Factoring Scenes into 3D Structure and Style

CMU-RI-TR-16-45

David Ford Fouhey

A.B. Middlebury College 2011

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Robotics
June 20, 2016

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:
Abhinav Gupta, Co-chair
Martial Hebert, Co-chair
Deva Ramanan,
William T. Freeman, Massachusetts Institute of Technology,
Andrew Zisserman, University of Oxford

Abstract

Given a single image of a scene, humans have few issues answering questions about its 3D structure like “is this facing upwards?” even though mathematically speaking this should be impossible. We have similarly have few issues accounting for this 3D structure in answering viewpoint independent questions like “is this the same carpet as the one in your office?”, even if the carpets were viewed from different views and have no pixels in common.

At the heart of the issue is that images are the result of two phenomena: the underlying 3D shape, which we call the 3D structure, and canonical texture that is applied to this shape, which we call the style. In the 3D world, these phenomena are distinct, but when we observe the world, they become mixed. Although the identity of both structure and style gets lost in the process, if we know about regularities in both phenomena, we can narrow down the possible combinations that could have produced our image.

This dissertation aims to better enable computer to understand images in a 3D way by factoring the image into 3D structure and style. The key is that we can take advantage of regularity in both phenomena to inform our interpretation. For instance, we do not expect carpet texture on ceilings or 75° angles between walls. By using regularities, especially ones discovered from large-scale data, we can winnow away the possible combinations of 3D structure and style that could have produced our image.

Acknowledgments

Advisors play an integral roles in a dissertation in the sense that they should appear in every category in an acknowledgment section – collaborators, supporters, advice givers, teachers, and many other things. I thus must first thank my advisors, Abhinav Gupta and Martial Hebert, for their many efforts over the past five years. I have enjoyed our time and the work we have done, and I appreciate all that you have done for me.

Thank you to my thesis committee, Deva Ramanan, Bill Freeman, and Andrew Zisserman, for their thoughtful advice, feedback, and flexibility throughout this process. You have all taught me a lot, and I appreciate you taking the time to be part of this.

I have also been fortunate to have a number of fantastic collaborators over the my years at CMU. Thank you to Alvaro Collet, Sidd Srinivasa, Vincent Delaitre, Alyosha Efros, Ivan Laptev, Josef Sivic, Daniel Maturana, Larry Zitnick, Xiaolong Wang, Adrien Matricon, Wajahat Hussain, Rohit Girdhar, Kris Kitani, and Andrew Zisserman. Collaborating with each of you was a pleasure, and I learned a lot from each of you. Thank you for teaching me and showing me things from a new perspective.

Work does not take place in a vacuum, and I have greatly benefited from the labs I have worked in. Thank you to all of my labmates in both Abhinav and, Martial's labs. I am especially indebted to Carl Doersch, Debadeepta Dey, Daniel Maturana, Ishan Misra, Varun Ramakrishna, Scott Satkin, Anish Sinha, Abhinav Shrivastava, Arun Venkatraman, and Xiaolong Wang, who immensely improved my papers, talks, and ideas and who supported me over these years. Thank you for sitting through the bad talks, hearing the bad ideas, and encouraging me. Thank you also to the Oxford VGG for taking me in for the summer and making me feel welcome.

Over the years, I have relied on the advice from a number of people. I especially thank Alyosha Efros, David Forsyth, Srinivas Narasimhan, Yaser Sheikh, Sidd Srinivasa, Larry Zitnick, and Andrew Zisserman for their support and their willingness to give candid advice. Thank you for guiding and encouraging me.

Thank you to Suzanne Lyons Muth, Lynnetta Miller, and Christine Downey, for all of their help during my time at the RI.

I would never have gotten here if not for a number of teachers in the past: Amy Briggs and Daniel Scharstein for getting me started on vision and robotics; Cris Dima for taking me in and teaching me so many things for a summer; and Bob Grossblatt for getting me started on programming.

Thanks to my friends and family for their support throughout the process. I especially thank my parents, sister, and brother for their love and encouragement over the many years. Finally, but far from least, I must thank Justine for her love, encouragement, support, proofreading, and understanding through thick and thin.

Support: The author gratefully acknowledges support from a National Science Foundation (NSF) Graduate Research Fellowship and a National Defense Science and Engineering Graduate (NDSEG) Fellowship over his years at Carnegie Mellon. This work was also supported by NSF IIS-0905402, NSF IIS-1320083, ONR-MURI N000141010934, ONR-MURI N000141612007 and a gift by Bosch Research & Technology Center.

Work on unsupervised surface normals was also supported by Dirección General de Investigación of Spain (project DPI2012-32168). Work on shape attributes was also supported by EPSRC Programme Grant Seebibyte EP/M013774/1. Work on functional constraints for single image 3D was also sponsored by the MSR-INRIA laboratory, the EIT-ICT labs, Google, ERC Activia, and the Quaero Programme, funded by OSEO.

Contents

List of Figures

List of Tables

Notation and Legends

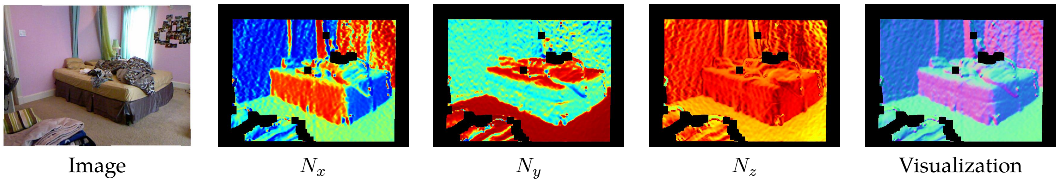


Image N_x N_y N_z Visualization

Figure 1: Legend and convention for Surface Normals, illustrated with ground-truth data. Warmer colors indicate higher values in N_x, N_y, N_z.

Mathematical Notation

$\mathbf{X}$	bold capital letters denote matrices
$\mathbf{x}$	bold lower-case letters denote column vectors
a	roman letters denote scalars
$X_{i,j}$	subscripted roman letters denote elements of matrices and vectors
$\mathbf{1}$	denotes a matrix of ones with size inferred from context

Orientation Legend and Convention

Surface normals are represented with respect to the camera axes.

A normal map is like a 3-channel image, with each 3-channel pixel $\mathbf{n} \in \mathbb{R}^3$ containing $[n_x, n_y, n_z]^T$. Note: $||\mathbf{n}||_2 = 1, n_i \in [-1, 1]$.

Surface normals are visualized by mapping each coordinate to a color (**blue: X**; **green: Y**; **red: Z**). This transformation is linear: the real-valued normal at each pixel $\mathbf{n}$ is converted to a 8-bit RGB visualization by the mapping $255(\mathbf{n}+1)/2$. Invalid or not-predicted normals are indicated by black regions or the underlying image. This is visualized in Fig. 1. To help visualize alignment, normal maps are sometimes blended with the underlying image.

Line-Labeling Conventions

We adopt the classic notation for representing changes in orientation and discontinuities as well as the color scheme used in [65] when visualizing them as feature maps.

$+$	Convex edge	blue
$-$	Concave edge	green
$\rightarrow$	Occluding edge	red

Chapter 1

Introduction

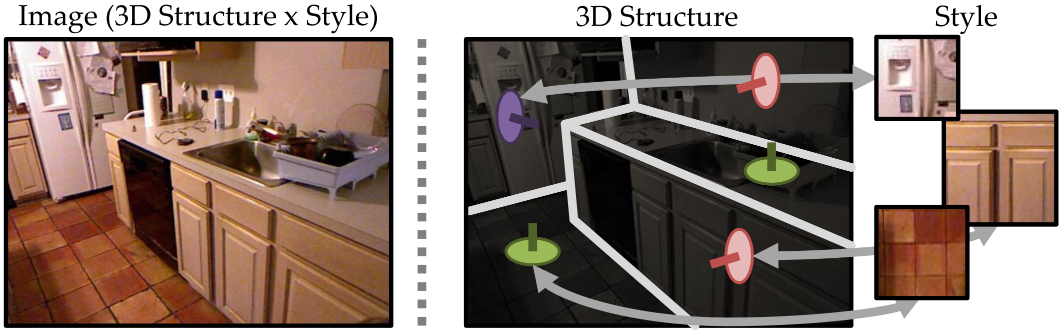


Figure 1.1: Images are the result of two phenomena – the 3D structure of the scene and the style, or underlying content. These are easily factored by humans, who use them to understand the world in a 3D way. This dissertation aims to help computers do the same.

Humans have an incredible ability to understand images in a single glance. Our knowledge goes far beyond simple pointing and naming of people, objects, and locations. Given a single image, we can answer complex questions about the 3D structure of the scene. For instance, the questions "what surfaces are facing upwards?" or "where could I sit?" pose no problems to humans. Further, despite having just one view of a scene, humans are able to subconsciously account for the underlying structure to answer questions like: "is that the same carpet as the one in your office?".

This ability is impossible from a strictly mathematical point of view: any image corresponds to an infinite number of 3D worlds. When each world is viewed, the rich 3D gets projected to a flat 2D image and there is no way to recover it. However, this mathematical reality of too few knowns and too many unknowns is remarkably inconsistent with our own perception of things: given a scene, we see. This dissertation is about giving computers the ability to do the same.

One main reason for the discrepancy between the theory and practice is that what we see in an image is the product of two entities: the 3D properties of the scene, which we call **3D structure**, and a viewpoint-independent texture on top of this structure, which we call

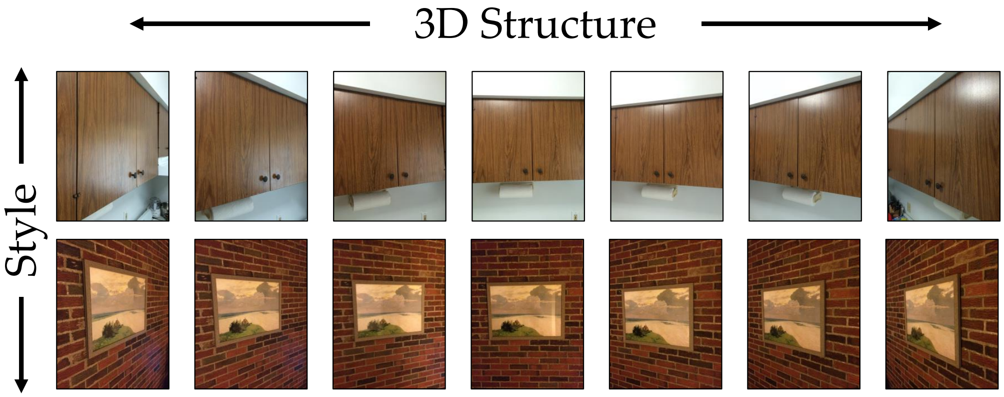


Figure 1.2: The joint space of 3D structure and style. Each column corresponds to a 3D structure / orientation with respect to the viewer; each row corresponds to a texture. The regularities present in this space mean that we can learn to disentangle the two variables.

style. These two properties can be thought of as the factors of an image, akin to the factors of a number. In other words, Image = 3D Structure × Style: the combination of the two factors describes a particular image and two images can be described as sharing the same factor. For instance, two apartments may share the same 3D structure, but one might have wood tiles and the other ceramic. From a graphics perspective, the 3D structure represents the meshes represented with respect to the camera (i.e., incorporating viewpoint) and the style represents the texture that is mapped on top. For instance, we could think of the 3D structure of Fig. 1.1 as a model of the cabinets and refrigerator and floor; the style is what is left after we account for this: the rectilinear detailing on the cabinet, the color of the tiles of the floor, the refrigerator front, etc. Notably, the style is view-independent since view is already encoded in the structure – thus the cabinets, floor, and refrigerator are represented with respect to a canonical view. In this dissertation, we use fronto-parallel or head-on, as our canonical view.

When the image is formed, the two are mixed together to produce the image we see: the canonical view texture of the cabinet, for instance, is distorted to appear as one expects when facing left. Once done, this operation cannot usually be undone, just as in the mathematical case: any image corresponds to an infinite number of pairs of factors, and we can compensate for a change in one factor with a change in the other. This is the main challenge of the dissertation.

If the image is result of mixing the two factors, apparently irreversibly, *why* should we be able to undo this procedure to recover the 3D structure? The answer lies in the regularity in the factorization. *In principle* there are an infinite number of potential pairs of factors for any image. However, *in practice* most pairs are improbable in terms of 3D structure and style. For instance, certain textures are more likely than others, e.g., a rectilinear texture, or *regularity in style*; 90° corners are more likely than 70° corners, or *regularity in 3D structure*; and wood tile floors are less likely to appear on the walls or *joint regularity*. These regularities mean that while many problems still remain theoretically impossible to solve, we can expect to make progress in practice.

The factorization similarly also hints at *what* representations we might prefer for 3D structure. Recall that if we keep the style constant, but change the 3D structure, the apparent texture in the image changes. For instance: if one walks back from a cabinet (i.e., changing

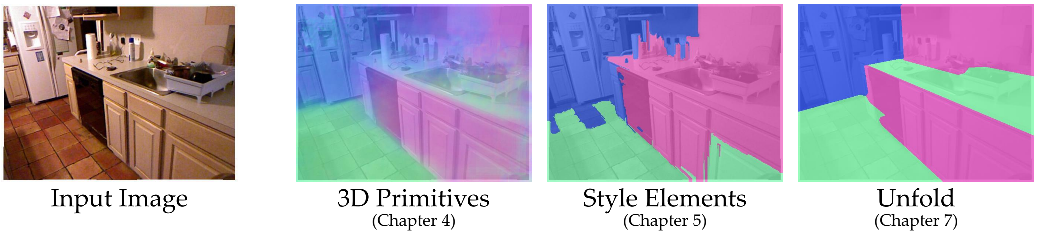


Figure 1.3: The approaches to normal prediction presented in this dissertation.

the distance/depth to the cabinet), it occupies less of ones field of view, or is scaled. If one rotates around the cabinet (i.e., changing the orientation of the cabinet with respect to the viewer), the texture undergoes perspective distortion. Now, consider an image of unfamiliar cabinets. While the style of the cabinets is presumably still composed of straight lines meeting at right angles, we may be unsure of the cabinet's true size. Thus, the latter cue with respect to orientation is still useful whereas the former cue with respect to depth is less so.

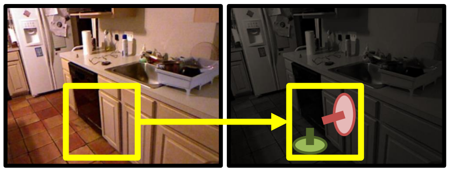
We thus use surface normals, or orientation, as a representation itself in this dissertation. We include additional higher-order and viewpoint-independent shape properties (e.g., planar, non-planarity) since these complement the view-dependent surface normal representation but are often perceivable from the same texture/style distortion in an image, as in the distortion of a polka dot dress for instance.

Finally, the factorization tells us something about *how* we should tackle the problem. 3D structure is physically-meaningful factor and we must treat it as such. Therefore, to tackle 3D structure, we ought to use what we already know about physical properties. For instance, we know that there is a relationship between orientation and the vanishing points in scenes, and that many parts of human scenes are planar apart from discontinuities. On the other hand, to tackle style, we ought to use data-driven techniques since the textures in the world are complex and not governed by physical laws. Throughout this dissertation, we have taken this approach of using a small bit of physical knowledge in combination with data-driven techniques.

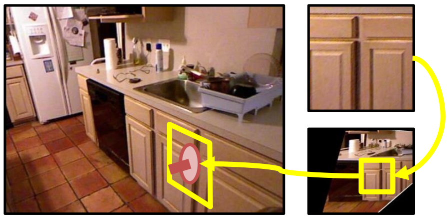
Concretely, the contributions of this dissertation are the following:

1. Dense surface normal estimation as a task for scene understanding. We introduce a number of techniques for solving this task, whose outputs we show in Fig. 1.3.
2. A data-driven technique for recovering dense surface normals from scenes based on the joint regularity of 3D structure and style.
3. A technique, driven by an explicit instantiation of our 3D structure/style factorization, that learns to recover dense normals without any explicit supervision.
4. A way to characterize higher-order 3D structure properties such as qualitative curvature, ground contact, or space occupancy. While these properties are primarily viewpoint-independent and the focus of this thesis, 3D Structure, is viewpoint-dependent 3D, many of these properties can be perceived in the same manner – texture distortion.
5. A scene parameterization that enables introduction of mid-level constraints for 3D structure based on convex and concave edges and produces discrete scene parses in terms of oriented planes and the edges between them.

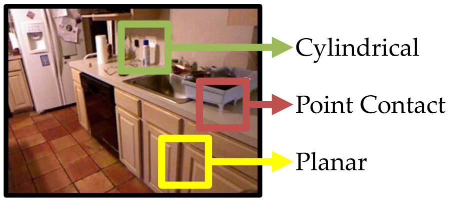
These contributions were previously presented in [51–54] and we briefly describe them now.



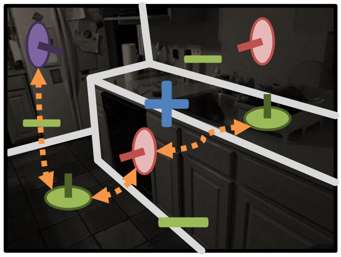
Data-driven recovery of surface normals via data-driven 3D primitives. Chapter 4 begins with an approach that looks at the combination of 3D structure and style. Our approach finds common mid-level patterns in a collection of images and dense surface normal maps. The idea is that certain combinations of 3D structure and style lead to image patches that clearly correspond to a particular underlying 3D structure. For instance, the part of the image corresponding to the leftwards facing cabinets in the kitchen in Fig. 1.1 could be nearly anything in principle; in practice, the patch almost certainly corresponds to leftwards facing cabinets.



Unsupervised surface normal recovery via style elements. One problem with our initial approach is that it is defined in the joint space of 3D structure and style. In other words, for any given style, there is a different element per 3D structure/orientation. Chapter 5 introduces style elements, which are defined in style space as opposed to image space. This makes more efficient use of the data and enables generalization over viewpoint; we also show that these style elements can be learned in an unsupervised way from ordinary image collections by putting a prior on 3D structure.



Higher-order 3D structure properties via 3D shape attributes. Another way to think about 3D structure is via global and non-metric properties. For instance, we understand that the tile floor is flat as opposed to curved, that the refrigerator door handle has a hole where we can grasp it, and that the dish drain touches the counter at a number of points. We explore this idea of interpreting objects in these terms in Chapter 6. These attributes – especially the ones based on curvature – offer a complementary view to the more metric and local approaches presented in this dissertation.



Constraints on 3D structure: We introduce physical constraints from detected convex and concave edges in Chapter 7. The overall idea is that if we can recognize convex (+) and concave edges (-), for instance in the image on the left, we can favor 3D structure interpretations that match these. We introduce a new parameterization of Manhattan-world indoor scenes that enables these sorts of mid-level constraints. Combined, the constraints and parameterization let us produce interpretations of the scene in terms of oriented planar pieces and the edges between them.

Together, these contributions cover many of the ingredients necessary for building a deep 3D understanding of scenes. The approaches tackle multiple levels of understanding:

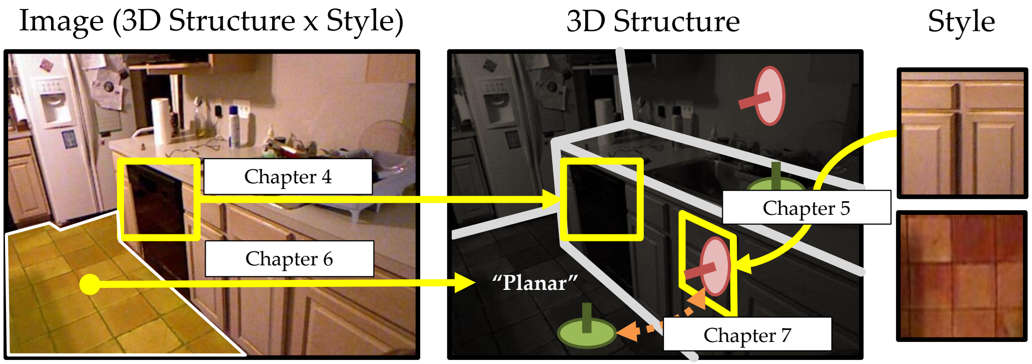


Figure 1.4: A summary of the techniques proposed in this dissertation, ranging from local cues to surface orientation to mid-level constraints to higher-order scene properties such as planarity

the 3D primitives and style elements provide local cues to geometry and produce per-pixel outputs; the constraints on 3D structure introduce mid-level constraints that control geometry between planar surfaces and produce planar parses; and the shape attributes provide a higher-level understanding in terms of shape categories, e.g., which surfaces are planar. In terms of supervision, we address both the supervised (Chapter 4, 6, 7) and unsupervised (Chapter 5) cases. Finally, we have investigated diverse domains: while many chapters address cluttered indoor scenes, we also investigate much less regular objects (Chapter 6) and (briefly) outdoor scenes (Chapter 5).

