection among the remaining detections. In our experiments, we set minimum overlap (intersection over union) to be 70%. Note that we also evaluate results with ground-truth landmarks to provide an upper bound analysis.

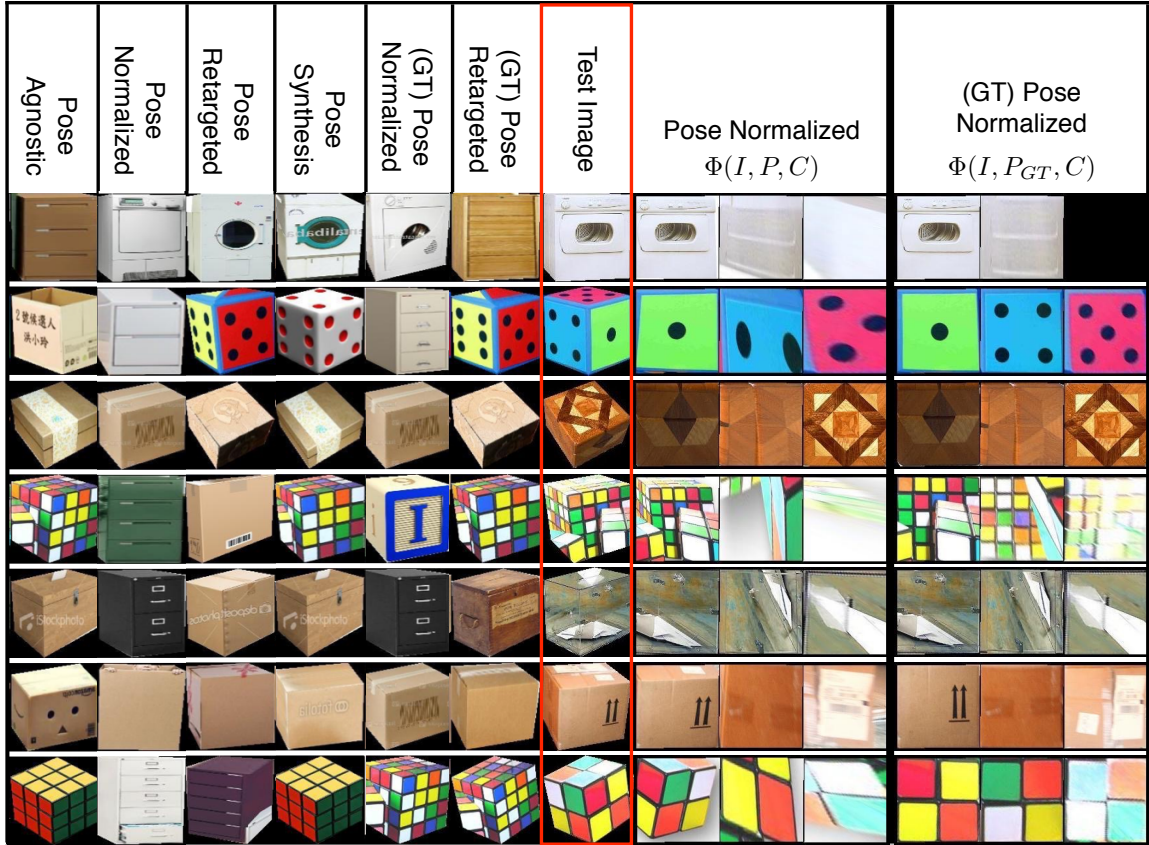


Figure 4.8: The **center red** column designates a test image, while the first six correspond to the NN-matches using our various representations. On the **right**, we visualize both the automatically-generated and ground-truth normalized image. Note how erroneous alignment create significant fluctuations in pose-normalized image, causing wrong NN-matches. Provided ground-truth alignment, pose-retargeting performs the best, while pose-synthesis performs best using automatic alignment.

4.4.1 Categorization

Our primary focus is categorization accuracy of categorization. Table 4.1 evaluates all approaches for object categorization accuracy, for both sets of features (CNN and HOG) and classifiers (SVM and NN).

Normalization: Pose-Normalized performs the worst of all methods, no matter the feature or classifier. One immediate explanation could be that the predicted landmarks are not of sufficient accuracy. To test this hypothesis, we also evaluate

accuracy given ground-truth landmarks. In this setting, Pose-Normalized *does* perform the best out of all methods. Hence one immediate conclusion is that current alignment systems are not of sufficient accuracy to realize the benefits of a pose-normalized representation. In general, we see a significant 11% drop in accuracy in using predicted versus ground-truth landmarks.

It is important to note that these results are based on our specific definition of pose-normalization. Analyzing other valid and plausible pose-normalization strategies (such as those mentioned Section 4.3.2) is subject of future research.

Feature Classifier	HOG	HOG	CNN	CNN
	NN	SVM	NN	SVM
Pose-Agnostic	43.3	58.9	42.6	56.3
Pose-Normalized	36.3	48.1	35.9	41.9
Pose-Retargeted	43	54.4	38.1	43.7
Pose-Synthesis	47.4	57.3	47	59.6
Pose-Normalized (GT)	46.7	63	44.4	54.1
Pose-Retargeted (GT)	54.8	62.2	45.9	56.7

Table 4.1: Categorization accuracy of various approaches. Chance performance on this 9-class task is roughly 11%. The top-four methods are fully automatic, making use of predicted landmarks estimated using the method proposed in Chapter 3 when needed. The bottom-two make use of ground-truth (GT) landmarks. Pose-Synthesis, although simple, consistently outperforms Pose-Normalized and pose-retargeting representation.

Retargeting: However, in almost all cases, Pose-Retargeted outperformed Pose-Normalized, validating our initial hypothesis that normalization (1) is more susceptible to errors in landmark predictions and (2) artificially weights/distorts foreshortened regions of the object.