Organization: We begin by reviewing past literature to properly place our work in broad context in Chapter 2. We point to more specifically related work throughout remaining chapters. We then briefly review surface normals in Chapter 3 as well as alternate representations and motivate our focus on them in this dissertation as our representation of 3D structure.

Once the proper context has been introduced, the core of the dissertation focuses on the methods that compose the contributions of the work. Chapter 4 introduces our technique that looks at the joint space of 3D structure and style and Chapter 5 shows how we can build a similar element-based technique on style. Chapter 6 shows a complementary view in which we focus on qualitative higher-order shape properties. These can be used as a source of additional constraints on our interpretations. Finally, Chapter 7 shows how to constrain the output on the scene level using convex and concave edges.

We then recap the dissertation by discussing additional directions. First, we discuss some of our contemporary work in Chapter 8 that puts this dissertation in the context of our broader research effort, including a discussion of our work on deep learning and human-centric constraints. Finally, Chapter 9 discusses future directions and Chapter 10 summarizes the contributions.

Chapter 2

Background

How can 3D be perceived? This fairly simple question has puzzled people ranging from artists to vision researchers for centuries. While a historical review of theories of 3D perception is beyond the scope of this dissertation, a number of landmarks stand out in terms of their relation to our work. This dissertation addresses the perception of 3D shape from a single image; therefore, we primarily focus on what can be seen in a single view, or what is available if we do not move our heads.

Our search for understanding how 3D can be perceived starts with evidence from psychology and neuroscience. This is because the strongest evidence that what we set out to do is possible lies in the fact that humans seem able to do it. We then turn to the goal of general 3D structure recovery in computer vision, a task that goes back to the beginning of vision. Finally, we put our factorization into a perspective in a long series of factorizations of images in computer vision.

2.1 Biological Vision

The relevant lessons from human perception have to do with the structure of 3D perception (i.e., what problems should we focus on?). Our coverage of human vision is motivated by the discrepancy between how it is believed humans actually perceive in 3D and widespread beliefs about 3D perception in the computer vision community. In particular, one common criticism of the work in this dissertation has boiled down to the idea that 3D perception is about the perception of *depth* from intrinsically *multi-view cues* such as triangulation via stereo or an observer undergoing motion. 3D understanding is thus merely an engineering question of putting two CCDs on cameras going forwards. While multi-view cues for depth are important, especially so for understanding scenes for an active observer, human vision suggests that it is just a piece of the 3D perception puzzle: the monocular cues and non-absolute-depth representations explored in this dissertation are a crucial component in 3D perception in humans and play a strong role even when multiple images are available.

The first lesson from biological vision is that monocular cues are an integral part of the vision process, even under general conditions, and not some backup solution in case one eye is covered. Cutting and Vishton's survey article on 3D perception [28] summarizes the sources of information used and the extent to which they can be used. Purely stereo cues, such as convergence and binocular disparity are largely not useful beyond near regions, whereas most monocular cues work at a much wider range of distances. Moreover, for a non-insignificant fraction of the population, such as the stereo-blind who perceive no disparity or those with weakened eyes, these binocular cues are entirely unavailable or ineffective beyond a meter.

It is possible to further study how these monocular and stereo cues are combined by presenting stimuli where the monocular and stereo cues contradict each other and seeing the resulting perception. Gehringer and Engel [58] presented binocular views of an Ames Room [2], in which monocular cues yield an illusion. In principle, the presence of binocular cues ought to let subjects see through the illusion. Nonetheless, while binocular views diminished the effect of the monocular illusion, the illusion persisted. This demonstrates that monocular cues are heavily used even when binocular cues are available and can even override binocular cues.

The second lesson is that orientation is important by itself and is not just the derivative of perceived depth. Koenderink, van Doorn and Kappers examined the relationship between orientation estimates and depth estimates by humans and concluded that "it is rather unlikely that the attitudes [i.e., normals] are derived from a pictorial depth map" [105]. Among other things, Koenderink notes in [101] that one fundamental flaw in the method of reducing normals to the derivative of the estimated depth is that the precision of all subsequent properties are limited by the precision of the initial depth estimate. This is additionally problematic since many 3D shape cues are for orientation rather than depth [105]. Indeed, Norman and Todd [126] found that orientation estimates were substantially more fine-grained than comparable depth estimates. We explore some reasons why this may be the case in Chapter 3.

The third lesson is that *multiple* representations may be necessary. Cutting and Vishton's survey [28] notes that cues such as texture gradient and lightness are less informative about depth itself, but instead about other shape properties. This multiple-representation view is also offered as an explanation for discrepancies between the accuracy of perception of different representations of shape, such as in the case of depth and normals [105, 126] and normals and curvature [92].

2.2 3D Structure Recovery

The relationship between 3D perception and computer vision dates back to the field's earliest years. Roberts' landmark dissertation, *Machine Perception of 3D Solids* [135], aimed to recover the 3D structure in terms of a known dictionary of 3D objects from a single line-drawing of a scene that indicated the visible 3D structure (i.e., normal and depth discontinuities). Subsequent work after Roberts worked within his framework of line-drawing interpretation but aimed to relax his assumptions, in particular the assumption of known objects. Notable efforts include work by Guzman [67] and the later, more systematic efforts, of Huffman [85] and Clowes [24]. These techniques, perhaps culminating with Sugihara's *Machine Interpretation of Line Drawings* [154], could interpret arbitrary scenes given just a line drawing of them. In contrast to much of modern computer vision, these techniques were driven primarily by a top-down reasoning component: in this case, the top-down component was physical consistency. Other work in this vein attempted to explain objects not in terms of lines, but volumes, including work with generalized cylinders [16]. This led to good initial success in constrained domains or using range data, for instance the ACRONYM system [20]. In each case, the base primitive or unit of inference (e.g., an edge or generalized cylinder) was defined fully in terms of 3D structure.

These techniques, however, were based on a faulty assumption: 3D structural primitives are easy to detect and style can be largely ignored. While perhaps to some degree true when working with range data, this is not at all true with natural images. As this became more apparent, work aimed at handling the fact that the world does not look like line-drawings of polyhedra began: Waltz [160] introduced work that could handle shadows, Kanade [94] non-volumetric objects, and Malik [121] curved surfaces. However, these could not save this line of work. Volumetric primitives proved similarly hard to detect in clutter, as noted in hindsight by [33]. The grand goal of recovering 3D properties from images then seemed to be too far away.

Given the apparent dead-end of unconstrained single-image 3D, research interest turned largely to other problems. In the single image case, there was reinstating of one of Roberts' assumptions, a known dictionary of objects, and this produced a tremendous amount of work that could recognize known objects [123]. While impressive and important, this work fundamentally could not generalize to new objects, except perhaps formed as composites of old ones. In the more general case, there was a flourishing of the geometry of multiple images, leading to sufficiently mature theory to publish a textbook [71]. While this theory produced strong results, the techniques used fundamentally depend on multiple images and thus have limited bearing on the monocular case.

Single image 3D saw a revival in the mid-2000s with the contemporary works of Saxena et al. [143] and Hoiem et al. [82]. Here, the base unit of inference was a segment which took a single label: for Hoiem it was a qualitative geometry label; for Saxena it was a normal and depth (although depth was the end-goal). The success of these methods is attributed to the power of data-driven learning. However, underlying this data-driven factor was the embracing of *style*: rather than hope that style would not interfere, as was the case with edges and geons, these approaches fully exploited style. Indeed, color was a strong cue for both. In principle, color has nothing to do with 3D structure. In practice, however, given our experience of the world, knowing that something is green ought to dramatically change our beliefs.

This produced a renaissance in single image 3D building off of these techniques, typically with a strong top-down component. For instance, one style of approach [74, 148] that has become popular is to model an indoor scene as a parameterized box: in this case, the problem becomes finding the extents of the box that best explain bottom-up orientation beliefs (e.g., orientation evidence from [81, 112]). A single outer shell of a box does a poor job of capturing a scene for many purposes, and thus newer top-down models also incorporate cuboids within the scene [76, 111, 147] or models from online 3D model repositories [8, 139, 140]. There has also been work exploring outdoor scenes as the composition of blocks [63].

There are a number of lessons to draw from the past 50 years of research into 3D perception by computers. Two are particularly relevant to this dissertation. The first is that style is crucial to obtaining good results: progress on unconstrainted images was first obtained when style was embraced (e.g., via data driven techniques that can leverage it) as opposed ignored (e.g., using geons or edges). The second is that top-down information must be combined with bottom up information: success comes when both bottom-up and top-down techniques are combined.

2.3 Factorization

The goal of taking an image and finding representations that can fully explain it also has a rich history. Its introduction is generally credited to Barrow and Tenenbaum's landmark paper on intrinsic images [12]. Modern day intrinsic image techniques have taken the form of explaining an image in terms of shading and reflectance [56,155,156], or further subdividing reflectance into shape and illumination [10]. Similarly, there have been considerable efforts to recovering shape from texture cues such as repeated texture elements [27, 47, 117, 120] (i.e., shape-from-texture) or assumed texture structure [174].

At first glance, this seems to be the same problem; in practice, these approaches have been treated as distinct fields. The primary distinctions are in terms of a trade-off between the level of detail expected and the images to which the approaches have been applied: 3D structure approaches have recovered coarser shape from more general scenes while intrinsic image approaches have recovered more detailed models from more constrained scenes.

Of all methods, our work shares the most in common with shape-from-texture [47,117, 120]. One crucial difference is that these methods seek to obtain shape from regularity *within* an image (i.e., the same element appearing repeatedly); our work, on the other hand obtains shape by regularity *across* images (i.e., we saw this element before in the training set). It is important to note, however, that these approaches provide a clear demonstration of how far one can go with physical knowledge.

Another influential factorization that inspired our framing of the problem is the separation of style and content, as proposed in [158] via bilinear models. In the original task, the model for one factor (e.g., the font of a printed letter or identity of a person) is linear when the other factor (e.g., the letter "A" or the pose of the head) is held fixed. The model for doing so is derived from data. Our dissertation is inspired by this factorization. However, in the domain of 3D structure and style, one of these factors has physical properties that we can model analytically as opposed to in a data-driven way.

One lesson from these approaches is the importance of physical modeling in terms of the underlying components. Unlike other factorizations, we are fortunate to have a component with physical interpretation (3D structure). This physical meaning ensures that we can do better than treating 3D as just another label: as demonstrated by the shape-from-texture literature, the right physical knowledge can go incredibly far.

2.4 This Dissertation

We see a number of lessons to be taken from past work. Biological vision suggests that (1) monocular cues are crucial; (2) orientation stands by itself as a representation; and (3) multiple representations are useful. Throughout, this is reflected in our focus on monocular techniques for orientation (Chapter 4, 5, and 7) and higher-order properties (Chapter 6).

From 3D structure recovery: we know that (1) we must embrace style and that (2) both top-down and bottom-up evidence is useful. Our base units of inference for orientation (Chapter 4, 5) aim directly to recover 3D structure based on local texture and including style as opposed to being based purely on 3D structure. We augment these base units with constraints at mid- and global levels: Chapter 7 shows how to use convex and concave edges and Chapter 6 introduces a global higher-order properties (e.g., planar) that could be used constrain reconstruction

The factorization literature suggests that the physical nature of one of our factors is important. We take advantage of it by incorporating information and techniques such as vanishing points and rectification Chapter 5, and via physical constraints in Chapter 7.

Chapter 3

Representing 3D Structure

One question this dissertation aims to address is: what are the right representations for 3D structure? Do we could model *depth* (how far is each point from us?), *orientation* (in what direction does each point face?), or *curvature* (how does the surface locally bend?) as a base property. It is also an important question: it is always possible to learn a function mapping inputs to outputs and evaluate how well the mapping can be learned. However, how do we know that trying to solve the right problem? This chapter aims to address this question and answer *what* properties we model and *why*.

There are two general strategies for building a 3D understanding. The first, and arguably simplest, is to aim to extract a property – absolute depth – from which all other 3D properties can be derived. This is appealing in the sense that once we have depth, all other properties should fall out. The second option is to instead try to extract properties, possibly many of them, that are more easily extracted from images due to regularities in the data. By extracting these properties, including perhaps sometimes absolute depth, we can obtain a better understanding.

We take the latter approach. In particular, this dissertation is primarily about the estimation of orientation properties as well as higher order properties such as qualitative curvature. This is a deliberate choice motivated by a number of reasons that we will explain in this chapter. Among other things, these orientation properties are more easily related to texture and are less affected by fundamental ambiguities in 3D perception. Additionally, as we show throughout in this dissertation and as has been attested in the human-vision literature, the indirect prediction approach produces worse results than directly predicting the properties. Thus, even if the final representation involves depth and its estimation from images, we should to also model orientation itself.

This chapter begins by mathematically defining properties in Section 3.1. We then motivate our choice of orientation, or surface normals and describe some practical issues in Section 3.2. Finally, we briefly discuss some other properties in Section 3.3 as it is likely that the correct representation involves multiple properties;

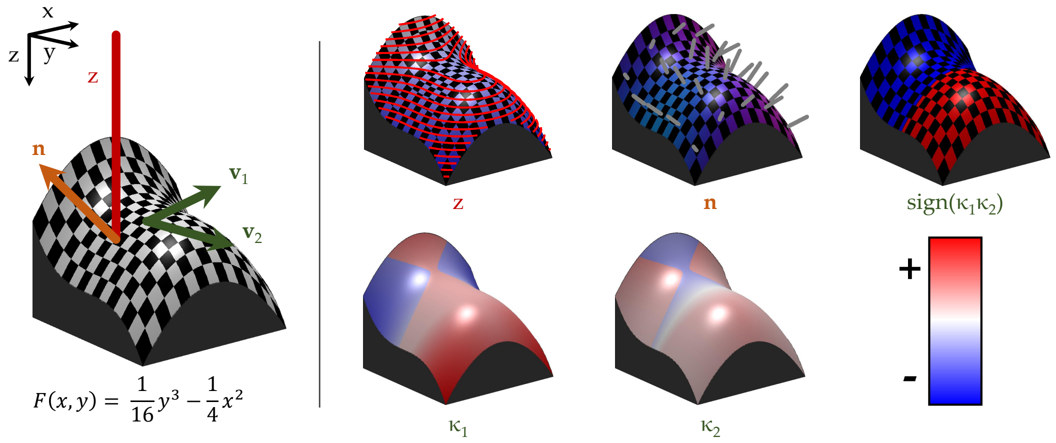


Figure 3.1: An illustration of the various properties considered in this dissertation, namely the zero (depth), first (surface normals), and second order (curvature) characterizations of shape. We show the isocontours of z, a quiver plot of $\mathbf{n}$ (with the common underlying color scheme used in this work), and various aspects of the curvature. Note that wherever the shape bends inwards in one direction and outwards in the other, the curvature signs have opposite signs; where they both bend consistently, the signs are the same.

3.1 Mathematical Background

We start by describing what we mean by each property. This has two advantages. First, intuition, while important, frequently fails in 3D: even Marr [122], for instance, was famously wrong regarding the relationship between the curvature of a surface and its occluding contour [99]. Second, the mathematical definitions force certain aspects of the properties to the fore. A more thorough introduction can be found in [23].

For mathematical convenience, we will work with a form known as a Monge form, or surface floating in 3D space away from the 2D $X - Y$ plane with distance defined as a continuous function $F(x, y)$. While this form cannot represent some objects, it is convenient to quickly introduce concepts since it results in fairly simple equations for each property. For convenience, let $F_x = \frac{\partial}{\partial x}F$, $F_{x,y} = \frac{\partial}{\partial y}\frac{\partial}{\partial x}F$, etc. We show renderings with a sample surface $\frac{1}{16}y^3 - \frac{1}{4}x^2$ in Fig. 3.1.

Depth. The depth, $Z(x, y)$, is the height of the surface or the distance to the camera, or the zeroth order description of the surface,

$$Z(x, y) = F(x, y). \tag{3.1}$$

From the beginning, one should note that depth is in absolute terms. Thus there is no inbuilt compensation for the fact that it is difficult to identify the scale in an image (distinguish a well-done scale model and the original object) or to tell if the image is a cropped shot of a far away object or a full-view shot of a close object.

Surface Normal. The *surface normal* $\mathbf{n}(x, y)$ quantitatively characterizes the direction the surface faces or the first-order way the surface changes. In particular, if we take the tangent plane at the point; the normal is normal vector facing off the plane. Mathematically, the normal is the partial derivatives of F constrained to be unit norm:

$$\mathbf{n}(x,y) = \frac{1}{\alpha(x,y)} \left[\begin{array}{c} -F_x(x,y) \\ -F_y(x,y) \\ 1 \end{array} \right] \tag{3.2}$$

where $\alpha(x,y) = \sqrt{F_x(x,y)^2 + F_y(x,y)^2 + 1}$ is the normalization factor that ensures the vector is unit norm. Note that if there no discontinuities in F – in other words if there are no occlusion boundaries – one can recover F up to a multiplicative scale factor and as well as the typical additive ambiguity involved in integration.

There are two potential difficulties here. The first is that $\mathbf{n}$ is always unit norm, which means that we must take some care in processing normals; the second is that one could also consider $-\mathbf{n}$ to be the normal. In other words, the normal could be interpreted as either pointing towards the camera, or could be pointing away from the camera. The only important thing is to keep a consistent interpretation.

Curvature. Finally, *curvature* characterizes the second-order way the surface changes at a point. These properties are defined with respect to a local tangent plane, which we can then map to the full 3D space. The properties come from matrices known as the first and second fundamental forms. The first fundamental form is a 2D matrix defined at each point x, y as:

$$\mathbf{I}(x,y) = \left[\begin{array}{cc} 1 + F_x(x,y)^2 & F_x(x,y)F_y(x,y) \\ F_x(x,y)F_y(x,y) & 1 + F_y(x,y)^2 \end{array} \right], \tag{3.3}$$

and the second fundamental form is defined as:

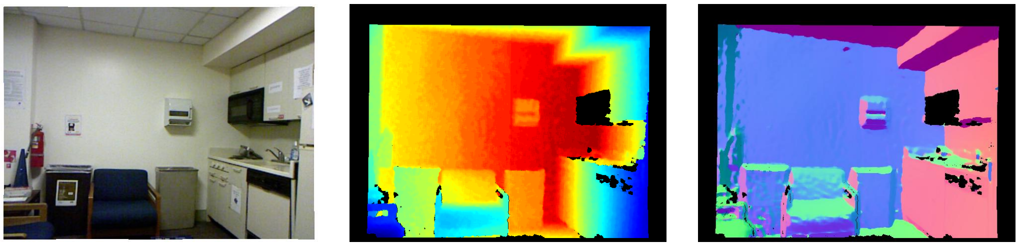
$$\mathbf{II}(x,y) = \alpha(x,y) \left[\begin{array}{cc} F_{x,x}(x,y) & F_{x,y}(x,y) \\ F_{x,y}(x,y) & F_{y,y}(x,y) \end{array} \right]. \tag{3.4}$$

Together, these produce a shape matrix $\mathbf{S}(x,y) = \mathbf{I}^{-1}(x,y)\mathbf{II}(x,y)$. The curvature is then described by the eigenvalues and eigenvectors of $\mathbf{S}$. Specifically, let κ_1, κ_2 and $\mathbf{v}_1, \mathbf{v}_2$ be the eigenvalues and eigenvectors of $\mathbf{S}$. The eigenvectors are referred to as the *principal directions*, and their corresponding values are referred to as the *principal curvatures*.

Intuitively: the principal directions $\mathbf{v}_1, \mathbf{v}_2$ are the directions in which the surface changes the most and the least and their corresponding values κ_1, κ_2 encode how the surface changes in that direction. The main item of interest for us are the principal curvatures κ_1, κ_2. The signs of κ_1, κ_2 define qualitatively how the surface changes: $\kappa > 0$ indicates a bending outwards (the outside of a sphere) whereas $\kappa < 0$ indicates bending inwards (the inside of a sphere); naturally, $\kappa = 0$ indicates flatness. The magnitudes of κ_1, κ_2 define quantitatively the extent to which they change (and are thus *not* scale-invariant). These are cumbersome to work with directly and typically curvature is characterized by some expression that combines both κs, for instance Gaussian curvature $\kappa_1\kappa_2$, or various sign properties.