One might think of Pose-Retargeted with ground-truth (GT) landmarks as an upper bound of Pose-Synthesis, because a perfect synthesis strategy should generate exactly those shapes that align with test images. Interestingly, Pose-Synthesis outperforms Pose-Retargeted-GT with CNN features. Because CNN features are invariant to large spatial deformations, we posit that the learned model can still benefit from

a larger training set that includes shapes outside the test set.

Agnostic vs Synthesis: Overall, we find that Pose-Agnostic performs quite well, consistently outperforming Pose-Normalized and Pose-Retargeted when using predicted landmarks. We attribute this behavior again to the fact that highly accurate landmarks are needed for reliable alignment. However, Pose-Agnostic will struggle to generalize to a viewpoint or shape not seen in the training set. Pose-Synthesis is attractive because it offers generalization without sensitivity to geometric misalignment.

Feature Classifier	HOG NN	HOG SVM	CNN NN	CNN SVM
Pose-Agnostic	43.3	58.9	42.6	56.3
Pose-Normalized	36.3	48.1	35.9	41.9
Pose-Retargeted	43	54.4	38.1	43.7
Pose-Synthesis (w/ background synthesis)	47.4	57.3	47	59.6
Pose-Synthesis (w/o background synthesis)	51.8	58.5	48.2	58.9

Table 4.2: Background synthesis effect in pose-synthesis approach. Similar to 4.1 the table shows categorization accuracy of various approaches. Chance performance on this 9-class task is roughly 11%. One hypothesis is that background provides useful contextual information for classification, thus pose-retargeted and pose-normalized representations are not using it. Here we show that the performance of pose-synthesis without background synthesis is even slightly better than pose-synthesis with background synthesis.

Training data: Because Pose-Normalized may perform better with limited training data (making generalization to unseen views more important), Figure 4.13 plots the accuracy of all methods for differing amounts of real training images. As expected, all methods do better with more data. Pose-Synthesis makes the most of additional data, producing a 15% improvement on average (probably because each additional training sample effectively adds 9 more view-perturbed examples). Agnostic, Retargeting and Normalized see average improvements of 12.7%, 9.8% and 6.5% respectively.

Synthesis strategies: An interesting question is the role of prior in pose-synthesis

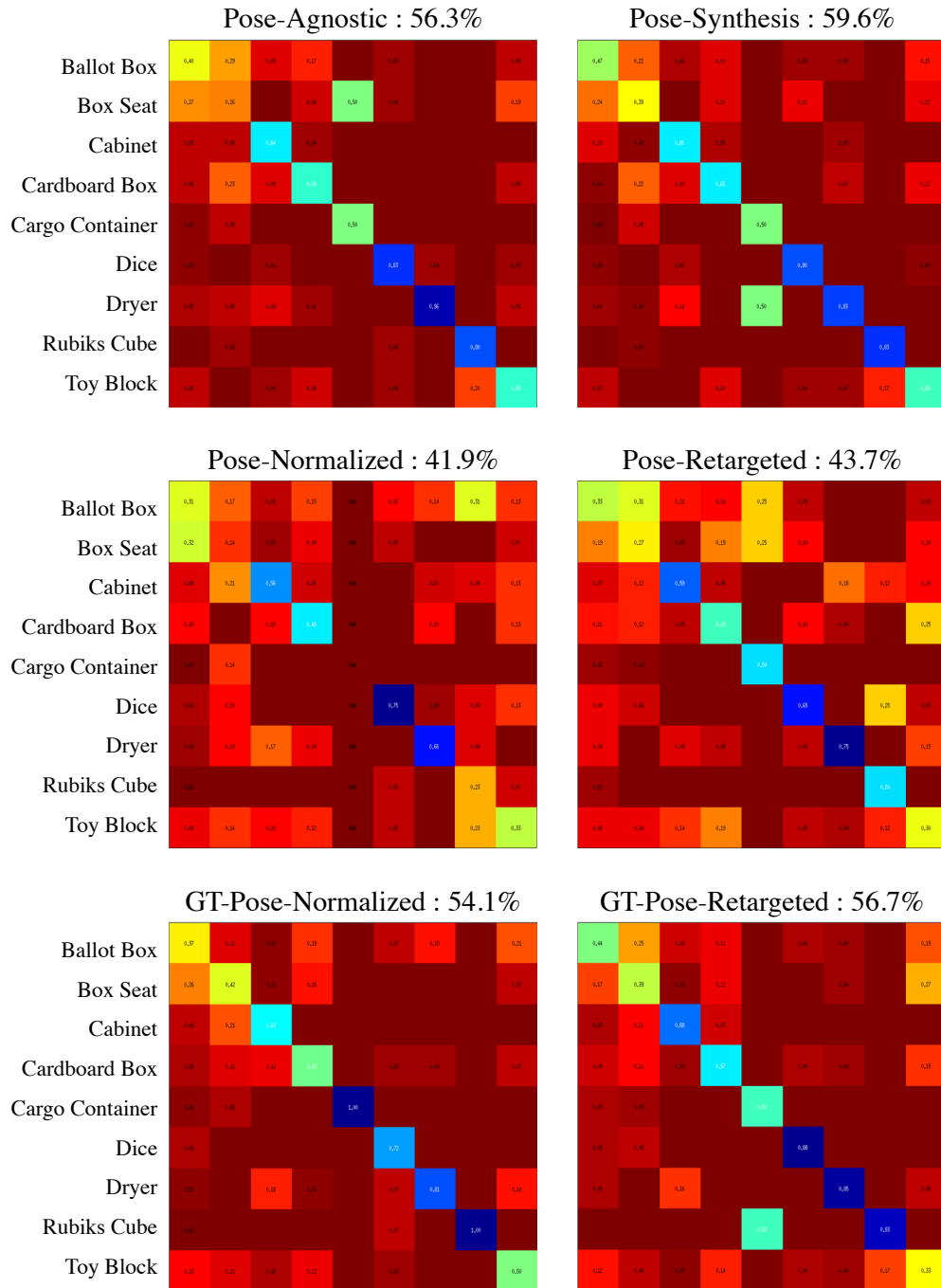


Figure 4.9: Confusion matrices for various representation using CNN+SVM. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.