3.2 Surface Normals

Most of this dissertation uses surface normals as its representation of 3D structure. This is a conscious decision that sets us apart from a large literature on *depth* estimation. Here, we introduce a few practicalities about normals that are useful for reading the subsequent chapters. We then motivate why we have chosen surface normals as our primary representation compared to depth.



Input Depth Normals

Figure 3.2: A typical scene imaged by a range sensor (data from [150]). Black regions indicate places where the sensor could not obtain a reading of distance.

3.2.1 Practical Considerations

Unlike depths, which are scalars and the direct output of range sensors, surface normals are unit vectors that are computed from range measurements. Thus, they require some degree of thought in terms of estimation and evaluation. Here, we give a brief introduction on: how one computes them; how we evaluate predictions of them; and, since vanishing points are used frequently in this dissertation, the relationship between vanishing points and the surface normals of the scene.

Computing and Visualizing. Given an analytic representation F of a surface, one could use Eqn. 3.2. However, in practice, we do not have an analytically F, but instead a range-map Z obtained from a range sensor, which can be backprojected to a 3D map indicating the XYZ coordinates of each point *with respect to the camera axes*.

We fit a normal to nearby points and then denoise. We first find each point's neighborhood $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and then perform PCA on P: we compute the mean $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ of P and then compute the covariance matrix $\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$. The least-squares normal $\mathbf{n}$ is given by the smallest eigenvector of $\boldsymbol{\Sigma}$. This approach is simple and can be trivially vectorized, but generally suffers from noise due the sensor and the difficulty of choosing the "right" set of neighbors. We thus denoise the resulting normals. In the past, we have used both a total-variational approach and a bilateral filter.

One complication throughout is the fact that range sensors fail to obtain readings at many locations of typical human scenes. We ignore these in generating training data and in evaluating. Among other things, typical sources of failure include specular surfaces, too-close and too-far points, direct sunlight, surfaces that are nearly-parallel to the camera axes. These are frequently handled in RGBD datasets (e.g., [150]) by interpolating the missing data from nearby known points. Since this dissertation is about the recovery of 3D properties from images, we instead ignore the missing data in both generating training data and evaluating methods: we cannot taint our ground-truth with imputed data.

Once estimated, the resulting normal map is visualized by linearly mapping the components of the normal to the three RGB channels. Bright green, for example, indicates upwards facing surfaces. Missing data is indicated by black. We visualize a normal map and its depth channel in Fig 3.2. Note that our visualization is not uniquely defined, even given the camera axes: one can permute the axes arbitrarily, and flip the signs, as seen in [38,109].

Evaluating. Given a set of predicted surface normals and ground-truth normals, we report a number of summary statistics about the amount by which the predicted normals deviate from the corresponding ground-truth ones. These metrics have been adopted by the community and used in subsequent surface normal estimation work by others (e.g., [9, 38, 108]).

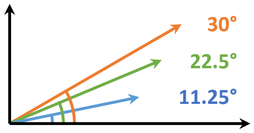


Figure 3.3: The thresholds visualized in 2D.

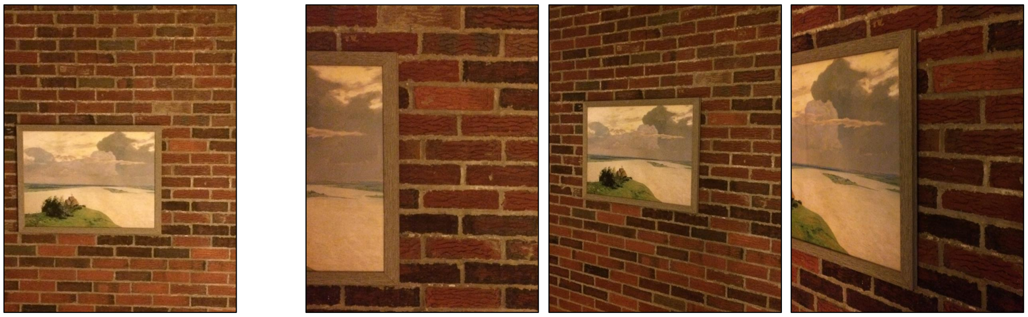
Given the set $\{\theta_i\}_{i=1}^n$ of all angular errors (i.e., $\theta_i = \cos^{-1}(\hat{\mathbf{n}}_i{}^T \mathbf{n}_i)$ for predicted and ground-truth normals $\hat{\mathbf{n}}_i$ and $\mathbf{n}_i$), we compute and report a number of summary statistics. These statistics are evaluated only on points that were actually measured by the 3D sensor generating ground-truth. We report the mean error $\frac{1}{n}\sum_{i=1}^n \theta_i$, median error, and root-mean squared error (RMSE) $\sqrt{\frac{1}{n}\sum_{i=1}^n \theta_i^2}$. These metrics, especially the mean and RMSE, blend both large and small errors together. This can hide important information for downstream applications: a system that is sometimes accurate and sometimes off by a large amount has different applications than a system that always produces mediocre estimates that are never accurate, but never completely wrong. We therefore additionally use percent-good-pixel metrics (PGP) adopted from the stereo literature [144]. These treat a pixel as correct or incorrect according to some threshold t, and are computed as $\frac{1}{n}\sum_{i=1}^n I(\theta_i < t)$. We use $t = 11.25°, 22.5°, 30°$ (or $\frac{1}{8}$, $\frac{1}{4}$, and $\frac{1}{3}$ of $90°$), which we illustrate in Fig. 3.3.

It should be noted that different local estimators and denoisers will produce different normals. In our experience, however, changing the ground-truth changes the absolute performance of methods but leaves the relative performance intact.

Relationship with Vanishing Points. There is a fundamental relationship between the dominant surface normals of a scene and its vanishing points. Specifically, given a projective camera with intrinsic matrix $\mathbf{K}$; the vanishing point $\mathbf{v}$, represented in homogeneous coordinates, corresponds to the normal $\mathbf{K}^{-1}\mathbf{v}$. There is an underlying sign ambiguity since the vanishing point corresponds to $-\mathbf{K}^{-1}\mathbf{v}$ as well: a vertical vanishing point infinitely far below the camera's principle point (typically the center of the image) corresponds to both upwards and downwards facing directions in the scene.

This enables a way to obtain substantially better results in man-made environments: we first estimate a coordinate frame with an auto-calibration system like [74, 137]. This set of 6 normals serves as our label space. For instance, we can use this to make our recognition problem simpler by factoring out 3D structure, as in Chapter 5; similarly, we can perform optimization over this space as in Chapter 7. Note that using this label space exclusively has downsides: we give up trying to detect smooth or non-axis aligned objects; additionally, the vanishing point estimator becomes a single point of failure and errors are propagated downstream. We explored a softer approach in which vanishing points are used as options in [162].

In practice, methods making this assumption have different performance characteristics. They have substantially improved performance on the PGP-11.25 metric: getting a highly accurate normal out of texture with a descriptor such as HOG [29] is difficult. On the other hand, results are now frequently off by $\approx 90°$ (when they were previously off by $50°$), which substantially increases the mean and RMSE.



Sample Texture Change in Depth Change in Orientation Change in Both

Figure 3.4: An illustration of how changes in orientation and depth affect the appearance of texture.

3.2.2 Why Surface Normals?

One natural question is: why surface normals? It is clear that we do need normals to answer questions in a scene, but why should we treat them as a separate problem?

There are a substantial number of benefits to working with surface normals, due to their *quantitative* nature as well as their representation of *orientation*. Specifically, normals are more related to style in a more useful way than either qualitative orientation or quantitative depth. But more broadly speaking compared to depth, they suffer from fewer ambiguities for estimation and evaluation. Moreover, directly modeling and estimating them is superior to indirectly estimating them, or treating them as just the derivative of estimated depth. Finally, normals have advantages as an output representation for subsequent systems (e.g., stereo, higher-level understanding systems) compared to both absolute depth and qualitative orientation.

Direct relationship to style. The primary advantage with respect to this dissertation is that the normals are more easily related to style compared to depth. This is perhaps best illustrated visually. We do so in Fig. 3.4 with a sample image and changes in both depth and orientation with respect to the camera. Changing the depth scales the texture; however, from just one image, it is not clear if the bricks and picture are small or if they are far away. On the other hand, changing the orientation of the surface induces perspective distortion: the previously almost-perpendicular lines are clearly no longer perpendicular.

Consider then what we must know about the world in order to recover depth or normals from these images. To recover depth, we must know about the true scale of the bricks, and use the relationship between the true and apparent scale to recover the depth. To recover normals, we can instead rely on knowledge of what bricks look like in general when viewed head-on, and use this relationship. Notably, this relationship holds *irrespective* of the size of the bricks, and we could probably generalize this relationship if we were only familiar with cinder blocks, for instance.

Similarly, consider representing these normals instead with a coarse qualitative understanding, for instance, as in [79], binning into effectively five types. One can readily imagine the slow transition between the various views presented in Fig. 3.4, and how the apparent texture would change as the view would change. A quantitative representation lets us acknowledge that there are many angles between the different views; a qualitative one, on the other hand, does not.

We take also wish to use this opportunity to note that the choice of coordinate frame is also important. To illustrate this, consider the alternate representation of some global coordinate frame (e.g., a frame extracted from three orthogonal vanishing points in a scene). The downfall of such a global representation is that this external frame breaks the relationship between camera viewpoint and perceived texture, as illustrated in 3.4: a brick wall's lines are usually not perpendicular, but do become perpendicular when viewed head-on. Nonetheless, in the global representation, these very different appearing textures have the same label. In comparison, change in texture directly reflects change in camera viewpoint, or equivalently pieces of texture with the same surface normal with respect to the camera's viewing direction look similar. This direct relationship is more appealing from a learning perspective.

Fewer ambiguities. It is understood (e.g., see [102]) that the perception of objects in space is modulo a number of ambiguities. Surface normals are inherently less affected by these ambiguities compared to depth. This has a number of practical consequences.

As an example, consider understanding the buildings depicted in Fig. 3.5. Even before analyzing the situation, note the relative confidence with which the normals of the buildings can be determined compared to their absolute depth. Now suppose the recovered understanding of the buildings was off by some additive global constant of 1 meter; this is perfectly plausible given the great distance of the buildings, but of no importance to understanding the structure. In a depth representation, this is immediately reflected in the error. On the other hand, as basic calculus tells us, this has no impact on the normals. Similarly, suppose Fig. 3.5 is revealed to actually be a miniature, and thus the depth estimates are off by a global multiplicative factor. The depth error is now disastrously high. The normals, however, are unaffected since all dimensions are scaled accordingly.

Due to this, depth is estimation rarely evaluated directly but instead, which suggest underlying issues with the representation. Most metrics in current use (e.g., see [38,96]) transform the depths to compensate. For instance, log-depth or dividing by the ground-truth depth is popular. Even after these transformations, Eigen et al. [39] showed that a large fraction of traditional evaluation metrics were driven by getting the absolute scale right (i.e., miniature-vs-real-life); accordingly, they introduced a new metric that compensates for this with a global translation. These transformations correct for scale issues and ensures that a 1m error at 2m is more important than at 20m. However, at the same time, they also hide any errors in fine details. In comparison, other hand, normals have been relatively successfully evaluated using direct comparisons with the ground-truth and no additional post-processing steps.

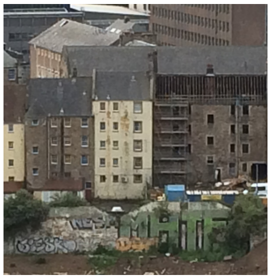


Figure 3.5: Buildings

As additional consequence, this means that normals can be readily estimated locally while absolute depth, on the other hand, is typically estimated by labeling the whole image. This means that normal-based methods can easily interpret parts of the scene: for instance, the same model ought to be able to work on a cropped chair and a chair in context of a whole scene. In the depth case, this is not necessarily the case, as the absolute depth is difficult to determine in the cropped case.

Direct estimation is a better learning problem. One tempting argument for depth estimation is that, in principle, one can derive many other properties from it once it is estimated. This reasoning goes against evidence in both human and machine vision of the both an analytic and empirical nature. The previous chapter, in Section 2.1, discussed some results to this effect from the human vision literature, and we now elaborate on why this might be the case and discuss its practical impact.

Analytically, it can be seen purely in the mathematical form that the goal of depth estimation does not necessarily lead to good orientation estimates. This is perhaps obvious to some, but worth repeating. As illustration, consider a 1 pixel tall and 2 pixel wide depthmap z_i and an estimated depthmap $\hat{z_i}$ for $i = 1, 2$. The goal of getting a good depth estimate is an objective function similar to $(z_1 - \hat{z_1})^2 + (z_2 - \hat{z_2})^2$, or minimizing the difference in depths (here, using an L_2 penalty). Already, it can be seen that there are no terms that resemble derivatives (i.e. normals). Consider then the goal of predicting normals. For simplicity, let us focus on only the x-component and use a finite-difference to compute it. The goal of getting a good surface normal estimate is $((z_1 - z_2) - (\hat{z_1} - \hat{z_2}))^2$, in the x-component, ignoring a normalization factor. In other words, the *difference* between the two pixels ought to be similar. There is thus no particular why reason depth estimates ought to predict normals.

This mismatch between the depth objective and the surface normal objective has substantial practical effects for computer vision. In our experience, especially as documented in Chapters 4 and 6, the indirect approach always does worse than directly predicting normals.

To demonstrate the point in a way that is independent of our own methods, we briefly report a simple experiment comparing a direct (i.e., predict normals) and an indirect (i.e., compute normals from predicted depths) approach. We take the approach of [38], which predicts, on a per-pixel basis, both depth and normals in RGB images using an identical architecture and which achieves state-of-the-art results. As an additional boost to the indirect approach, the objective used to train the depth network also tries to ensure that finite-difference normals from the output depth match the ground-truth finite-difference normals. We then compute normals from the depthmap using ground-truth camera intrinsics and the identical technique used to compute the ground truth (including cross-bilateral denoising *not* used on the normal predictions). It should be noted that this setup is deliberately favorable to the indirect method: the ground truth is identical, the depth objective even incorporates normals, and the depths are per-pixel. Moreover, the depths are per-pixel as opposed to per-segment, which would produce a series of flat fronto-parallel planes. However, the indirect method does substantially worse: 9% fewer pixels are within $11.25°$, and the mean error is 6% higher. If one wants a high-quality normals, the choice then is clear.

Immediate applications to other understanding systems. As discussed in Chapter 2, monocular cues play a substantial role in human perception even in free-viewing conditions. Since the publication of our first paper on the topic, monocular techniques for surface normals, including the techniques we proposed, have been integrated into traditional multi-view systems.

For instance, the system introduced in Chapter 4 was subsequently used in both monocular SLAM [25] and stereo [68] systems and was found to improve results compared to traditional approaches. Similar results were found in [69], which used another normal estimation system [109], and in [57], which used normal prediction in the context of multi-view stereo.

Surface normals are attractive as a cue for subsequent systems, especially 3D ones, in comparison to qualitative representations and to depth representations. The advantage of

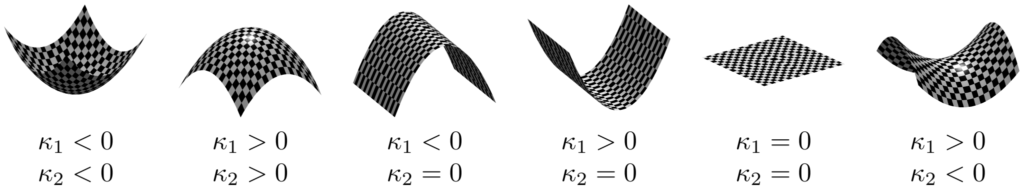

$\kappa_1 < 0$ $\kappa_2 < 0$
$\kappa_1 > 0$ $\kappa_2 > 0$
$\kappa_1 < 0$ $\kappa_2 = 0$
$\kappa_1 > 0$ $\kappa_2 = 0$
$\kappa_1 = 0$ $\kappa_2 = 0$
$\kappa_1 > 0$ $\kappa_2 < 0$

Figure 3.6: Qualitative/Sign Curvature in the style of Koenderink

the quantitative aspect of normals is that they have *physical* meaning. Thus, they directly have bearing on the resulting 3D interpretation and can be integrated in a meaningful and principled way. The advantage compared to depth is that estimated normals complement traditional multi-view cues (e.g., triangulation) more than estimated depth. Triangulation already reveals depth and thus, while a second, monocular, source of information is useful check (e.g., see [142]), the properties being predicted are the same. On the other hand, texture- or shading-based approaches provide new information to triangulation techniques. Consider looking at a building in the distance: properties beyond rough distance (e.g., its orientation, where the windows go in) cannot be seen by triangulation alone due to the quadratic growth in stereo error. On the other hand, we trivially see building orientation via the texture caused by repeated elements.

3.3 Other Properties

This dissertation is primarily about surface normals. In particular, this chapter has argued that surface normals should be a "first-class" representation that is modeled and estimated from images by themselves and that normals are particularly attractive for monocular perception. The main aim of this argument is to put the rest the idea that normals are merely the derivatives of estimated depth, not that normals somehow are the only form of 3D that is perceived. In fact, the very facts that support the use of normals as a representation, such as the existence of property-specific cues, support the use of multiple representations in a 3D understanding.

Qualitative Curvature: Some of Chapter 6's properties are based on curvature, drawing inspiration from [102]. The core idea is that the principal curvatures κ_1, κ_2 are cumbersome to work with on their own; thus the sign is used instead. We illustrate six possible combinations of curvature signs $+/0/-$ in Figure 3.6. Note that there are three other equivalent configurations, formed by taking reordering $+-$, $+0$ and -0. Additionally, note that the tubular structures $(+0/-0)$ could also be slanted, so long as the slant is constant.

Our work focuses on cases where one or more of the principal curvatures are zero. Those familiar with differential geometry or [102] may recall that these cases are mathematically degenerate in the sense that any deviation, however slight, will change the curvature class. This is correct, but due to the overwhelming dominance of planar and cylindrical objects in human-made scenes, these cases are important objects of study. Note that these sign curvatures correspond nicely with Biederman's geons [15]: the sign of a surface's curvature is directly related to the sign of the curvature of its occluding contour [99], even under perspective projection [130].

Depth: Finally, this chapter should not to be read as an argument *against* depth, but instead as an argument against depth to the exclusion of all other representations. In addition to ordinal measurements via occlusion boundaries (a different representation compared to metric depth), clearly some form of depth is perceivable in natural images beyond simply integrating surface orientation.

Chapter 4

Recovering 3D Structure via Mid-level Primitives

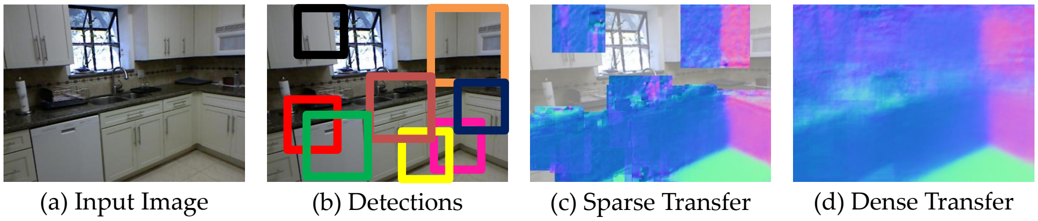

(a) Input Image (b) Detections (c) Sparse Transfer (d) Dense Transfer

Figure 4.1: We propose to find visually discriminative and geometrically informative primitives. Given an input image (a), these can be recognized (b) and convey the local geometry (c). We can use these sparse results to recover a dense interpretation via label transfer (d).

At the heart of the 3D inference problem is the question: what are the right primitives for inferring the 3D world from a 2D image? It is not clear what kind of 3D primitives can be directly detected in images and be used for subsequent 3D reasoning. There is a rich literature proposing a myriad of 3D primitives ranging from edges and surfaces to volumetric primitives such as generalized cylinders, geons and cuboids. While these 3D primitives make sense intuitively, they are often hard to detect because they are not discriminative in appearance. On the other hand, primitives based on appearance might be easy to detect but can be geometrically uninformative.

In this chapter, we propose data-driven geometric primitives which are **visually discriminative**, or easily recognized in a scene, and **geometrically informative**, or conveying information about the 3D world when recognized. Our primitives can correspond to geometric surfaces, corners of cuboids, intersection of planes, object parts or even whole objects. What defines them is their **discriminative** and **informative** properties. We formulate an objective function which encodes these two criteria and learn these 3D primitives from indoor RGBD data (see Fig. 4.2 for some examples of discovered primitives). We then demonstrate that our primitives can be recognized with high precision in RGB images (Fig. 4.1(b)) and convey a great deal of information about the underlying 3D world (Fig. 4.1(c)). We use these primitives to densely recover the surface normals of a scene from a single image via simple transfer (Fig. 4.1(d)). Our 3D primitives significantly outperform the state-of-the-art and generalize well across datasets.