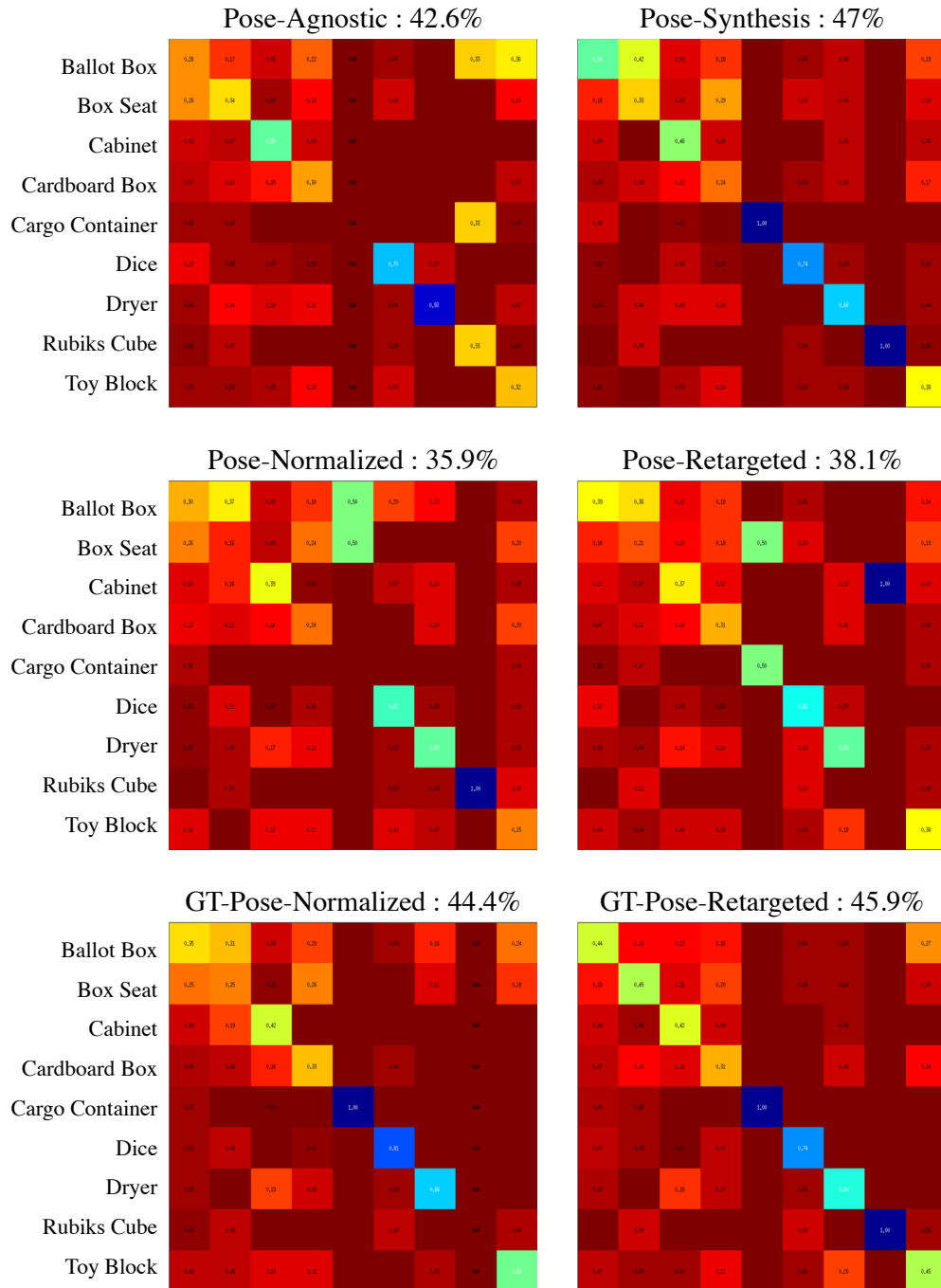


Figure 4.10: Confusion matrices for various representation using CNN+NN. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.