4.1 Background

Early work focused on geometric primitives, for instance [24, 85, 154] which used 3D contours as primitives. Contours were first detected and the 3D world was inferred via a consistent line-labeling over the contours. While successful on line-drawings, these approaches failed to work on natural images: 3D contour detection is unquestionably difficult. Other research focused on volumetric 3D primitives such as generalized cylinders [16] and geons [15]. However, although these primitives produced impressive demos such as ACRONYM [20], they failed to generalize well.

Beginning with [79] and [143], there was a push towards learning-based approaches. The most commonly used primitives include oriented 3D surfaces [79, 112, 143, 166] represented as segments in the image, or volumetric primitives such as blocks [63] and cuboids [75, 111, 167]. However, since these primitives are not discriminative, a global consistency must be enforced, e.g., by a learned model such as in [143], a hierarchical segmentation [79], physical and volumetric relationships [111], recognizing primitives as parts of semantic objects [166], or assuming a Manhattan world and low-parameter room layout model [30, 74, 148, 171]. Non-parametric approaches provide an alternative approach to incorporating global constraints, and instead use patch-to-patch [72], scene [98], or 2D-3D [140] matching to obtain nearest neighbors followed by label transfer. While all of these constraint-based approaches have improved 3D scene understanding, accurate detection of primitives still remains a major challenge.

On the other hand, there have been recent advances in the field of detection enabled by discriminative appearance-based approaches. Specifically, instead of using manually defined and semantically meaningful primitives or parts, these approaches discover primitives in labeled [19, 40], weakly-labeled [36] or unlabeled data [152]. While these primitives have high detection accuracy, they might not have consistent underlying geometry. Building upon these advances, our work discovers primitives that are both **discriminative** and **geometrically informative**. One exception is the contemporary work of Owens et al. [127], which found discriminative patches in the context of a different 3D domain – multi-view stereo. Unsurprisingly, many of their discovered primitives are similar to ours.

4.2 Approach

4.2.1 Learning

Given a set of training images and their corresponding surface normals, our goal is to discover geometric primitives that are both discriminative and informative. The challenge is that the space of geometric primitives is enormous, and we must sift through all the data to find geometrically-consistent and visually discriminative concepts. Similar to object discovery approaches, we pose it as a clustering problem: given millions of image patches, we group them so each cluster is discriminative (we can learn an accurate detector) and geometrically consistent (all patches in the cluster have consistent surface normals).

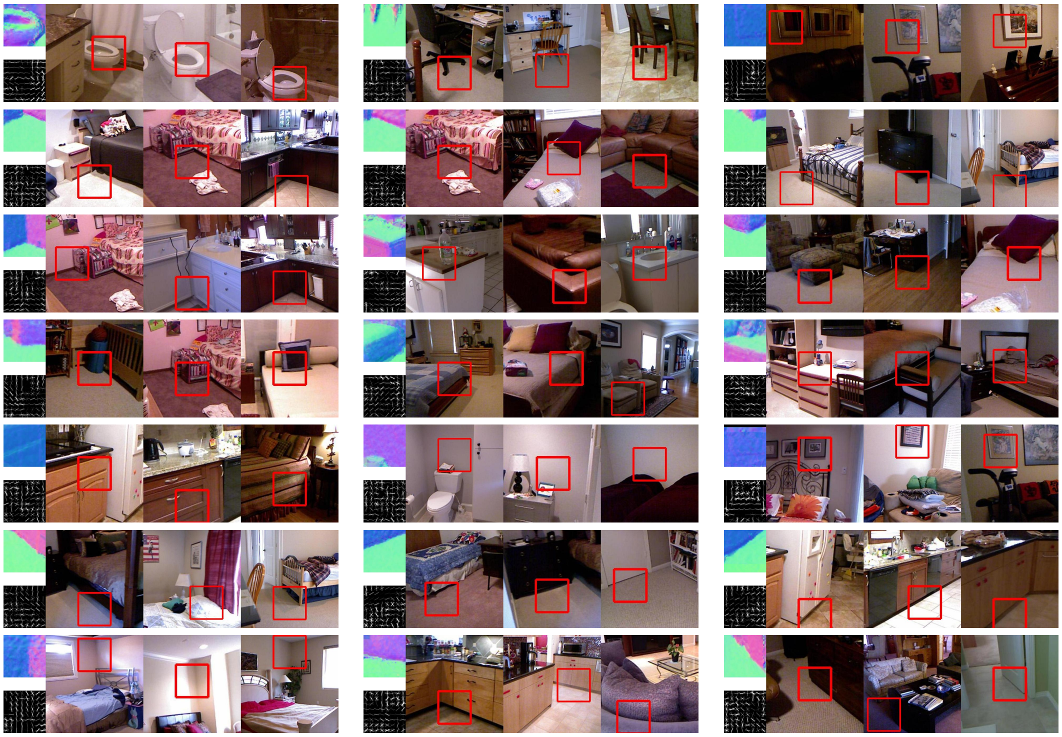


Figure 4.2: Example primitives discovered by our approach. (Left: canonical normals and detectors; right: patch instances). Our discovery method finds a wide range of primitives: **Row 1:** objects and parts, **Rows 2-4:** 3 and more plane primitives, **Row 5:** 1 plane, **Row 6-7:** 2 planes. Our primitives contain visual synonyms, (e.g., row 6), in which different appearance corresponds to the same underlying geometry.

Mathematically, we formulate the problem as follows. As input, we have a collection of image patches in the training dataset, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. Each patch has a geometric component $\mathbf{x}_i^G$ (a 2D array of surface normals scaled to a canonical scale) and appearance representation $\mathbf{x}_i^A$ (HOG). Our goal is to cluster the data and learn geometric primitives. Each primitive is represented as $\langle \mathbf{w}, \mathbf{N}, \mathbf{y} \rangle$ where $\mathbf{w}$ is the weight vector of a linear support vector machine (SVM) learned in appearance space, $\mathbf{N}$ represents the underlying geometry of the primitive and $\mathbf{y} \in \{0,1\}^m$ is an instance indicator vector with $y_i = 1$ for instances of the primitive and zero otherwise. Ideally, we would like to minimize the following objective function:

$$\min_{\mathbf{y},\mathbf{w},\mathbf{N}} R(\mathbf{w}) + \sum_{i=1}^{m} c_1 y_i \Delta(\mathbf{N}, \mathbf{x}_i^G) + c_2 L(\mathbf{w}, \mathbf{x}_i^A, y_i), \quad s.t. \quad |\mathbf{y}|_1 = s \tag{4.1}$$

where R is a regularizer on the classifier, each c_i trades off between terms, Δ is a distance measure over 3D geometry, and L is a loss function. To avoid trivial solutions, we also constrain the membership so each cluster has at least s members.

The first term regularizes the primitive's detector, the second enforces consistent geometry across instances so that the primitive is geometrically informative, and the third ensures that the clusters' instances can be distinguished from other patches (that is, the loss function over the correct classification of training patches). In our case, we use an SVM-based detector and represent geometry with surface normals. Therefore, we set $R(\mathbf{w})$ to $||\mathbf{w}||_2^2$, Δ to the mean per-pixel cosine distance between patches' surface normals, and L to hinge-loss on each $\mathbf{x}_i^A$ with respect to $\mathbf{w}$ and $\mathbf{y}$. Note that there will be many local minima, with each minimum $\mathbf{w}, \mathbf{y}, \mathbf{N}$ corresponding to a single primitive: as shown in Fig. 4.2, many 3D primitives are discriminative and informative. We obtain a collection of primitives by finding the many minima, which we do via multiple initialization.

Exactly minimizing this objective function is an extremely difficult problem, but its optimization is a chicken-and-egg problem: if we knew a good set of geometrically-consistent and visually discriminative patches, we could train a detector to find them, and if we had a detector for a geometrically consistent visual concept, we could find instances. We propose an iterative solution for the optimization. Given $\mathbf{y}$, we can compute $\mathbf{N}$ and learn $\mathbf{w}$. Once we have $\mathbf{w}$, we can refine our membership $\mathbf{y}$ and repeat.

We therefore minimize Eqn. 4.1 in the style of block coordinate descent: cycling through the parts of our representation ($\langle \mathbf{w}, \mathbf{N}, \mathbf{y} \rangle$), we optimize one part while holding the others fixed. Pragmatically, this entails: given an initialization in terms of membership $\mathbf{y}$, we train a detector $\mathbf{w}$ to separate the elements of the cluster from geometrically dissimilar patches from negative examples $\mathcal{V}$ (found via the canonical form $\mathbf{N}$). In this work, we also supplement our negative data with 6000 randomly chosen outdoor images $\mathcal{W}$ from [73]. Once a detector has been trained, we scan our training set $\mathcal{I}$ for the top most-similar patches to $\mathbf{w}$, and use these patches to update membership $\mathbf{y}$.

Minimizing w.r.t. N: given the primitive instances, we compute the canonical form $\mathbf{N}$ of the current instances as the per-pixel average over the instances. Note that we represent each geometric form or surface normal patch at a canonical scale.

Minimizing w.r.t. w: given the primitive instances and a canonical form, we want to train a detector that can distinguish the primitive instances from the rest of the world. We then train a linear SVM $\mathbf{w}$ to separate the positive instances from geometrically dissimilar patches in the negative set $\mathcal{V}$ and all patches in $\mathcal{W}$. We define geometrically dissimilar patches as patches with mean per-pixel cosine distance from $\mathbf{N}$ greater than $70°$. We use a high-threshold to include definite mistakes only and promote generalization.

Minimizing w.r.t. y: Given $\mathbf{w}$ is found, we pick $\mathbf{y}$ by selecting a geometrically consistent set among the top detections of $\mathbf{w}$ in $\mathcal{I}$; in our experiments, we use the s-member subset that approximately minimizes the intra-set cosine distance.

Initialization: We initialize the method with $\mathbf{y}$. We initialize our algorithm with a greedy approach to find independently visually and geometrically compact groups in randomly sampled patches. First, we sample random square patches throughout the training set $\mathcal{I}$ at multiple scales and compute their appearance and geometric forms. For every patch, we find $s-1$ nearest neighbors in appearance and geometry space by intersecting the nearest neighbor lists (computed with Euclidean distance on HOG and mean per-pixel cosine distance respectively). We group the query patch with its neighbors to initialize the primitive instances. For a training set of 800 images, we produce $\approx 3,000$ primitive candidates.

If done directly, this sort of iterative technique (akin to discriminative clustering [170]) has been demonstrated to overfit and produce sub-optimal results. We therefore adopt the cross-validation technique of [36,152] in which the datasets are divided into partitions. We use two partitions; we initialize identities $\mathbf{y}$ and train the detector $\mathbf{w}$ on partition 1; then we update the identities $\mathbf{y}$ and train $\mathbf{w}$ on partition 2; following this, we return to partition 1, etc. This can be thought of as a constraint on the membership variable that changes from

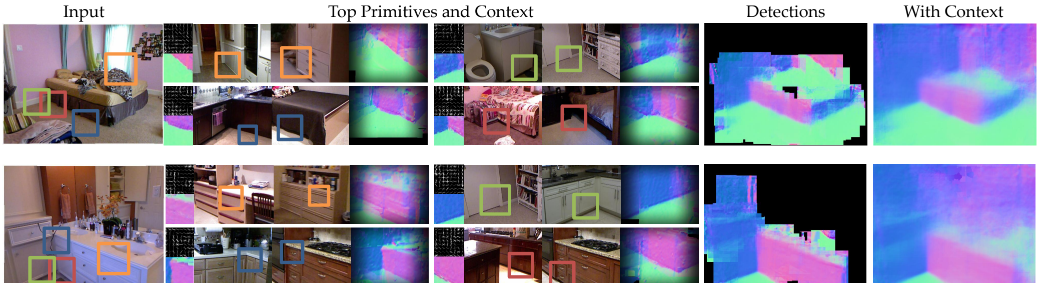


Figure 4.3: Qualitative Results: The above figure shows two examples of how our primitives help us in accurate 3D interpretation of images in both the sparse and dense setting.

iteration to iteration.

4.2.2 Inference

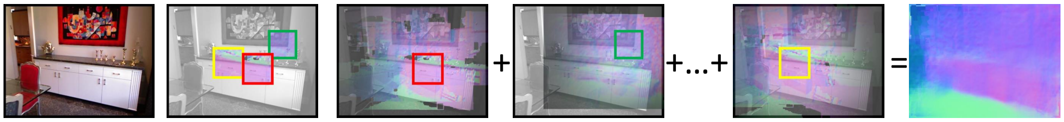
Our approach extracts geometrically consistent and visually discriminative primitives from RGBD data. These primitives can then be detected with high precision and accuracy in new RGB images to develop a 3D interpretation of the image. To demonstrate the effectiveness of our primitives, we first try a simple label-transfer method to propagate labels, which aims to produce the most likely 3D explanation for each pixel independently. This initial investigation of a simple method is deliberate: if we filter the primitives through a complex inference method, we cannot separate gains due to the primitive mining approach and gains due to the inference technique. Nonetheless, despite its simplicity, we show that the method credibly outperforms the state of the art in Section 4.3. In Chapter 7, we present an alternate approach that reasons about the joint probability of pixels, and obtains further gains.

Our procedures are illustrated in Figs. 4.4 and 4.3 along with sample results. In the case of Fig. 4.4, most parts of the scene (like the painting and bookshelf) are easy to parse with our method: we expect that at least some detectors would have high responses on them. Moreover, if we were to parse just these regions, this would be satisfactory for many tasks. However, if we want to predict every single pixel in every single image as in other vision tasks, some other parts of the image (such as the blank wall) are hopelessly difficult to parse from local evidence alone. To handle these, we seek a dense interpretation that explains these image regions via label propagation. In our case, our label propagation happens by transferring not only the primitives, but also their context from training time.

Technical Approach: Given a test image, we run the bank of the 3D primitives' learned detectors $\mathbf{w}$. This yields a collection of T detections $(\mathbf{d}_1 \ldots \mathbf{d}_T)$. We warp the surface normals so the primitive detections and s training instances per detection are aligned, producing a collection of sT aligned surface normal images $(\mathbf{M}_{1,1} \ldots \mathbf{M}_{1,T} \ldots, \mathbf{M}_{s,T})$. We infer the pixels of a test image as a linear combination of the surface normals of these aligned training images with weights determined by detections:

$$\hat{M}(\mathbf{p}) = \frac{1}{Z} \sum_{i,j=1,1}^{s,T} \text{score}(\mathbf{d}) \exp(-||\mathbf{p} - \mathbf{d}_j||^2/\sigma^2_{\text{spatial}}) \mathbf{M}_{i,j}(\mathbf{p}),$$

where Z is a normalization term. The first term gives high weight to confident detections and to detections that fire consistently at the same absolute location in training and test



(a) Input (b) Sparse Detections (c) Primitive-aligned Label Transfer (d) Final Result

Figure 4.4: An illustration of our simple inference approach. We align the training images via detections and predict the test image as a weighted average of the training images.

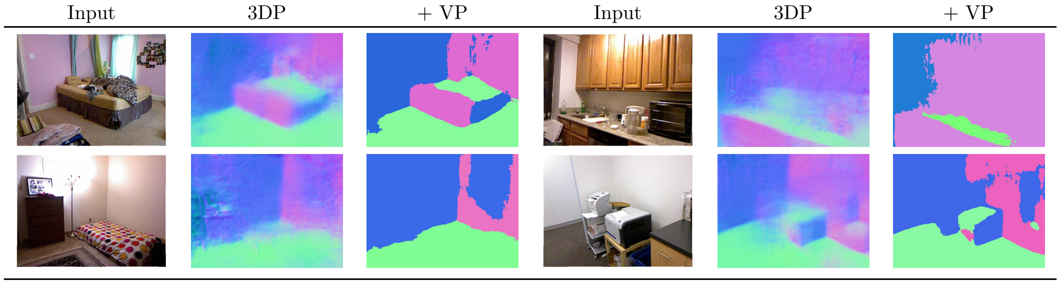


Figure 4.5: Qualitative results of the Manhattan-world adjustments. Top-row: successful results. Bottom-row: failure cases; note the blobby artifacts in all examples and implausible changes in normals mid-wall. These are caused by inferences that are close-to-equidistant to two vanishing points and thus results are adjusted arbitrarily.

image (e.g., floor primitives should not be at the top of an image). The second term is the spatial term and gives high weight for transfer to pixels near the detection and the weight decreases as a function of the distance from the location of the primitive detection. Each direction is processed separately and the resulting vector re-normalized, corresponding to a maximum-likelihood fit of a von Mises-Fisher distribution.

Incorporating Vanishing Points: Indoor human scenes generally are dominated by three mutually orthogonal directions. This assumption, known as the Manhattan-World assumption, has been enormously successful in indoor understanding tasks. We add this assumption to our method as follows. We first estimate three orthogonal vanishing points via the method of Hedau et al. [74], which in turn is a modification of the method of Rother [137]. We then compute the normal associated with each vanishing point, yielding three normals $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ as well as three negations $-\mathbf{n}_1, -\mathbf{n}_2, -\mathbf{n}_3$ (e.g., if $\mathbf{n}_1$ is a left wall, $-\mathbf{n}_1$ is a right wall). We then independently snap each predicted normal to the nearest vanishing-point normal. This approach and its limitations is illustrated in Fig. 4.5, and we introduce a more constrained decoding approach in Chapter 7.

4.3 Experiments

We now describe the experimental setup used to validate our approach to estimating local orientation as well as our results.

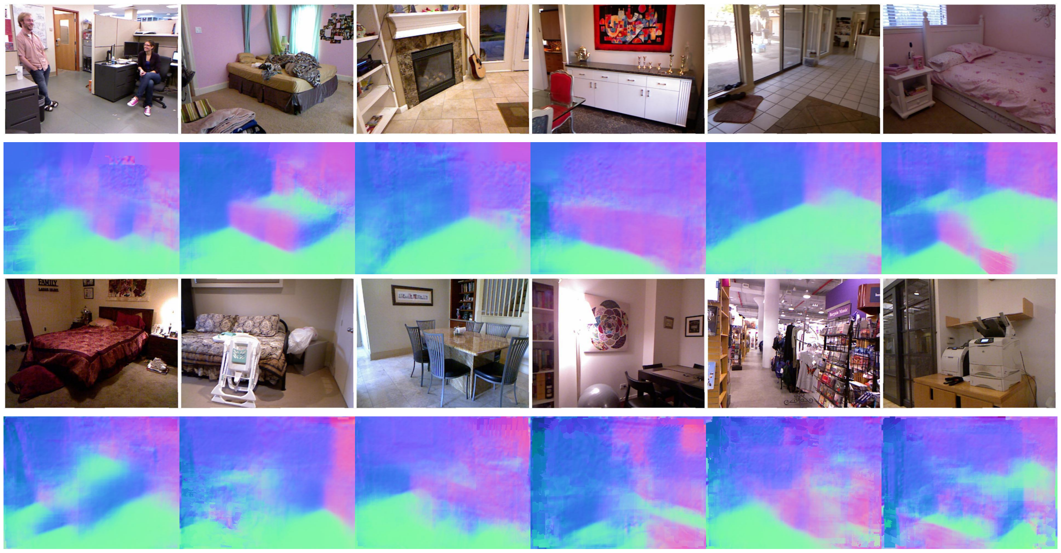


Figure 4.6: Example normal maps inferred via our data-driven 3D primitives. **Top row:** selected results; **bottom row:** results automatically regularly sampled to show the (left-to-right/good-to-bad) range of the proposed method from $1/7$th to $6/7$th percentile.

4.3.1 Datasets

We primarily evaluate our approach on the NYU Depth v2 [150] dataset (NYUv2). This dataset contains $1,449$ registered RGB and depth images from a wide variety of real-world and cluttered scenes. We ignore values for which we cannot obtain an accurate estimate of the surface normals due to missing depth data. To facilitate comparison with other methods, we report results for the train-test splits used in [150], which introduced the dataset.