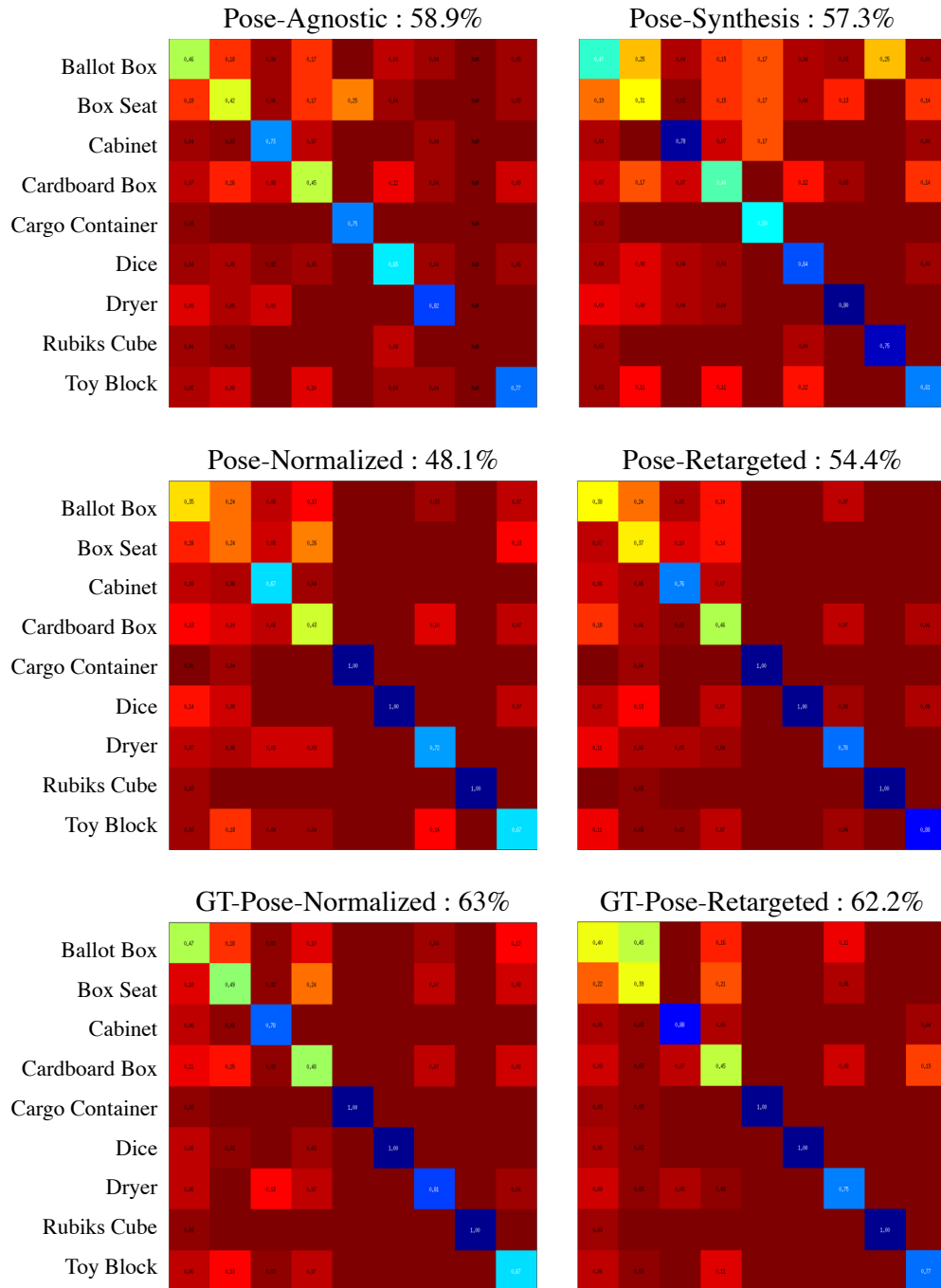


Figure 4.11: Confusion matrices for various representation using HOG+SVM. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.

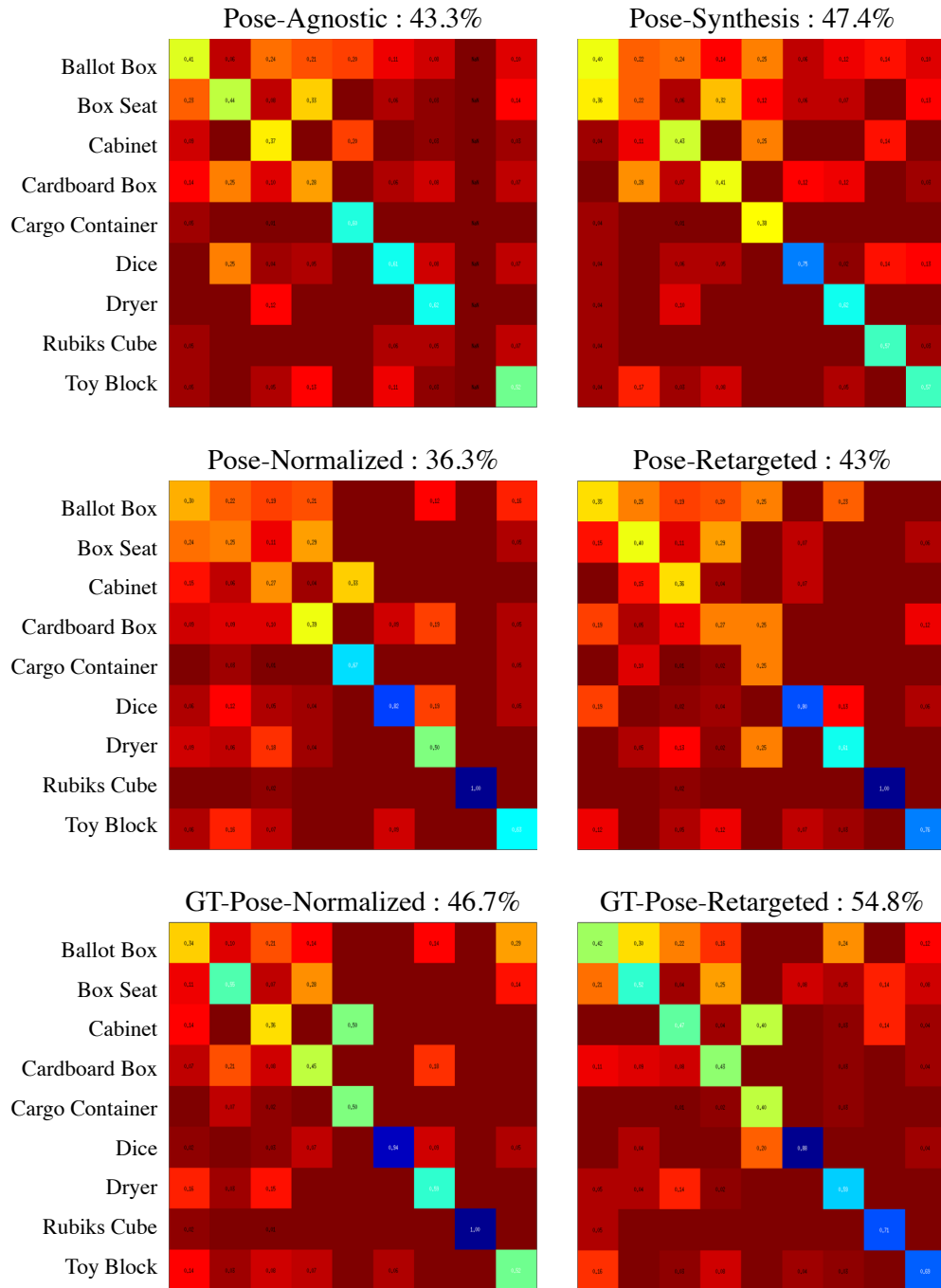


Figure 4.12: Confusion matrices for various representation using HOG+NN. Rows and columns respectively represent ground truth and predicted label. Ballot box, box seat, cabinet, cardboard box and toy block are harder to categorize due to large variation in their appearance.

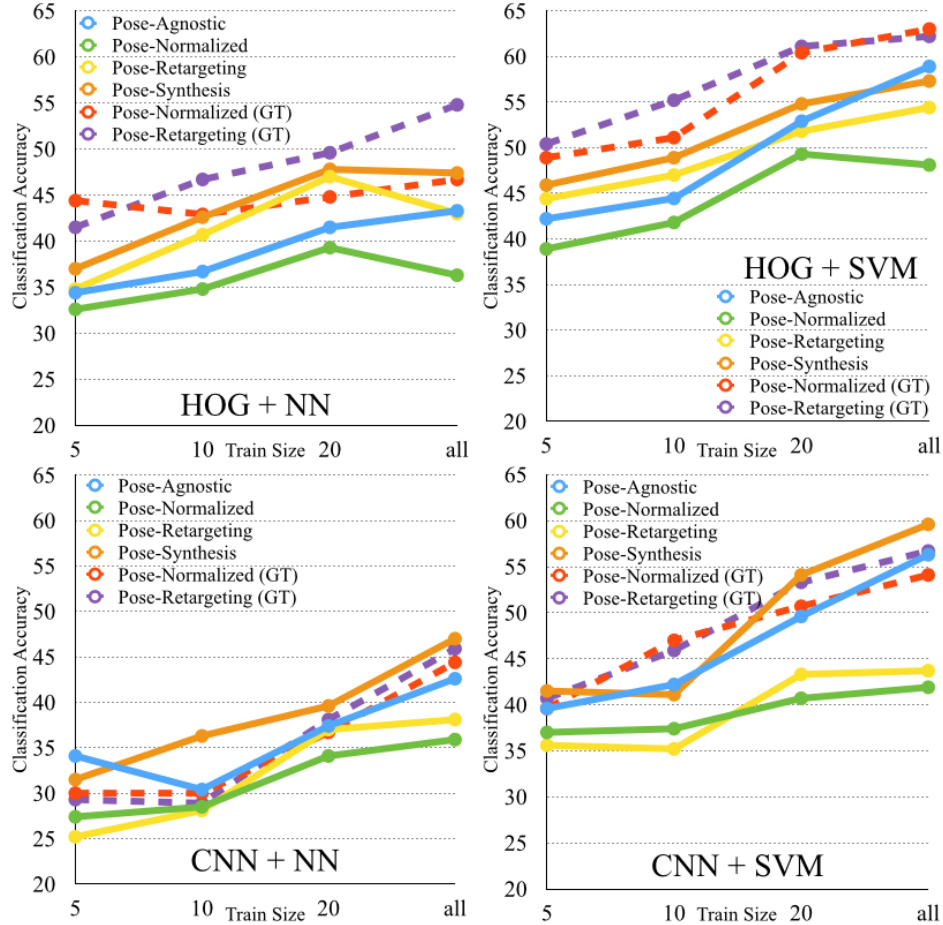


Figure 4.13: Accuracy of all methods for differing amounts of real training images.

and pose-retargeting. As discussed in 4.3.5, pose-retargeting provides strong prior on object pose while pose-synthesis can provide none or weak prior depending on the synthesis strategy employed. Figure 4.14 shows categorization results with various synthesis strategies, each providing different prior by varying perturbation interval. We have evaluated three strategies using: (a) $[-15, 0, +15]$, (b) $[-30, -15, 0, +15, +30]$ and (c) $[-30, 0, 30]$ as perturbation intervals. Intuitively, (a) provides stronger prior and (c) provides weaker prior than others. As expected, (a) outperforms (b) and (c) although (b) synthesizes a much larger space.

One might conjecture from these experiments that pose-retargeting is the optimal approach as it provides the strongest form of prior. That is true only if we have accurate estimation of the object pose at test time. Therefore, we argue that pose-

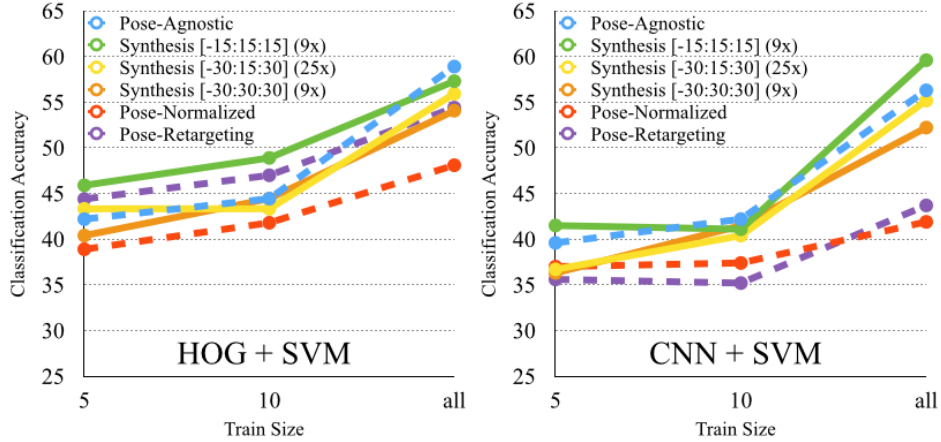


Figure 4.14: Analysis of various synthesis strategies. We show that synthesis based on uninformed pose prior does not perform very well. However, imposing weak prior through training data perturbation is very effective. The curves show that categorization accuracy decreases as the perturbation interval increases.

synthesis is optimal as it does not require pose estimation at test time while providing a framework to employ pose prior from training data.