4.3.2 Baselines

We compare against the state of the art in a variety of different related tasks to demonstrate the utility of our approach. Specifically, we compare against eight baselines in sparse and dense prediction. The first five test against a variety of alternate existing methods for computing normals either directly or indirectly. The sixth tests the contribution of geometric supervision. The last two test against standard basic approaches for labeling images.
(1) Lee et al. [112]: The plane-sweeping orientation map finds sparse vanishing-point aligned planar surfaces from lines. For dense predictions, we use NN-based filling.
(2) Hoiem et al. [79]: The geometric context approach predicts quantized surface normals in five directions using multiple-segmentation based classifiers. We retrain the classifiers using the NYU data by assigning each pixel to its nearest quantized normal. At test time, we take the label with maximum posterior probability at each pixel. We tried two quantizations for the left/right vertical classes, $[\pm 1, 0, 0]^T$ and $[\pm\sqrt{2}/2, 0, \sqrt{2}/2]^T$, and used the quantization with better performance ($[\pm 1, 0, 0]$).
(3) Hedau et al. [74]: This baseline builds on geometric context classifiers (including one for clutter), which we retrain on NYU using identical settings as (2) and uses structured prediction to predict a vanishing-point-aligned room cuboid.

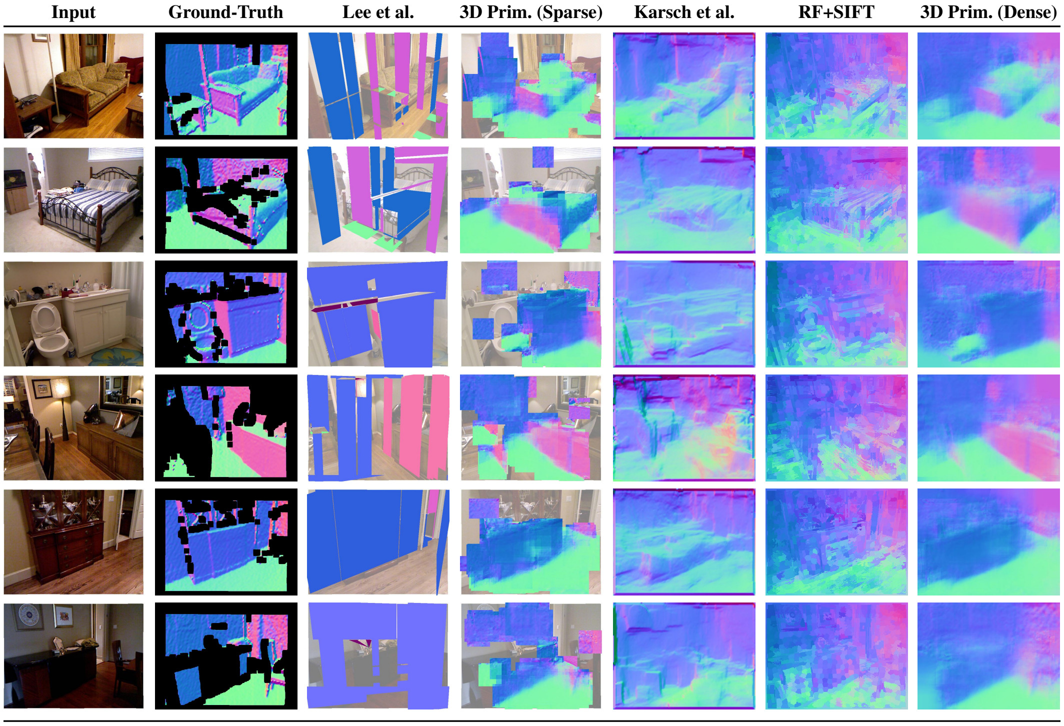


Figure 4.7: Qualitative results of our technique in comparison to some of the baselines.

(4) Karsch et al. [98]: One can also produce surface normals by predicting depth and computing normals on the results; we do this with the depth prediction method of Karsch et al. [98], which out-performs other methods for depth prediction by a considerable margin in all evaluation criteria.

(5) Saxena et al. [143]: We also compare with surface normals computed from depth predicted by Make 3D using the pre-trained model.

(6) Singh et al. [152]: We compare against this appearance based primitive discovery approach to test whether our geometric supervision adds anything. We replace our geometric primitives with unsupervised mid-level patches discovered by [152], and use the same inference pipeline.

(7) RF + SIFT: We train a random forest (RF) regressor to predict surface normals coordinate-wise inside superpixels, followed by re-normalizing. As features, we use a histogram of dense-SIFT [118] features over the superpixel (codebook size 1K, hard-assignment) as well as the superpixel's centroid. To generate superpixels, we use over SLIC [1] superpixels ($S = 20$, $M = 100$). We compute both via VLFeat [159]. We additionally experimented with a standard LM filter bank [114], which gave poorer results.

(8) SVR + SIFT: We follow the same pipeline as above, but train an ϵ-Support Vector Regressor (SVR) using a Hellinger kernel, implemented by direct feature map. Since the Hellinger kernel is designed for histograms, we represent the superpixel's location as a histogram, by binning the pixels' locations into 100 bins, subdividing the image into 10 bins in X and Y.

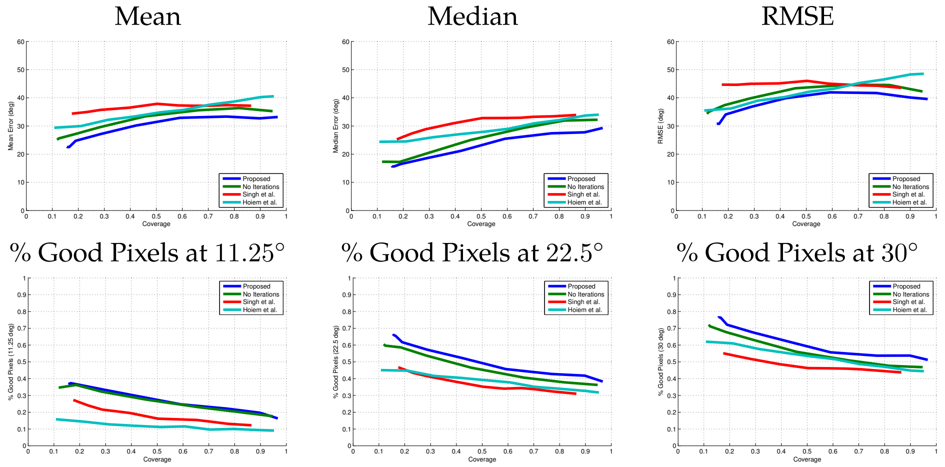


Figure 4.8: Performance vs. Coverage for 3D Primitives and some baselines

4.3.3 Qualitative Results

We first discuss our method qualitatively. We show our results in comparison to the state of the art in Fig. 4.7, and selected results as well as evenly-spaced by-performance results in Fig. 4.6. Our method can accurately convey the 3D structure of scenes, including objects such as filing cabinets even when these details deviate from the typical structure of a scene (e.g., Fig. 4.6 upper-left). Our method, however, does not register any objects smaller than a filing cabinet; handling the single-image 3D estimation of these small objects in such scenes is an open question. Additionally, note that our results accurately capture inter-object boundaries (note the floor-object boundaries) even though we do not use a segmentation.

Note that the dense results not only fill in the missing parts of the scene, but also smooths away some incorrect detections. Note for instance, the upwards-facing detection on the left of the last row of Fig. 4.7 that is smoothed in the dense prediction version via nearby context transfers. As expected, this can result in oversmoothed results as well.

4.3.4 Quantitative Results

We now evaluate our method quantitatively. To evaluate our sparse results, we compute performance-coverage curves which quantify the trade-off between performance (prediction accuracy) and coverage (the fraction of the dataset that is predicted). These are the akin to the purity-coverage graphs of Doersch et al. [34] used for weakly-labeled element discovery methods. To evaluate dense results, we compute per-pixel accuracies.

To better put the results in context, we analyzed the noise inherent in our dataset. In addition to helping characterize the results, this provides a lower bound on the methods used. We annotated a randomly chosen set of planar surfaces in 100 images by picking points randomly and then annotating the plane they lie on if the region is planar. Each plane should have a single normal; we quantify the degree to which this is true by computing the angle between the points on the plane. The median angular error is $5.2°$.

Sparse results. We show performance-coverage curves for our method compared to a number of baselines in Fig. 4.8. Specifically, we compare against Singh et al. [152], Hoiem et

Table 4.1: Results training on the NYU Depth v2 dataset using the standard train-test split.

	Summary Stats. (°)			% Good Pixels		
	(Lower Better)			(Higher Better)		
	Mean	Median	RMSE	11.25°	22.5°	30°
	With Manhattan World Constraints					
Lee et al.	43.3	36.3	54.6	25.1	40.4	46.1
Hedau et al.	40.0	23.5	54.1	34.2	49.3	54.4
3D Primitives.	**36.0**	**20.5**	**49.4**	**35.9**	**52.0**	**57.8**
	Without Manhattan World Constraints					
Karsch et al.	40.7	37.8	46.9	8.1	25.9	38.2
Hoiem et al.	41.2	35.1	49.2	9.0	31.2	43.5
Saxena et al.	48.0	43.1	57.0	10.7	27.0	36.3
RF + SIFT	36.0	33.4	41.7	11.4	31.4	44.5
SVR + SIFT	36.6	33.6	42.5	10.6	30.6	44.0
3D Primitives	**34.2**	**30.0**	**41.4**	**18.6**	**38.6**	**49.9**

al. [82], as well as only one round of the iterative procedure to test whether the primitives improve with iterations. To rank pixels in patch-based methods we use the normalizer Z in Eqn. 4.2; intuitively, this is the sum of the calibrated detector scores. To rank pixels for Geometric Context [82], we sweep over classifier confidence, calculated as the posterior probability of the most confident label.

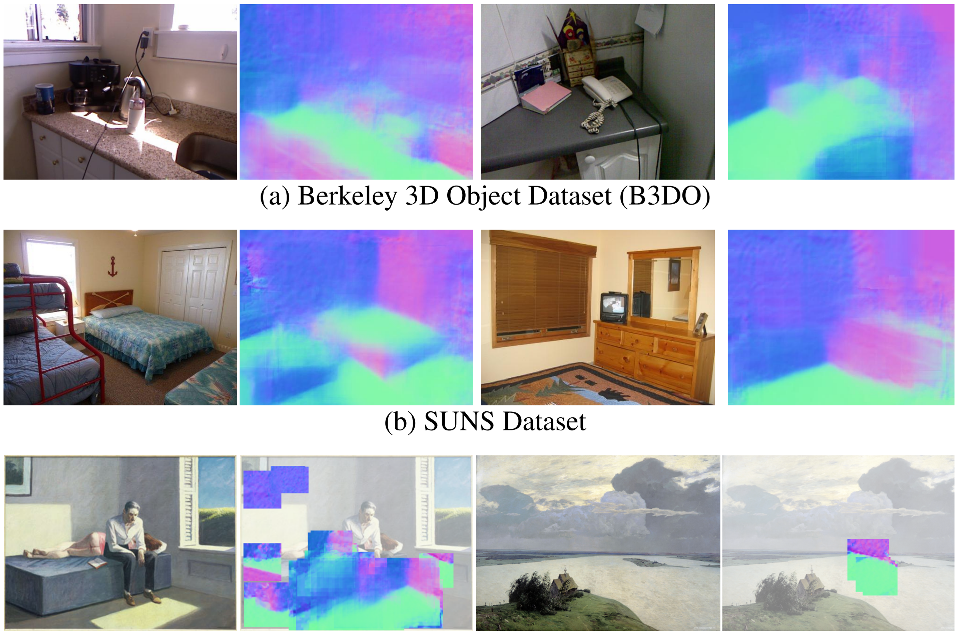
Our approach outperforms the baselines and the initialization at every coverage level. The gains are particularly pronounced in the high-precision low-coverage regime. Note that our method does not reach 100% coverage; the remaining unpredicted pixels correspond to regions that cannot be predicted accurately from local evidence (i.e., for which all primitives have an extremely low response) thus necessitating our context-transfer technique. The most confident results are substantially more accurate than the rest of the predictions. This ability of our method to identify its most accurate predictions enables it to pass its confidence to applications using 3D as an input as a valuable cue for subsequent reasoning.

Dense results. We summarize our comparison with the baselines in Table 4.1. We separately evaluate approaches making the Manhattan-world assumption and those not making it; we analyze how making this assumption changes results in the subsequent section. Our approach outperforms the baselines in each category (Manhattan-world/non-Manhattan-world) in all metrics. As discussed, this is important as each metric captures a different aspect of performance. Hedau et al.'s method, for instance does well on median error (as many of its predictions are completely correct as it uses the vanishing points) but does poorly on RMSE (as many of its predictions are completely incorrect).

4.3.5 Cross Dataset Performance

We now discuss evaluations done across datasets in which we take a model learned on NYU and apply it to another dataset without any adjustments or tuning.

Qualitative Evaluations. We show results on a number of datasets in Fig. 4.9. Note that some scenes are atypical of layout prediction datasets, especially NYU [150]: note the close-up views in B3DO, for instance. Our method accurately analyzes the images. Note that



(a) Berkeley 3D Object Dataset (B3DO)

(b) SUNS Dataset

Excursion into Philosophy
Edward Hopper, 1959

Above Eternal Peace
Isaac Levitan, 1894

(c) Art

Figure 4.9: Cross-dataset results on B3DO (top), SUNS (middle), and artwork (bottom).

for instance, in the case of *Above Eternal Peace*, the correct answer is not to predict: thanks to its random Flickr negative set, our method identifies that most of *Above Eternal Peace* corresponds to none of its primitives.

Quantitative Evaluations. Most of the datasets on which we test our method do not come with corresponding ground truth normals or depth. We therefore also evaluate on the B3DO dataset [88]. Our method again outperforms other methods on B3DO. Additionally, despite a substantial domain shift (B3DO has substantially more close-up views as well as tilted cameras, for instance), our method's performance loss is small.

4.4 Outdoors

As a demonstration of the generality of our techniques, we report some qualitative results on the KITTI dataset [59], which consists of street scenes collected with both conventional cameras and a LIDAR scanner. We process the point-clouds to produce per-point normals, reproject the point-cloud into the image and interpolate. We downsample the video dataset to 1165 frames and hold out approximately 25% of the data for evaluation. Some sample frames are shown in Fig. 4.10. A few things can be noted from these frames: there are many more horizontal surfaces compared to indoors; other than this ground-plane, there are many fewer planar surfaces, as detectable by the decreased number of single-color regions.

We run the learning procedure without modification. Fig. 4.11 shows some examples of

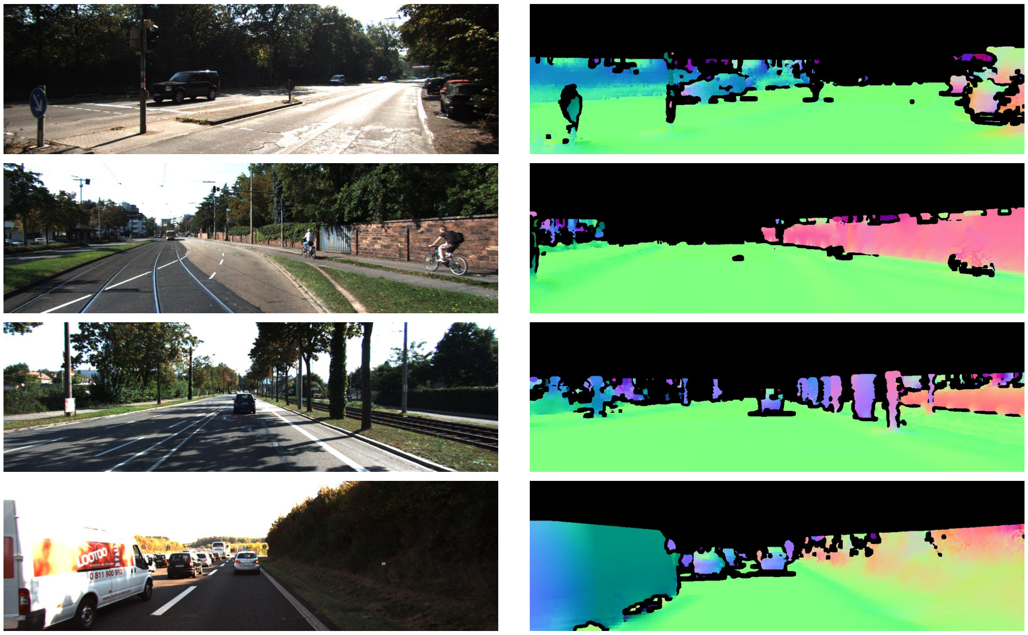


Figure 4.10: Sample data from KITTI

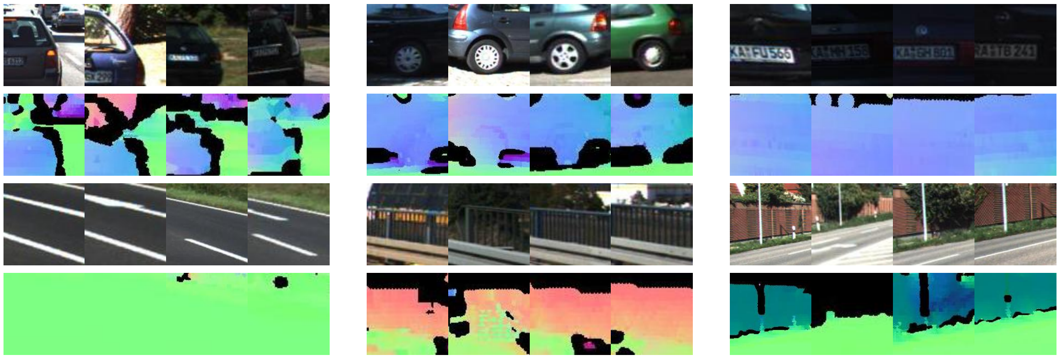


Figure 4.11: Primitives discovered outdoors. **Row 1:** car parts; **row 2:** road markings and objects on the side of the road.

the primitives that are learned by the approach. The approach learns to pick up on things like car parts, license plates, and lane markings. This is because these are frequent, easy to detect, and tend to be informative of geometry. For instance, license plates are heavily textured and only occur on the side of the car that is facing the camera.

We show results of the inference method in Fig. 4.12. To help better visualize the results, we disable fully horizontal patches. KITTI is dominated by horizontal surfaces and thus the large bulk of the patches learned are horizontal; these tend to wash out the more interesting patches. The method works well, extracting the sides of the road and many of the vehicles. One failure that is apparent when looking at the cars is that the method handles them by

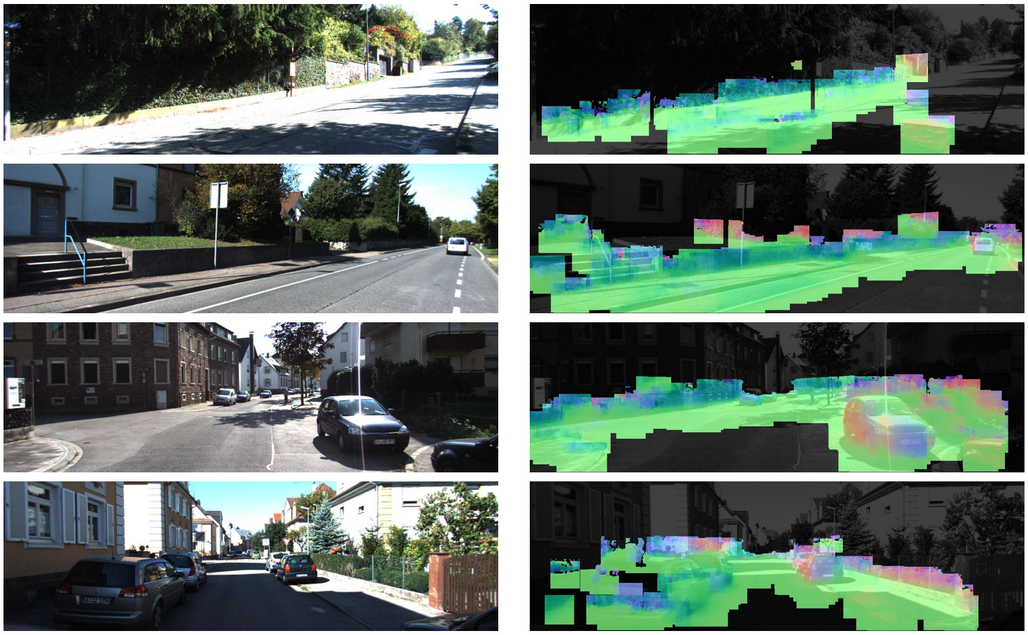


Figure 4.12: Results outdoor. Rows 1, 2: random results; Rows 3, 4: selected.

getting the discriminative parts of the car correct and interpolating in between. Similarly, although frequently the concave corners are correctly localized, their scale is sometimes incorrect: note the incorrect extent predicted on the car back in the lower left corner of row 4. Similarly, KITTI contains few convex edges, and thus these are poorly represented in the data.