Background synthesis: One might hypothesize that background image present in pose-agnostic and pose-synthesis representation provides useful contextual information for categorization. We evaluated the effect of background synthesis in the pose-synthesis representation. We have tested it by (similar to pose-retargeted) masking the background in train and test images. As shown in table 4.2, pose-synthesis without background synthesis in fact performs slightly better than pose-synthesis with background.

4.4.2 Landmark localization

To evaluate landmark localization, we use standard approach of counting the number of correctly localized landmarks. A landmark is correctly localized if it lies within t pixels of the ground-truth location, where $t = \%15$ of the square root of the area of the ground-truth box. We use NN-variants of our Pose-Agnostic and Pose-Synthesis models to generate landmark predictions, simply reporting back landmarks associated

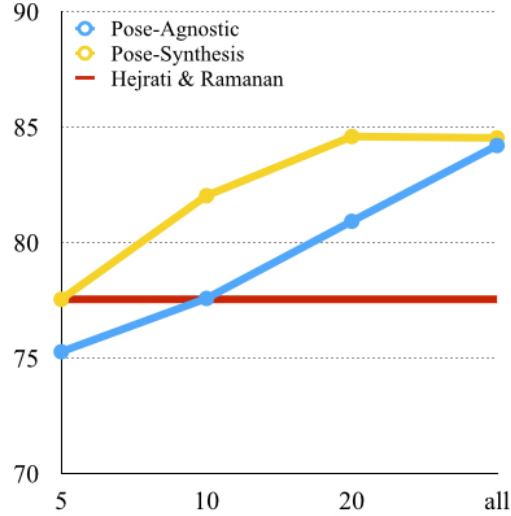


Figure 4.15: Pose estimation accuracy of Pose-Synthesis and Pose-Agnostic (where we assume all training images have been annotated with landmarks), compared to the method proposed in Chapter 3. Our models outperform prior art with as little as 5 training images per class.

with the closest-matching training image (be it a real image or a synthesized one).

Figure 4.15 shows the result of our proposed approaches compared to the state-of-the-art. Both Pose-Agnostic (84.2%) and Pose-Synthesis (84.5%) significantly outperform the previously state-of-the-art method proposed in Chapter 3 (77.5%), which itself outperformed numerous other approaches on this dataset [62, 62]. In particular, our approaches reach state-of-the-art performance trained on only 5 images per category.

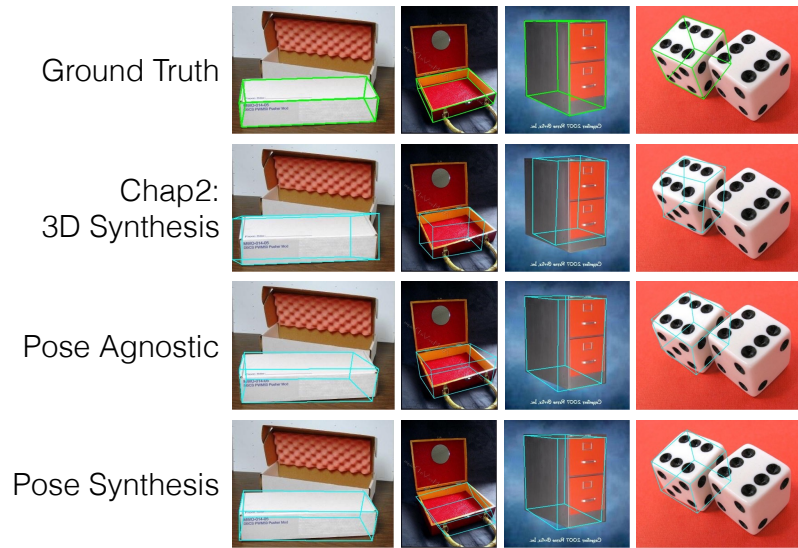


Figure 4.16: Samples of our landmark localization results. We compare the approach in Chapter 3 of this thesis with pose-agnostic and pose-synthesis approaches.

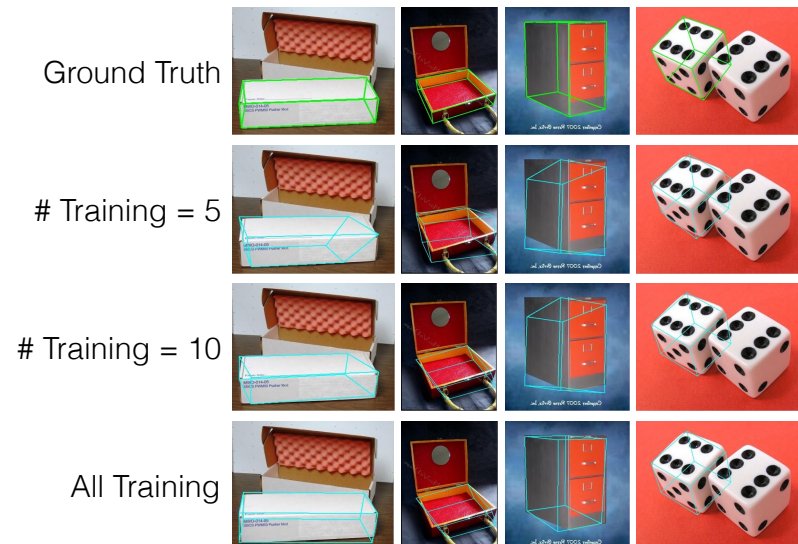


Figure 4.17: Landmark localization results using pose-synthesis approach using differing amount of training data.