4.5 Discussion

This chapter introduced a technique for predicting surface normals from images. The approach is based on the idea that certain combinations of 3D structure and style are obvious in images. The inference approach was relatively simple yet produced good interpretations of new images both indoors and outdoors.

We address two of the approach's short-comings of the approach in subsequent chapters. The first is the issue of inefficient use of the data: the data-driven primitives model combinations of 3D structure and style. This means that data is not shared across viewpoints and that one has to see a particular combination of 3D structure and style to recognize it. Chapter 5 addresses this with style elements, which operate *across* viewpoints. The second is that there is no reasoning involved. We address this in Chapter 7 with an approach that jointly optimizes over the entire scene to try to ensure coherency.

Chapter 5

3D Structure from Unsupervised Style/Structure Factorization

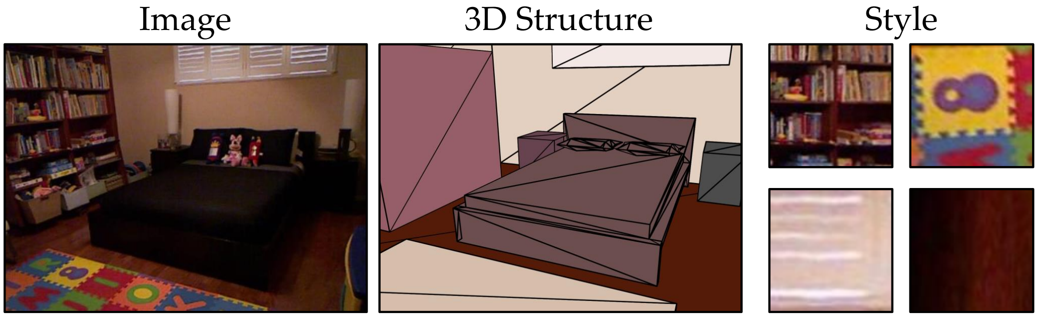


Figure 5.1: How can we learn to understand images in a 3D way? In this chapter, we show a way to do this without using a single 3D label using our 3D structure and style factorization In this chapter, we learn *style elements* that recognize texture (e.g., bookshelves, tile floors) rectified to a canonical view. Rather than use explicit supervision, we use the regularity of indoor scenes and a hypothesize-and-verify approach. We thus learn models for single image 3D without seeing a single explicit 3D label. 3D model from [62].

Consider the image in Fig. 5.1. When we see this image, we can easily recognize and compensate for the underlying 3D structure: for example, we have no trouble recognizing the orientation of the bookshelves and the floor. But how can computers do this? The traditional answer is to follow supervised approaches like the one we took in the past chapter: simply collect large amounts of labeled data to learn a mapping from RGB to 3D. In theory, this is mathematically impossible, but the argument is that there is sufficient regularity to learn the mapping from data. In this chapter, we take this argument one step further: we claim that there is enough regularity in indoor scenes to learn a model for 3D scene understanding without ever seeing an explicit 3D label.

At the heart of our approach is the central observation of this dissertation: images are a

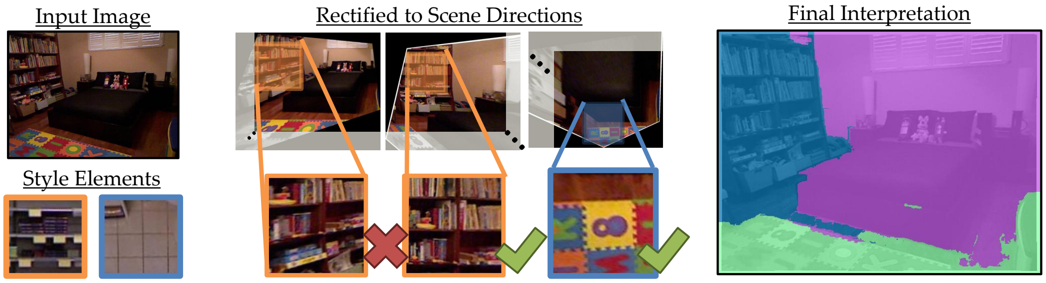


Figure 5.2: We infer a 3D interpretation of a new scene with style elements by detecting them in the input image rectified to the main directions of the scene. For instance, our bookshelf style-element (orange) will respond well to the bookshelf when it is rectified with the correct direction (facing leftwards) and poorly when it is not. We show how we can automatically learn these style elements, and thus a model for 3D scene understanding without any 3D supervision. Instead, the regularity of the world acts as the supervisory signal.

product of two separate phenomena, 3D structure and style. Based on this observation, we propose **style elements** as a basic unit of 3D inference. Style elements detect the presence of style, or texture that is correctly rectified to a canonical fronto-parallel view. They include things like cabinets, window-blinds, and tile floors. We use these style elements to recognize when a texture has been rectified to fronto-parallel correctly. This lets us recognize the orientation of the scene in a hypothesize-and-verify framework: for instance, if we warp the bookshelf in Fig. 5.2 to look as if it is facing right, our rectified bookshelf detector will respond strongly; if we warp it to look as if it is facing left, our rectified bookshelf detector will respond poorly.

In this chapter, we show that we can learn these style elements in an unsupervised manner by leveraging the regularity of the world's 3D structure. The key assumption of our approach is that we expect the structure of indoor scenes to resemble an inside-out-box *on average*: on the left of the image, surfaces should face right and in the middle, they should face us. In other words, we can put a prior on 3D structure over the entire dataset. We show how this prior belief can validate style elements in a hypothesize-and-verify approach: we propose a style element and check how well its detections match this belief about 3D structure over a large set of unlabeled images; if an element's detections substantially mismatch, our hypothesis was probably wrong. To the best of our knowledge, this is the unsupervised learning-based approach for 3D scene understanding from a single image.

Why unsupervised? We wish to show that unsupervised 3D learning can be effective for predicting 3D. We do so on two datasets: NYUv2, a standard 3D dataset, and Places-205, which contains scenes not covered by Kinect datasets, such as supermarkets and airports. Even though our approach is unsupervised: (1) Our method nearly matches comparable supervised approaches on NYUv2: it is within $< 3^\circ$ of 3DP [51] and better in many metrics on vertical regions. (2) When fused with 3DP, our method achieves state-of-the-art results in 4/6 metrics on NYUv2. (3) As an unsupervised approach, our method can learn from unlabeled Internet images like Places. This is fundamentally impossible for supervised methods, which must resort to pre-trained models and suffer performance loss from the domain shift. Our approach can use this data and outperforms 3DP by 3.7%.

Why Style Elements? Operating in this style space lets us learn about the world in a viewpoint-independent fashion. In this chapter, we show how this enables us to learn unsupervised models for 3D, but we see broader advantages to this: first, we can detect particular combinations of style and structure that were not present at training time, which is impossible in many existing models; second, since our style elements are viewpoint-independent, we can share information across different viewpoints. We illustrate these advantages in Fig. 5.3: our method learns one element for all the orientations of the cabinets, but a standard viewer-centric approach learns one element per orientation.

5.1 Background

The notion of style elements is related to a long line of work on learning-based 3D structure recovery as well as mid-level elements, but deviates in a number of key ways. In doing so, it shares much in common with texture-based approaches that exploit regularity within an image or which recover geometry by trying to undo perspective distortion. We briefly review the salient differences before presenting the method.

The approach presented in this chapter is a departure from past learning-based approaches for two reasons. The first and most immediately striking one is that it does not require an explicit supervisory signal: instead of using Kinect or LIDAR or some other physical sensor, it learns its model from the regularity present in the data. But the perhaps more interesting difference is that its representation is view-independent as it uses a canonical representation, unlike all previous work on learned approaches (to the best of our knowledge). In principle, this lets it generalize knowledge across viewpoints: what it has learned about left-wards-facing televisions applies to right-wards-facing televisions. The most similar work is [108], which uses canonicalized depths.

This canonicalization approach thus then shares more with optimization-based recovery of 3D properties. In the past, the idea of figuring out the 3D structure by optimizing properties of the unwarped image has been used in shape-from-texture (e.g., [47,120]) and modern texture analysis [174] and compression [161] approaches. These works are complementary to our own: many obtain a detailed interpretation on presegmented regions or in specific domains by optimizing some criterion such as regularity within one image or a single domain. Our style elements on the other hand, are discovered automatically via the regularity in large amounts of data, and are more general than instance-level texture patterns. They can further interpret novel, generic non-presegmented scenes, although our interpretations on these cluttered scenes are more coarse in comparison.

Apart from 3D estimation, canonicalization has been used has been in recognition. Warping images to a canonical view has been used to boost performance of local patch descriptors for tasks like location recognition [153,165], in which 3D structure is known or estimated via pre-trained models, or in detection [46,75], in which it is given at training time. Our work, on the other hand, is unsupervised and jointly reasons about 3D structure (i.e., how the texture is being canonicalized) and style (the final result).

Our approach uses visual elements discovered from a large dataset and draws from a rich literature on discriminative patch-discovery [19,35,36,93,152]. Like Juneja et al. [93], we take a hypothesize-and-verify approach which filters a large set of candidate patch hypotheses by patch detections on a dataset. Among these works, our work is most closely related to discriminative patches for 3D [51,127] or visual-style-sensitive patches [113]. The primary difference, however, capture the Cartesian product of appearance and the label (style or 3D), meaning that for these frameworks to capture an oven-front at a particular

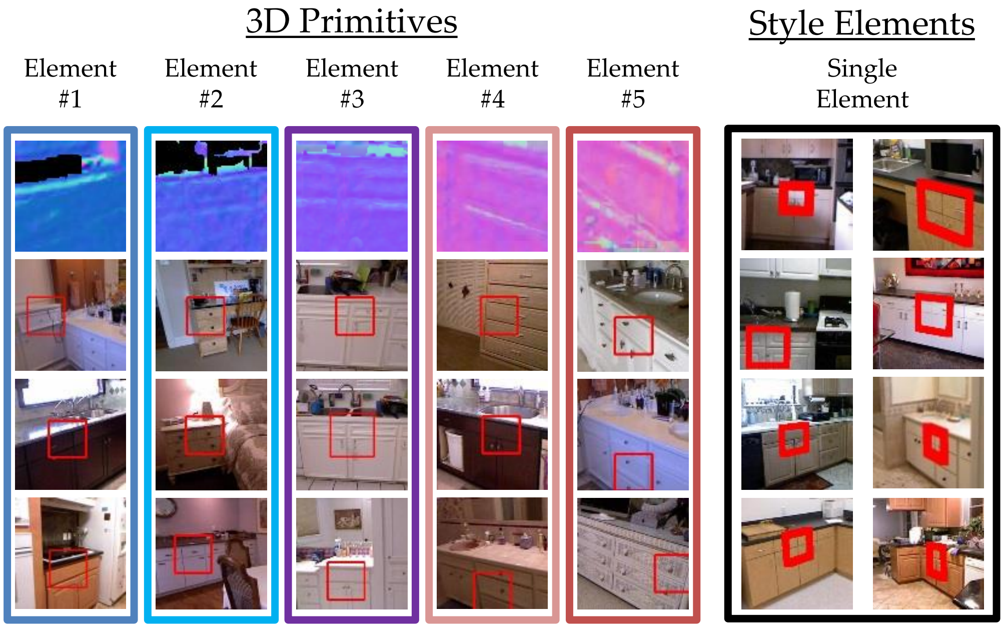


Figure 5.3: An illustration of sharing enabled by style elements. **Left:** Five 3DP models, all of which represent cabinets, but at different views. **Right:** Top detections from a single style element model. 3DP detects *each and every* cabinet orientation with a separate element because it is viewer-centric; representing the cabinet in terms of style compactly recognizes *all* cabinet orientations with one element.

angle, they need to see an oven-front at that particular angle. On the other hand, our approach analytically compensates for 3D structure by rectifying the image data. Thus our elements can predict labels not seen at training time (e.g., an oven at a previously unseen angle). We illustrate this in Fig. 5.3.

5.2 Overview

Given a dictionary of discovered style elements, we can use this dictionary of detectors in rectified images to determine the orientation of surfaces: the elements only respond when the scene is rectified correctly. But how do we obtain this dictionary of correctly rectified style elements if we do not have 3D labels?

In Section 5.3.2, we show how to solve this chicken-and-egg problem with a hypothesize-and-verify approach: we hypothesize a style element, run it on the dataset, and check whether its pattern of detections is plausible. We evaluate the style element's detections by comparing it with a prior that assumes that the world is an inside-out-box. Training thus takes a collection of RGB images as input, and produces a dictionary of detectors as output. In Section 5.3.3, we describe how to use these style elements to interpret a new image: we run our style elements in a new image, and the detector responses vote for the underlying structure.

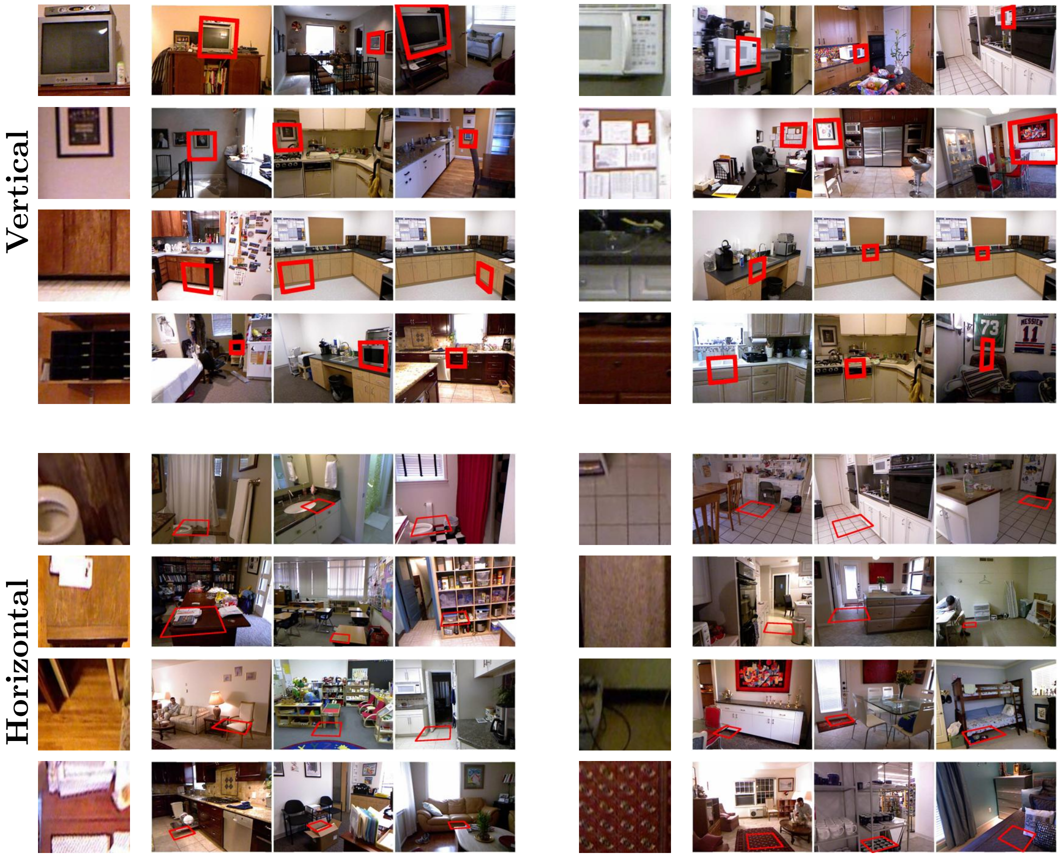


Figure 5.4: Selected style elements automatically discovered by our method. In each, we show the element on the left and its top detections on the discovery set; these and other detections are used to identify good and bad style elements. Notice that the top detections of most vertical style elements have a variety of orientations.

As this work is unsupervised, we make some assumptions. We use the Manhattan-world assumption [26] to reduce our label space to three orthogonal directions; we find these directions and rectification homographies for them using vanishing points estimated by [74]. We note, however, that there can be other directions present; we simply do not learn or detect style elements on them. We further assume that the images are upright so we can process the horizontal and vertical directions separately. Finally, our method models each style element as having a single label.

5.3 Approach

Our method begins with a discovery set of images and finds style elements that will help us interpret a new image. This task entails determining the orientation of surfaces throughout the discovery set so that we can obtain fronto-parallel rectified representations of the texture.

Since we have no explicit 3D labels, this task seems hopeless: in theory, each part of

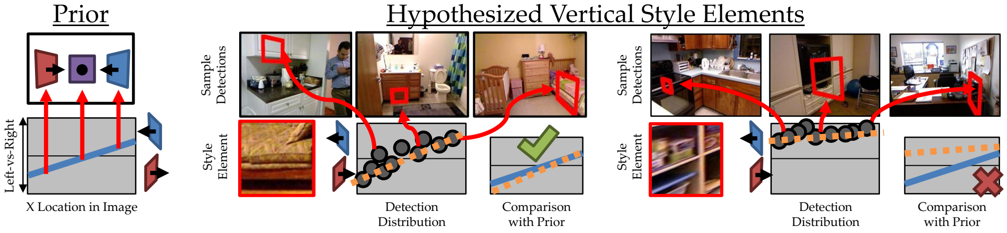


Figure 5.5: **Selecting hypotheses by their detections.** We compare hypothesized style elements' detections with our prior. We show a good (left) and a bad (right) style hypothesis in red squares. For each, we show a scatter plot of their detections on the discovery set, plotting the x-location in the image on the x-axis and how left-vs-right the detection is on the y-axis. We illustrate a few of these detections: for instance, the bedroom scene in the middle is a leftwards facing detection on the right-side of the image. On the far left, we show what our prior expects – a steady change from right-facing to left-facing. We rank elements by how well their detections match this prior: for instance, we reject the incorrect style element on the right since it predicts that everything faces left, which is unlikely under the prior.

each image could face any direction! We take a hypothesize-and-verify approach that lets us inject knowledge via a prior on the 3D structure of scenes. We guess a large number of style elements by rectifying the images and sampling patches. Most guesses are wrong, but some are right. We identify the correct ones by computing agreement between our prior and what each hypothesis would imply about the 3D structure of the world.

5.3.1 Prior

Consider the TV on the top-left of Fig. 5.4. How can we know that it is a good style element (i.e., rectified to be fronto-parallel) without knowing the underlying 3D of the image? While we do not know the 3D at that location, if we looked at the whole discovery set, we would observe a distinct pattern in terms of where TVs appear and in what direction they face: due to the regularity of human scenes, TVs on the left-hand-side of the image tend to face rightwards; on the right-hand-side, they tend to face leftwards. Thus, if we were to run our TV detector over the discovery set, we would expect to see this same distribution. On the other hand, it would be suspicious if we had a detector that only found leftwards facing TVs irrespective of where they appear in the image. We now explain how to formalize this intuition by constructing a prior that gives a probability of each orientation as a function of image location; this lets us score hypothetical style elements by their detections.

Our goal is to build a prior that evaluates the likelihood of a surface orientation as a function of pixel coordinate. Our overarching assumption is that our images are taken with an upright camera inside a box. Then, as in [81], we factor the question of orientation into two independent questions – "is the region vertical or horizontal?" and "if it is vertical, which vertical direction does it face?". We then assume the probability of vertical/horizontal depends on the y-coordinate in the image. For the vertical direction, we note that if we assume the world is a box, we can determine how likely each vertical direction is at each pixel as a function of its x coordinate.

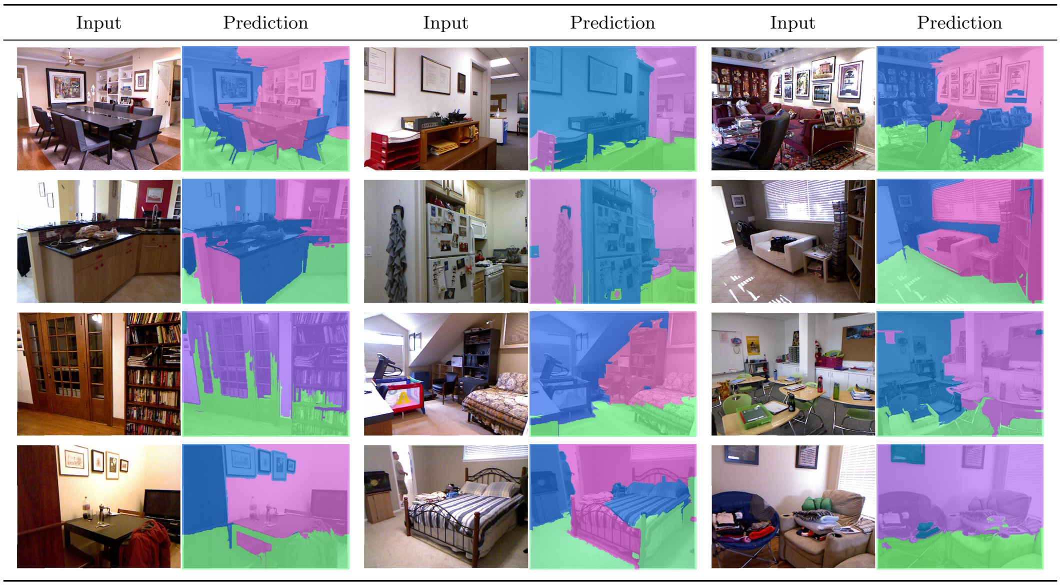


Figure 5.6: Sample results on NYUv2. First two rows: selected; last two row: random. In row 1, notice that the sides of objects are being labeled (discernible via conflicting colors), even though this severely violates the prior. In row 2, notice that our predictions can extend across the image or form convex corners: even though our prior is a box, we ignore in light of evidence from style elements.

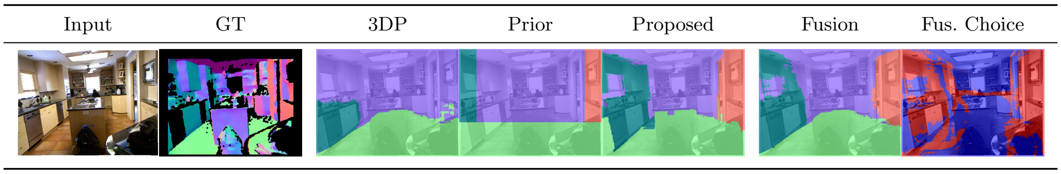


Figure 5.7: Comparison with the *supervised* 3DP method. The methods have complementary errors and we can fuse their results: 3DP often struggles with near-perpendicular surfaces; however these are easy to recognize once rectified by the proposed method. Our method has more trouble distinguishing floors from walls. We show the fusion result and which prediction is being used (Red: proposed; blue: 3DP).

We formalize this prior as follows, proceeding analytically since we do not have access to data. Since we expect horizontal surfaces like floors to be more common at the bottom of the image, we model the vertical/horizontal distinction with a negative exponential on the y-location, $\propto \exp(-y^2/\sigma^2)$. Since the camera is upright, the horizon determines the sign of the horizontal directions. For vertical directions, we assume the camera is inside a box with aspect ratio $\sim$ Uniform$[1, 2]$ and all rotations equally likely. The likelihood of each direction (left-to-right) as a function of x location can then be obtained in a Monte-Carlo fashion: we histogram normals at each location over renderings of 100K rooms sampled according to the assumptions.

5.3.2 Hypothesizing-and-Verifying Style Elements

Now that we have a way to verify a style element, we can use it in a hypothesize-and-verify approach. We first explain how we generate our hypotheses and then how we use the prior introduced in the previous section to verify hypothesized style patches.

We first need a way to generate hypotheses. Unfortunately, there are an infinite number of possible directions to try at each pixel. However, if we assume the world is a box, our search space is dramatically smaller: there are only 6 possible directions and these can be obtained by estimating Manhattan-world vanishing points in the image. Once we have rectified the image to these main scene directions, we sample a large collection ($\approx 25K$ total) of patches on these rectified images. Each patch is converted to a detector via an ELDA detector [70] over HOG [29]. Most patches will be wrong because the true scene geometry disagrees with them. One wrong hypothesis appears on the right of Fig. 5.5 in which a shelf has been rectified to the wrong direction.

We sift through these hypotheses by comparing what their detection pattern over the discovery set implies about 3D structure with our prior. For instance, if a style patch corresponds to a correctly rectified TV monitor, then our detections should, on average, match our box assumption. If it corresponds to an incorrectly rectified monitor then it will not match. We perform this by taking the ELDA detector for each patch and looking at the location and implied orientations of the top $1K$ detections over the training set. Since our prior assumes vertical and horizontal are separate questions, we have different criteria for each. For vertical surfaces, we compute average orientation as a function of x location and compare it to the average orientation under the prior, using the mean absolute difference as our criterion. For horizontal surfaces, our prior assumes that x location is independent from horizontal sign (i.e., floors do not just appear on the left); we additionally do not expect floor to share many style elements with ceilings. We thus compute the correlation between x and horizontal sign and the purity of up-vs-down labelings in the top firings. We illustrate this for two hypothesized vertical style elements in Fig. 5.5.

We use these criteria to rank a collection of hypothetical vertical and horizontal style elements. Our final model is the top 500 from each. We show some of the style elements we discover on NYU v2 in Fig. 5.4.

5.3.3 Inference

Given a new image and our style elements, we combine our prior and detections of the style elements to interpret the scene. We extract three directions from vanishing points to get our label space and run the style elements on the rectified images. The detections and the prior then vote for the final label. We maintain a multinomial distribution at each pixel over both whether the pixel is horizontal-vs-vertical and the vertical direction. Starting with the prior, we add a likelihood from detections: we count overlapping detections agreeing with the direction, weighted by score. We then take the maximum response, deciding whether the pixel is horizontal or vertical, and if the latter, the vertical orientation.

Our method produces good interpretations in many places, but does not handle ambiguous parts like untextured carpets well. These ambiguities are normally handled by transferring context [51] or doing some form of learned reasoning [52,74,111,148]. Without explicit 3D signal, we rely on unsupervised label propagation over segments: we extract multiple segmentations by varying the parameters of [44] ($\sigma = 0.5, 1, 1.5, 2$; $k = 100, 200$; $\min = 50, 100$). Each segment assumes the mode of its pixels, and the final label of a pixel is the mode over the segmentations.

5.3.4 Implementation Details:

We finally report a number of implementation details of our method; more details appear in the supplement. *Patch representation:* Throughout the approach, we use HOG features [29] with a 8×8 pixel cells at a canonical size of 80 pixels. *Rectification:* we obtain Manhattan-world vanishing points from [74] and rectify following [172]: after auto-calibration, the remaining parameters up to a similarity transform are determined via vanishing point orthogonality; the similarity transform is handled by aligning the Manhattan directions with the image axes and by operating at multiple scales. Sample rectified images appear in Fig. 5.2: our detectors are discovered and tested on these images. At test time, we max-pool detector responses over multiple rectifications per vertical direction. *Initial Patch Pool:* Our hypotheses are obtained by rectifying each image in the discovery set to the scene directions and following the sampling strategy of [152] while rejecting patches whose corresponding quadrilateral has area $< 100^2$ pixels.

5.4 Experiments

We now describe experiments done to validate the approach. We are guided by the following questions: (1) How well does the method work? (2) Can the approach be combined with supervised methods? and (3) Are there scenarios that only an unsupervised approach can handle?

To answer the first two questions, we use the NYUv2 [150] dataset, which has gained wide acceptance; we find that our method does nearly as well as a comparable supervised method and that a simple learned fusion of the methods matches or surpasses the state-of-the-art in 4/6 metrics among methods not using the larger video dataset. To answer the final question, we use the Places-205 dataset [175], which has many locations not covered by Kinect datasets. Supervised approaches must resort to a model trained on existing datasets, but our method can adapt to the dataset. We find that this enables us to outperform a comparable supervised method by a large margin (3.7%).

5.4.1 Experiments on NYU v2

We first document our experimental setup. We follow standard protocols and compare against the state-of-the-art.

Data: We use the standard splits of [150]. We use the ground-truth normals from [109] but found similar conclusions on those from [51].

Evaluation Criteria: As introduced by [51], we evaluate results on a per-pixel basis over all over valid pixels. We report the mean, median, RMSE and the fraction of pixels with error below a threshold t, or PGP-t (percent good pixels) for $t = 11.25°, 22.5°, 30°$. Like [81], our model breaks the task into a vertical/horizontal problem and a vertical subcategory problem. We evaluate both, and evaluate the vertical task on surfaces within $30°$ of the y-axis.

Baselines: We stress that our goal as an unsupervised method is not to outperform supervised methods but instead to show that our approach is effective. We report results of all methods that could be considered state-of-the-art at the time of submission, including even

Table 5.1: Evaluation on all pixels on NYU v2. The most informative comparison is with 3DP and UNFOLD. Our unsupervised approach nearly matches 3DP and in combination with 3DP, obtains state-of-the-art results in 4/6 metrics. Starred methods do not use the Manhattan-world assumption.

	Summary Stats. (°)			% Good Pixels		
	(Lower Better)			(Higher Better)		
	Mean	Median	RMSE	11.25°	22.5°	30°
3DP+Prop.	34.6	**17.5**	48.7	**40.6**	**54.7**	**59.7**
Ladicky* [109]	**33.5**	23.1	**44.6**	27.7	49.0	58.7
UNFOLD [52]	35.2	17.9	49.6	**40.5**	54.1	58.9
3DP [51]	36.3	19.2	50.4	39.2	52.9	57.8
Proposed	38.6	21.7	52.6	36.7	50.6	55.4
Lee et al. [112]	43.8	35.8	55.8	26.8	41.2	46.6
		(With External Data)				
Wang [162]	26.9	14.8	-NR-	42.0	61.2	68.2
Eigen* [37]	23.7	15.5	-NR-	39.2	62.0	71.1

Table 5.2: Ablative analysis

	Mean	Median	RMSE	11.25°	22.5°	30°
Full	38.6	21.7	52.6	36.8	50.6	55.4
No Segm.	39.6	23.4	53.5	35.7	49.3	54.0
Prior	43.2	30.2	56.7	33.1	45.2	49.8
3DP on Prior	41.7	27.0	55.6	34.6	47.1	51.6

those from methods using the much larger video dataset [37, 162]. The most informative comparison is with the Manhattan-world version of 3DP [51] because it keeps two sources of variance fixed, the base feature and the Manhattan-world assumption.

Combining with supervised methods: We learn a model, termed *3DP+Prop* that fuses our method with 3DP. Following [119], we learn random forests (100 trees, cross-validated min-children) on training data to predict whether each method's outputs are within 22.5° of the ground-truth. We train separate forests for our method before segmentation propagation, its vertical predictions only, and 3DP. We use for features the confidences and normals from all methods and the image location. At test time, we take the prediction that is most likely to be correct.

Results: We first report qualitative results in Fig. 5.6. Our method is unsupervised but obtains an accurate interpretation on most scenes. The method frequently picks up on small details, for instance, the right-wards facing chair back (1st row, left) and the side of the shelving in (1st, middle). These small details, as well as the correct inwards-pointing box of the refrigerator (2, middle), are unlikely under the box prior and demonstrate that the method is not simply regurgitating the prior. Our unsupervised propagation is sometimes too aggressive, as seen in (2nd, right, under the blinds; 3rd, left, on the floor), but helps with boundaries.

We report quantitative results in Table 5.1. Our method is always within 2.5° of the

Table 5.3: **Vertical** evaluation. Our method outperforms a 3DP on 3/6 metrics, but fusing the two gives a substantial boost

	Summary Stats. (°)			% Good Pixels		
	(Lower Better)			(Higher Better)		
	Mean	Median	RMSE	11.25°	22.5°	30°
Prior	39.3	26.0	52.0	30.4	46.7	53.2
Proposed	**33.9**	**19.7**	**46.5**	35.1	**53.2**	**59.7**
3DP [51]	**33.9**	19.9	**46.5**	**36.4**	52.6	58.8
3DP+Prop.	32.1	18.0	44.6	37.5	55.2	61.5

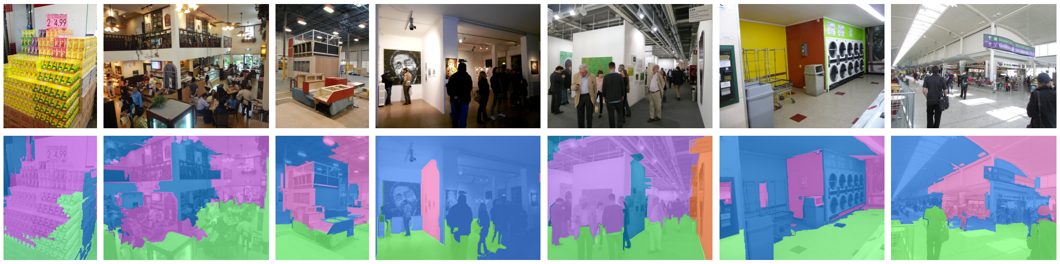


Figure 5.8: Results on Places-205. Note the stark contrast with NYUv2. Our method can learn from this new data despite a lack of labels.

most immediately comparable supervised method, 3DP, and is always within 3.8° of [52], which fuses multiple models in a complex MIQP formulation. We also substantially outperform [112], the other unsupervised method for this task. Our simple fusion of 3DP and our method obtains state-of-the-art results in 4/6 metrics among methods using only the original 795 training images. Since the Manhattan-world assumption is rewarded by some metrics and not by others, fair comparison with non-Manhattan-world methods such as [37, 109] is difficult: e.g., on PGP-11.25, due to our use of vanishing points, our method is within 2.4% of [37], which uses orders of magnitude more data.

We report ablative analysis in Table 5.2: we greatly outperform relying on only the prior (i.e., not using detection evidence), especially on the median, and segmentation helps but does not drive performance. Training 3DP using label maps from the prior yielded worse performance. This is because 3DP relies heavily on junctions and edges in the normal map, and the prior does not provide this detailed information. We found our method to be insensitive to parameters: changing the box prior's aspect ratio or the prior weight by a factor of two yields changes $< 0.7°$ and $< 0.6\%$ across metrics; many settings produced better results than the settings used (more details in the supplement).

Our result is better in relative terms on the vertical task, as can be seen in Table 5.3: it matches 3DP in 2 metrics and bests it in 3. This is because many indoor horizontal surfaces are defined by location and lack of texture, which our single-plane HOG patch approach cannot leverage; 3DP, on the other hand, learns structured patches and its vocabulary captures horizontal surfaces via edges and corners. Again, fusing our method with 3DP improves results.

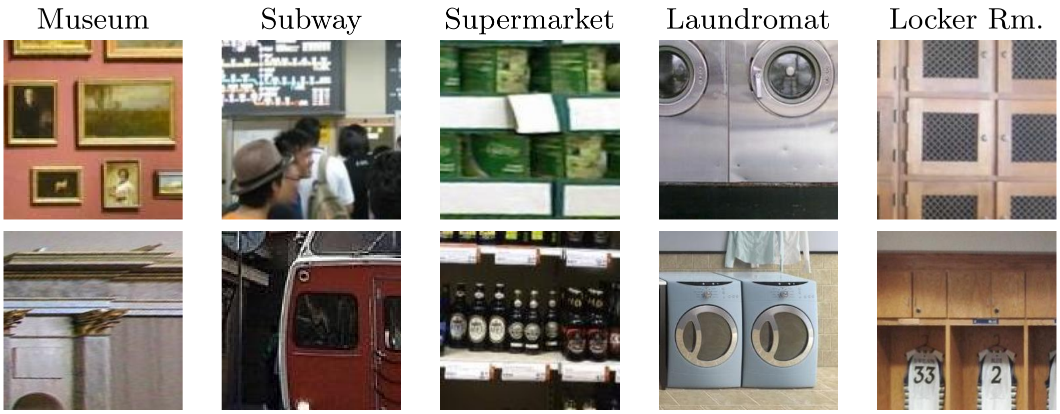


Figure 5.9: Example vertical elements learned on Internet images.

Table 5.4: Accuracy on Places-205, subdivided by category.

	Airport	Art Gallery	Conference.	Locker Rm.	Laundromat
3DP	50.1	64.2	63.0	65.9	65.1
FC- [162]	51.6	70.3	**71.2**	**69.4**	**71.9**
Prop.	**54.0**	**71.1**	64.3	67.8	71.4
Museum	Restaurant	Shoe Shop	Subway	Supermarket	Avg.
58.9	57.6	59.9	**58.2**	49.3	59.2
61.2	**63.2**	**64.0**	56.2	57.7	**63.7**
64.0	60.5	62.1	52.3	**61.9**	62.9

5.4.2 Internet Images

We now investigate tasks that only an unsupervised method can do. Suppose one wants to interpret pictures of supermarkets, airport terminals, or another place not covered by existing RGBD datasets. The only option for a supervised approach (besides collecting new data at great expense) is to use a pre-trained model. However, with our unsupervised 3D approach, we can learn a model from images alone.

Data: We collected a subset of 10 categories (Airport, art gallery, conference room, locker room, laundromat, museum, restaurant, shoe shop, subway, supermarket) from the Places-205 dataset [175] that are not present in 3D datasets and annotated them with Manhattan-world labelings. We took at most 700 images from each category, and set aside 200 for evaluation. We sparsely annotated 10 superpixels in each image, each randomly selected to avoid bias; each could be labeled as one of the 6 possible Manhattan normals or passed if multiple directions were present. Since our label space is Manhattan world, we removed images where vanishing points could not be estimated, identified as ones where the independent estimates of [74, 112] disagree.

Results: We learned an unsupervised model on each category and compared its performance with models pretrained on NYU, including 3DP and a fully convolutional variant of [162] which obtains within 1.1% PGP-30 on NYU. Note that standard supervised methods cannot compensate for this shift. We show qualitative results illustrating the dataset and our approach in Fig. 5.8: even NYU contains no washing machines, but our approach can learn elements for these from Internet data, as seen in Fig. 5.9. On the other hand, pretrained methods struggle to adapt to this new data. We report results in Table 5.4. Our approach outperforms 3DP, pretrained on NYUv2, in 9/10 categories and overall by 3.7% and outperforms [162] in 4/10 categories, with gains as large as 3.9%. Our labels are sparse so we verify the gain over 3DP with a 95% bootstrapped confidence interval; our approach consistently outperforms 3DP ($[2.7, 4.8]$).

5.5 Discussion

This chapter demonstrated an explicit instantiation of our factorization in the form of style elements that recognize style in its canonical form. These elements can be learned from ordinary data without 3D training data. Additionally, these style elements enable sharing across viewpoints and can be recognized even in new 3D structure configurations. As a broader point, these demonstrate the value in modeling the 3D understanding problem as something beyond labeling each pixel with a set of geometric labels.

Chapter 6

Predicting Higher-Order 3D Structure

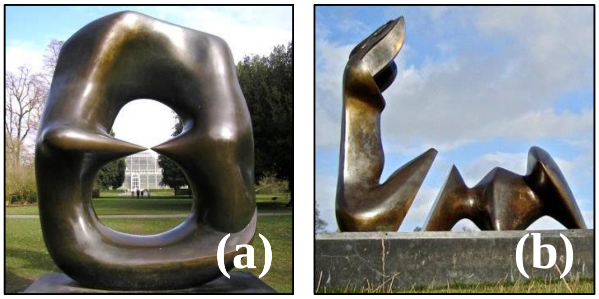


Figure 6.1: How would you describe the shape of (a) and contrast it with (b)? Probably not by quantifying the depth at each pixel, but instead characterizing the overall 3D shape: this chapter proposes to think of 3D structure in these terms.

Much of this dissertation focuses on per-pixel metric properties of scenes such as dense surface normals; these properties are also viewpoint-dependent in the sense that they change as one moves about. While important, these dense viewpoint-dependent metric maps do not tell the whole story of objects. As an example, consider the sculpture in Fig. 6.1(a). How would you characterize the sculpture to a friend in such a way that she knew that you were referring to the one in Fig. 6.1(a) as opposed to (b)? Repeating to her a metric map such as normals or depth would certainly raise eyebrows and might not be very helpful. Instead it would be more sensible to give a high-level qualitative description using shape, holes, curvature, sharpness/smoothness, etc.

The objective of this chapter is to infer such generic *3D shape* properties directly from appearance. We term these properties **3D shape attributes** and introduce a variety of specific examples, for instance planarity, thinness, and point-contact, to concretely explore this concept. Such attributes can be derived from an estimated depthmap in principle. However, this indirect approach is inferior in both theory and practice, as we argued in Chapter 3 for the specific case of normals and as we will show in this chapter with baselines: view dependence, insufficient resolution, errors and ambiguities in the reconstruction result in worse performance.

As with classical object attributes and relative attributes [42,45,128], 3D attributes offer a means of describing 3D object shape when confronted with something entirely new – the *open world problem*. This is in contrast to a long line of work which is able to say something about 3D shape, or indeed recover it, from single images *given* a specific object class, e.g. faces [17], semantic category [95] or cuboidal room structure [74]. While there has been success in determining *how* to apply these constraints, the problem of *which* constraints to apply is much less explored, especially in the case of completely new objects. Used inappropriately, scene understanding methods such as the one introduced Chapter 4 produce either unconstrained results in which walls that should be flat bend arbitrarily or planar interpretations, such as the one that will be introduced in Chapter 7 in which non-planar objects like lamps are flat. Shape attributes can act as a generic way of representing top-down properties for 3D understanding, sharing with classical attributes the advantage of both learning and application across multiple object classes.