4.5 Conclusion

We provided theoretical and empirical analysis of different representations for the novel task of categorizing cuboidal objects, examining Pose-Agnostic, Pose-Normalized, Pose-Retargeted, and Pose-Synthesis-based models. We show that the problem of categorical cuboid classification is a useful testbed for investigating the interplay of shape and geometry while solving a practical task.

Our empirical analysis (using HOG and CNN features and non-linear SVM and nearest neighbor classifiers) reveals the surprising result that Pose-Retargeting and Pose-Synthesis provides a simple approach for dealing with geometric variation that significantly outperforms the common-place technique of Pose-Normalization.

Our empirical analysis might be somewhat restricted by our choice of classifier and features. One might improve the performance of Pose-Normalized representation by employing more complex non-linear classifiers that take advantage of geometric features in addition to appearance. However, our theoretical analysis is independent of these choices and still suggest that Pose-Synthesis provides a simple framework for categorization and pose estimation while allowing to exploit geometric priors from training data.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] ARIE-NACHIMSON, M., AND BASRI, R. Constructing implicit 3d shape models for pose estimation. In *ICCV* (2009).
- [2] BALLARD, D. H. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition* 13, 2 (1981).
- [3] BELHUMEUR, P. N., JACOBS, D. W., KRIEGMAN, D., AND KUMAR, N. Localizing parts of faces using a consensus of exemplars. In *CVPR* (2011), IEEE, pp. 545–552.
- [4] BINFORD, T. Survey of model-based image analysis systems. *The International Journal of Robotics Research* 1, 1 (1982), 18–64.
- [5] BLANZ, V., AND VETTER, T. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 187–194.
- [6] BOURDEV, L., AND MALIK, J. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 1365–1372.
- [7] BOWYER, K., AND DYER, C. Aspect graphs: An introduction and survey of recent results. *International Journal of Imaging Systems and Technology* 2, 4 (1990), 315–328.
- [8] BRANSON, S., VAN HORN, G., BELONGIE, S., AND PERONA, P. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952* (2014).
- [9] CHANG, C.-C., AND LIN, C.-J. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 27.
- [10] CRIMINISI, A., PÉREZ, P., AND TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on* 13, 9 (2004), 1200–1212.
- [11] DALAL, N., AND TRIGGS, B. Histograms of oriented gradients for human detection. In *CVPR* (2005).

- [12] DALAL, N., AND TRIGGS, B. Inria person dataset, 2005.
- [13] DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 248–255.
- [14] DESAI, C., AND RAMANAN, D. Detecting actions, poses, and objects with relational phraselets. *ECCV* (2012).
- [15] EVERINGHAM, M., ESLAMI, S. A., VAN GOOL, L., WILLIAMS, C. K., WINN, J., AND ZISSERMAN, A. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision* 111, 1 (2014), 98–136.
- [16] EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., WINN, J., AND ZISSERMAN, A. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. <http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html>.
- [17] FARRELL, R., OZA, O., ZHANG, N., MORARIU, V. I., DARRELL, T., AND DAVIS, L. S. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 161–168.
- [18] FELZENSZWALB, P. F., GIRSHICK, R. B., MCALLESTER, D., AND RAMANAN, D. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, 9 (2010), 1627–1645.
- [19] FORSYTH, D., MUNDY, J. L., ZISSERMAN, A., COELHO, C., HELLER, A., AND ROTHWELL, C. Invariant descriptors for 3 d object recognition and pose. *IEEE TPAMI* 13, 10 (1991), 971–991.
- [20] GAO, T., PACKER, B., AND KOLLER, D. A segmentation-aware object detection model with occlusion handling. In *CVPR* (2011).
- [21] GIRSHICK, R., FELZENSZWALB, P., AND MCALLESTER, D. Object detection with grammar models. In *NIPS* (2011).
- [22] GLASNER, D., GALUN, M., ALPERT, S., BASRI, R., AND SHAKHNAROVICH, G. Viewpoint-aware object detection and pose estimation. In *ICCV* (2011), IEEE, pp. 1275–1282.
- [23] GOULD, S., FULTON, R., AND KOLLER, D. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 1–8.
- [24] GU, C., AND REN, X. Discriminative mixture-of-templates for viewpoint classification. *ECCV* (2010), 408–421.

- [25] GUPTA, A., EFROS, A. A., AND HEBERT, M. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 482–496.
- [26] HEDAU, V., HOIEM, D., AND FORSYTH, D. Recovering the spatial layout of cluttered rooms. In *Computer vision, 2009 IEEE 12th international conference on (2009)*, IEEE, pp. 1849–1856.
- [27] HEDAU, V., HOIEM, D., AND FORSYTH, D. Thinking inside the box: Using appearance models and context based on room geometry. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 224–237.
- [28] HOIEM, D., EFROS, A. A., AND HEBERT, M. Recovering surface layout from an image. *International Journal of Computer Vision* 75, 1 (2007), 151–172.
- [29] HORN, B. *Robot vision*. The MIT Press, 1986.
- [30] HUANG, G. B., RAMESH, M., BERG, T., AND LEARNED-MILLER, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst, 2007.
- [31] JAIN, A. K., AND LI, S. Z. *Handbook of face recognition*, vol. 1. Springer, 2005.
- [32] JOACHIMS, T., FINLEY, T., AND YU, C. Cutting plane training of structural SVMs. *Machine Learning* (2009).
- [33] JONES, M., AND VIOLA, P. Fast multi-view face detection. In *CVPR 2003*.
- [34] KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 55.
- [35] KANG, Y., LEE, K.-T., EUN, J., PARK, S. E., AND CHOI, S. Stacked denoising autoencoders for face pose normalization. In *Neural Information Processing* (2013), Springer, pp. 241–248.
- [36] KRAUSE, J., STARK, M., DENG, J., AND FEI-FEI, L. 3d object representations for fine-grained categorization. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on (2013)*, IEEE, pp. 554–561.
- [37] LAMDAN, Y., AND WOLFSON, H. Geometric hashing: A general and efficient model-based recognition scheme. In *Second International Conference on Computer Vision* (1988), pp. 238–249.
- [38] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

- [39] LI, Y., GU, L., AND KANADE, T. A robust shape model for multi-view car alignment. In *CVPR* (2009).
- [40] LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P., AND ZITNICK, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*. Springer, 2014, pp. 740–755.
- [41] LOPEZ-SASTRE, R., TUYTELAARS, T., AND SAVARESE, S. Deformable part models revisited: A performance evaluation for object category pose estimation. In *Computer Vision Workshops (ICCV Workshops)* (2011).
- [42] MERRELL, P., AND MANOCHA, D. Model synthesis: A general procedural modeling algorithm. *Visualization and Computer Graphics, IEEE Transactions on* 17, 6 (2011), 715–728.
- [43] OTT, P., AND EVERINGHAM, M. Shared parts for deformable part-based models. In *CVPR* (2011).
- [44] PEPIK, B., STARK, M., GEHLER, P., AND SCHEILE, B. Teaching geometry to deformable part models. In *CVPR* (2012).
- [45] RAZAVIAN, A. S., AZIZPOUR, H., SULLIVAN, J., AND CARLSSON, S. CNN features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382* (2014).
- [46] RUSSELL, B. C., TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. Labelme: a database and web-based tool for image annotation. *International journal of computer vision* 77, 1-3 (2008), 157–173.
- [47] SAVARESE, S., AND FEI-FEI, L. 3d generic object categorization, localization and pose estimation. In *ICCV* (2007), IEEE, pp. 1–8.
- [48] SCHNEIDERMAN, H., AND KANADE, T. A statistical method for 3d object detection applied to faces and cars. In *CVPR* (2000), vol. 1, IEEE, pp. 746–751.
- [49] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [50] SUN, M., SU, H., SAVARESE, S., AND FEI-FEI, L. A multi-view probabilistic model for 3d object classes. In *CVPR* (2009), IEEE, pp. 1247–1254.
- [51] SZELISKI, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [52] TAIGMAN, Y., YANG, M., RANZATO, M., AND WOLF, L. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* (2014), IEEE, pp. 1701–1708.

- [53] THOMAS, A., FERRAR, V., LEIBE, B., TUYTELAARS, T., SCHIEL, B., AND VAN GOOL, L. Towards multi-view object class detection. In *CVPR* (2006), vol. 2, IEEE, pp. 1589–1596.
- [54] TORRALBA, A., MURPHY, K., AND FREEMAN, W. Sharing visual features for multiclass and multiview object detection. *PAMI* 29, 5 (2007), 854–869.
- [55] TORRESANI, L., HERTZMANN, A., AND BREGLER, C. Learning non-rigid 3d shape from 2d motion. *Advances in Neural Information Processing Systems 16* (2003).
- [56] TORRESANI, L., HERTZMANN, A., AND BREGLER, C. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30, 5 (2008), 878–892.
- [57] TORRESANI, L., YANG, D., ALEXANDER, E., AND BREGLER, C. Tracking and modeling non-rigid objects with rank constraints. In *CVPR* (2001), vol. 1, IEEE, pp. I–493.
- [58] VEDALDI, A., AND LENC, K. Matconvnet – convolutional neural networks for matlab. *CoRR abs/1412.4564* (2014).
- [59] VEDALDI, A., AND ZISSERMAN, A. Structured output regression for detection with partial occlusion. In *NIPS* (2009).
- [60] VIOLA, P., AND JONES, M. Rapid object detection using a boosted cascade of simple features. In *CVPR* (2001), vol. 1, IEEE, pp. I–511.
- [61] XIAO, J., HAYS, J., EHINGER, K., OLIVA, A., TORRALBA, A., ET AL. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on* (2010), IEEE, pp. 3485–3492.
- [62] XIAO, J., RUSSELL, B., AND TORRALBA, A. Localizing 3d cuboids in single-view images. In *NIPS* (2012).
- [63] YANG, Y., AND RAMANAN, D. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR* (2011).
- [64] ZHANG, J., MARSZALEK, M., LAZEBNIK, S., AND SCHMID, C. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision* 73, 2 (2007), 213–238.
- [65] ZHANG, N., PALURI, M., RANZATO, M., DARRELL, T., AND BOURDEV, L. Panda: Pose aligned networks for deep attribute modeling. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* (2014), IEEE, pp. 1637–1644.

- [66] ZHAO, W., CHELLAPPA, R., PHILLIPS, P. J., AND ROSENFELD, A. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)* 35, 4 (2003), 399–458.
- [67] ZHU, L., CHEN, Y., TORRALBA, A., FREEMAN, W., AND YUILLE, A. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. *Pattern Recognition* (2010).
- [68] ZHU, X., AND RAMANAN, D. Face detection, pose estimation, and landmark localization in the wild. In *CVPR* (2012).
- [69] ZHU, X., VONDRICK, C., RAMANAN, D., AND FOWLKES, C. Do we need more training data or better models for object detection?. In *BMVC* (2012), pp. 1–11.
- [70] ZIA, M., STARK, M., SCHIELE, B., AND SCHINDLER, K. Revisiting 3d geometric models for accurate object shape and pose. In *ICCV Workshops* (2011), IEEE, pp. 569–576.