There are two natural questions to ask: what 3D attributes should be inferred, and how to infer them? In Section 6.2, we introduce our attribute vocabulary, which draws inspiration from and revisits past work in both the computer and the human vision literature. We return to these ideas with modern computer vision tools. In particular, as we describe in Section 6.4, we use Convolutional Neural Networks (CNNs) to model this mapping and learn a model over all of the attributes as well as a shape embedding.

The next important question is: what data should we use to use to investigate these properties? We use photos of modern sculptures from Flickr, and describe a procedure for gathering a large and diverse dataset in Section 6.3. This data has many desirable properties: it has much greater variety in terms of shape compared to common-place objects; it is real and in the wild, so has all the challenging artifacts such as severe lighting and varying texture that may be missing in synthetic data. Additionally, the dataset is automatically organized into: *artists*, which lets us define a train/test split to generalize over artists; *works* (of art) irrespective of material or location, which lets us concentrate on shape, and *viewpoint clusters*, which lets us recognize sculptures from multiple views and aspects.

The experiments in Section 6.5 show that we are indeed able to infer 3D attributes. However, we also ask the question of whether we are actually learning 3D properties, or instead a proxy property, such as the identity of the artist, which in turn enables these properties to be inferred. We have designed the experiments both to avoid this possibility and to probe this issue, and discuss this there.

6.1 Background

How do 2D images convey 3D properties of objects? This is one of the central questions in any discipline involving perception – from visual psychophysics to computer vision to art. This chapter draws on each of these fields, for instance in providing data and inspiration to picking the particular attributes or probing our learned model.

One motivation for our investigation of shape attributes is a long history of work in the human perception community that aims to go beyond metric properties and address holistic shape in a view-independent way. As noted in Chapter 3, amongst many others, Koenderink and van Doorn [102] argued for a set of shape classes based on the *sign* of the principal curvatures and also that shape perception was not metric [103,104], and Biederman [15] advocated shape classes based on non-accidental *qualitative* contour properties. This chapter aims to try to merge these qualitative properties with large-scale visual learn-

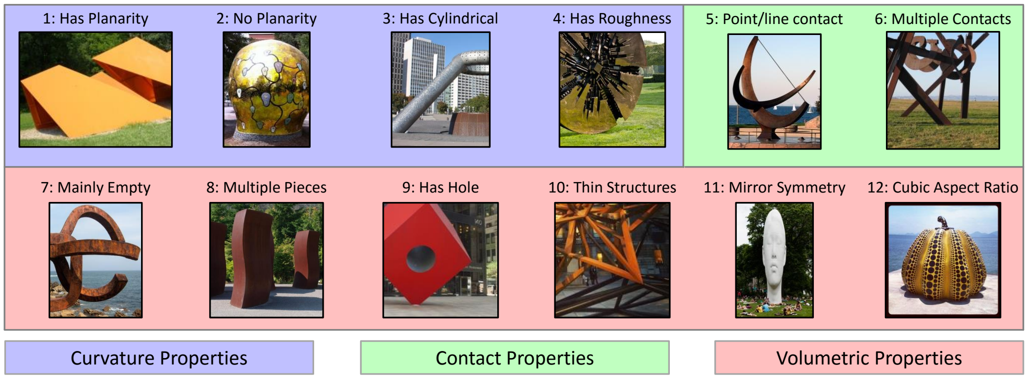


Figure 6.2: The 3D shape attributes investigated in this paper, and an illustration of each from our training set. Additional sample annotations are shown in Fig. 6.5.

ing. An additional source of inspiration from this domain is from work trying to use mental rotation [149, 157] to probe how humans represent shape; here, we use it to probe whether our models have learned something sensible.

A great deal of research in early computer vision sought to extract local or qualitative cues to shape, for instance from apparent contours [99], self-shadows and specularities [100, 177]. Recent computer vision approaches to shape perception, including the ones presented in Chapters 4 and 5 have increasingly reduced 3D understanding to the task of inferring a viewpoint-dependent 3D depth or normal at each pixel [11, 39, 51]. These predictions are useful for many tasks but do not tell the whole story. This work aims to help fill this gap in a complementary way by revisiting these non-metric qualitative questions. Some exceptions to this trend include the qualitative labels explored in [63, 81] like porous, but these initial efforts had limited scope in terms of data variety, vocabulary size, and quantity of images.

Our focus is 3D shape understanding, but we pose our investigation into these properties in the language of attributes [42, 45, 107, 110, 128] to emphasize their key properties of communicability and open-world generalization. The vast majority of attributes, however, have been semantic and there has never been, to our knowledge, a systematic attempt to connect attributes with 3D understanding or to study them with data specialized for 3D understanding. Our work is most related to the handful of coarse 3D properties in [42]. In addition to having a larger number of attributes and data designed for 3D understanding, our attributes are largely unaffected by viewpoint change.

6.2 3D Attribute Vocabulary

Which 3D shape attributes should we model? We choose 12 attributes based on questions about three properties of historical interest to the vision community – curvature (how does the surface curve locally and globally?), ground contact (how does the shape touch the ground?), and volumetric occupancy (how does the shape take up space?).

Fig. 6.2 illustrates the 12 attributes, and sample annotations are shown in Fig. 6.5, with more in the supplementary material. We now briefly describe the attributes in terms of curvature, contact, and volumetric occupancy.

Curvature: We take inspiration from a line of work on shape categorization via curvature led by Koenderink and van Doorn (e.g., [102]). Most sculptures have a mix of convex and concave regions, so we analyze where curvature is zero in at least one direction and look for (1) piecewise planar sculptures, and (2) sculptures with *no* planar regions (many have a mix of planar/non-planar), (3) sculptures where one principal curvature vanishes (e.g., cylindrical), and (4) rough sculptures where locally the surface changes rapidly.
Contact: Contact reasoning plays a strong role in understanding (e.g., [63,76,78]); we characterize ground contact via (5) point/line contact-vs-solid contact, and (6) whether it contacts the ground in multiple places.
Volumetric: Reasoning about free-space has long been a goal of 3D understanding [76, 111, 136]. We ask (7) the fraction of occupied space; (8) whether the sculpture has multiple pieces; (9) whether there are holes; (10) whether it has thin structures; (11) whether it has mirror symmetry; and (12) whether it has a cubic aspect ratio. These are, of course, not a complete set: we do not model, for example, enclosure properties or differentiate a hole from a mesh.

Note, of the 12 attributes, 10 are relatively unaffected by a geometric affine transformation of the image (or 3D space) – only the cubic aspect ratio and mirror symmetry attributes are measuring a global metric property.

6.3 Gathering a Dataset of 3D Shapes

In order to investigate these 3D attributes, we need a dataset of sculptures that has a diversity of shape so that different subsets of attributes apply. We also require images spanning many viewpoints of each sculpture in order to investigate the estimation of attributes against viewpoint.

Since artists often produce work in a similar style (Calder's sculptures are mostly piecewise planar, Brancusi's egg-shaped) we therefore need a variety of artists and multiple works/images of each. Previous sculpture datasets [4, 5] are not suitable for this task as they only contain a small number of artists and viewpoints.

Thus we gather a new dataset from Flickr. We adopt a five stage process to semi-automatically do this: (i) obtain a vocabulary of artists and works (for which many images will be available); (ii) cluster the works by viewpoint; (iii) clean up mistakes; (iv) query expand for more examples from Google images; and (v) label attributes. Note, organization by artist is not strictly necessary. However, artists are used subsequently to split the works into train and test datasets: as noted above, due to an artists' style, shape attributes frequently correlate with an artist; Consequently artists in the train and test splits should be disjoint to avoid an overly optimistic generalization performance. The statistics for these stages are given in Tab. 6.1.

6.3.1 Generating a vocabulary of artists and works

Our goal is to generate a vocabulary of artists and works that is as broad as possible. We begin by producing a list of artists, combining manually generated lists with automatic ones, and then expand each artist to a list of their works.

The manual list consists of the artists exhibited at six sculpture parks picked from online top-10 lists, as well as those appearing in Wikipedia's article on Modern Sculpture. An automatic list is generated from metadata from the 20 largest sculpture groups on Flickr:

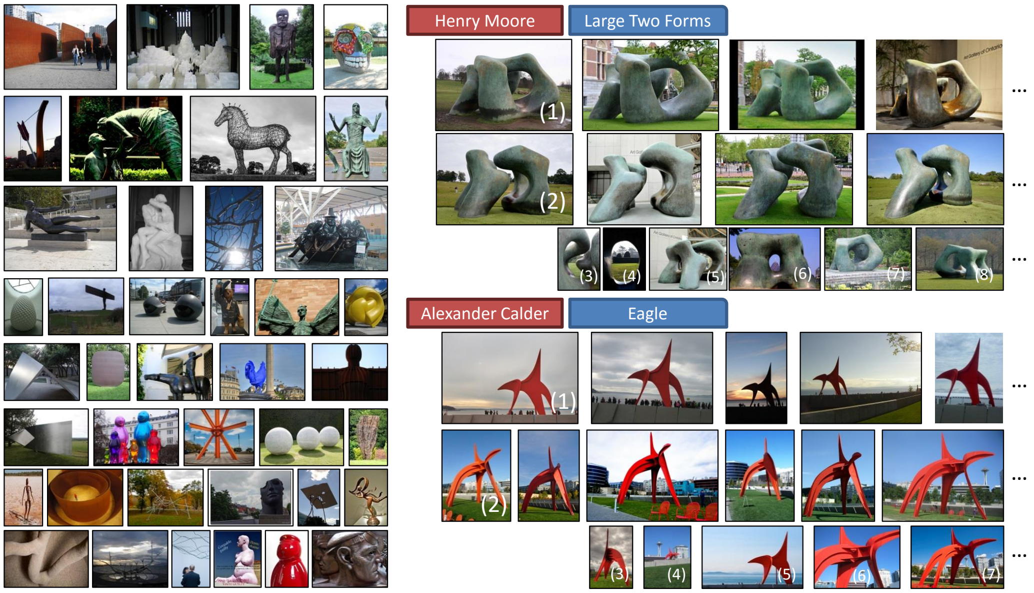


Figure 6.3: The dataset consists of 143K images of sculptures that were gathered from Flickr and Google images. A representative sample is shown on the left. Note the great variety in shape, material, and style. More samples appear in the supplementary material. Our data has structure in terms of artist, work, and viewpoint cluster (shown numbered on the right). Each is important for investigating 3D shape attributes.

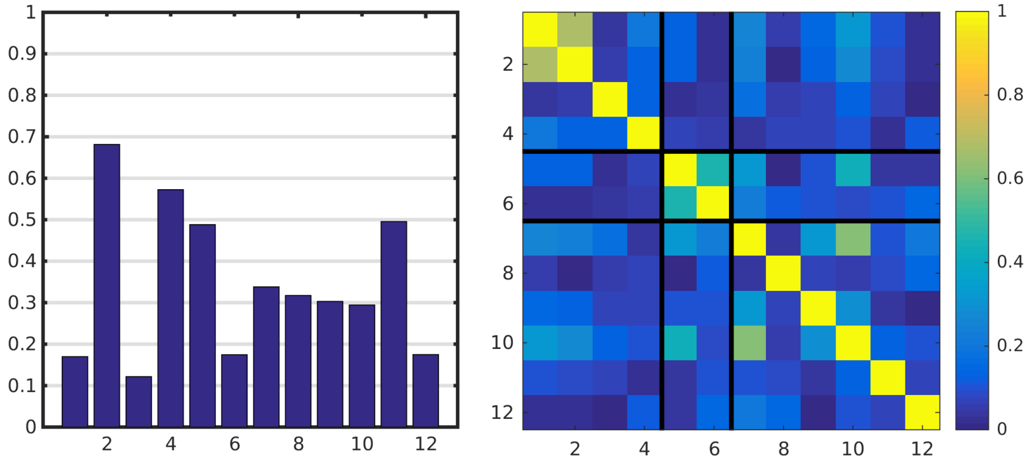
we analyze image titles for text indicating that a work is being ascribed to an artist, and take frequent bigrams and trigrams. The two lists are manually filtered to remove misspellings, painters and architects, a handful of mistakes, and artists with fewer than 250 results on Flickr. This yields 258 artists (95 from the manual list, and 163 from the automatic).

We now find a list of potential works for each artist using both Wikipedia and text analysis on Flickr. We query the sculptor's page on Wikipedia, possibly manually disambiguating, and propose any italicized text in the main body of the article as a possible work. We also query Flickr for the artists' works (e.g., Tony Smith Sculpture), and do n-gram analysis in titles and descriptions in front of phrases indicating attribution to the sculpture (e.g., "by Tony Smith"). In both cases, as in [132], stop-word lists were effective in filtering out noise. While Wikipedia has high precision, its recall is moderate at best and zero for most artists. Thus querying Flickr is crucial for obtaining high quality data. Finally, images are downloaded from Flickr for each work of each artist.

6.3.2 Building viewpoint clusters

Images from each work are partitioned into *viewpoint clusters*. These clusters are image sets that, for example, capture a different visual aspect of the work (e.g. from the front or side) or are acquired from a particular distance or scale (e.g. a close up). Fig. 6.3 shows example viewpoint clusters for several works.

There are two principal reasons for obtaining viewpoint clusters: (i) it enables recogni-



(a) Positive Frequency (b) Correlation

Key: (1) Planar (2) No Planar (3) Cylindrical (4) Rough (5) Point Contact (6) Multiple Contact (7) Empty (8) Multiple Pieces (9) Holes (10) Thin (11) Symmetric (12) Cubic

Figure 6.4: (a) Frequency of each attribute (i.e., # positives /# labeled); (b) Correlation between attributes.

Table 6.1: Data statistics at each stage and the trainval/test splits.

Stage	Images	Artists	Works	View. Clusters
Initial	352K	258	3412	–
View Clust.	213K	246	2277	16K
Cleaned	97K	242	2197	9K
Query Exp.	143K	242	2197	9K
Trainval/Test	109K/35K	181/61	1655/532	7.2K/2.1K

tion of a work from different viewpoints to be evaluated; and (ii) it makes label annotation more efficient as attributes are in general valid for all images of a cluster. Note, it might be thought that attributes could be labelled at the work level, but this is not always the case. For example, the hole in a Henry Moore sculpture or the ground contact of an Alexander Calder sculpture may not be visible in some viewpoint clusters, so those clusters will be labelled differently from the rest (i.e. no hole for the former, and unknown for the latter).

Clustering proceeds in a standard manner by defining a similarity matrix between image pairs, and using spectral clustering over the matrix. The pairwise similarity measure takes into account: (i) the number of correspondences (that there are a threshold number); (ii) the stability of these correspondences (using cyclic consistency as in [176]); and (iii) the viewpoint change (the rotation and aspect ratio change obtained from an affine transformation between the images). Computing correspondences requires some care though since sculptures often do not have texture (and thus SIFT like detections cannot be used). We follow [3] and first obtain a local boundary descriptor for the sculpture (by foreground-

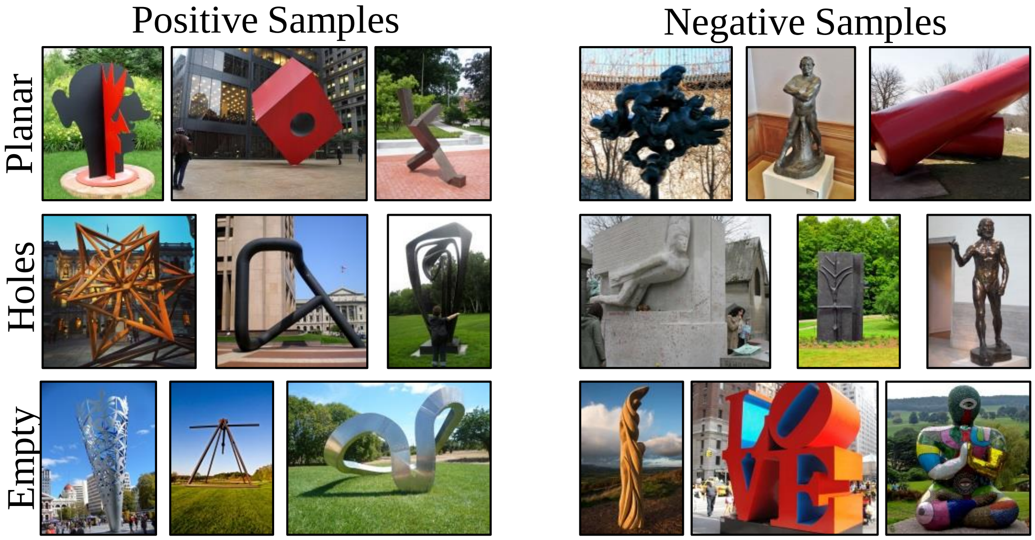


Figure 6.5: Sample positive and negative annotations from the dataset for the planar, has-holes, and empty attributes. More examples are included in the supplementary material.

background segmentation and MCG [7] edges for the boundaries), and then obtain geometrically consistent correspondences using an affine fundamental matrix. Finally, a loose affine transformation is computed from the correspondences (loose because the sculpture may be non-planar, hence the earlier use of a fundamental matrix).

In general, this procedure produces clusters with high purity. The main failure is when an artist has several visually similar works (e.g. busts) that are confused in the meta-data used to download them. We also experimented with using GPS, but found the tags to be too coarse and noisy to define satisfactory viewpoint clusters.

6.3.3 Data Cleanup

The above processes are mainly automatic and consequently make some mistakes. A number of manual and semi-automatic post-processing steps are therefore applied to address the main failings. Note, we can quickly manipulate the dataset via viewpoint clusters as opposed to handling each and every image individually.

Cluster filtering: Each cluster is checked manually using three sample images to reject clearly impure clusters.

Regrouping: Some of the automatically generated works are ambiguous due to noisy meta-data: for instance “Reclining Figure” describes a number of Henry Moore sculptures. After clustering, these are reassigned to the correct works.

Outlier image removal: A 1-vs-rest SVM is trained for each work, using fc7 activations of a CNN [106] pretrained on ImageNet [138]. Each work’s images are sorted according to the SVM score, and the bottom images ($\approx 10K$ across all works) flagged for verification.

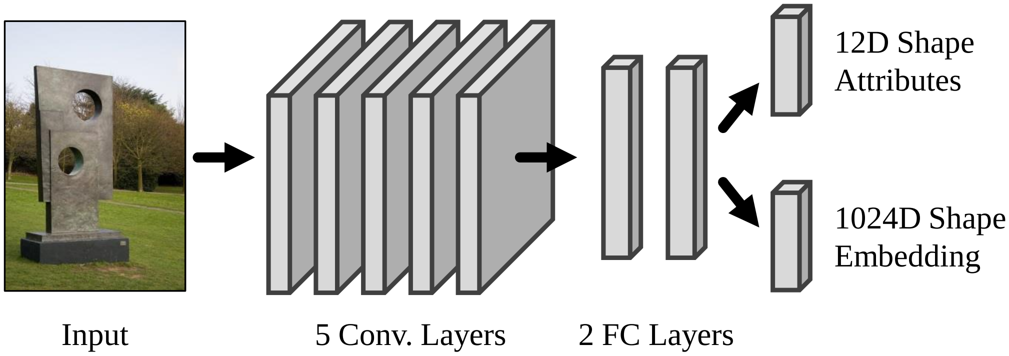


Figure 6.6: The multi-task network architecture, based on VGG-M. After shared layers, the network branches into layers specialized for attribute classification and shape embedding.

6.3.4 Expansion Via Search Engines

Finally, we augment the dataset by querying Google. We perform queries with the artist and work name. Using the same CNN activation + SVM technique from the outlier removal stage, we re-sort the query results and add the top images after verification. This yields $\approx 45K$ more images.

6.3.5 Attribute Labeling

The final step is to label the images with attributes. Here, the viewpoint clusters are crucial, as they enable the labeling of multiple images at once. Each viewpoint cluster is labeled with each attribute, or can be labeled as N/A in case the attribute cannot be determined from the image (e.g., contact properties for a hanging sculpture). One difficulty is determining a threshold: few sculptures are only planar and no sculpture is fully empty. We assume an attribute is satisfied if it is true for a substantial fraction of the sculpture, typically 80%. To give a sense of attribute frequency, we show the fraction of positives in Fig. 6.4(a).

The dataset is also diverse in terms of combinations of attributes and inter-attribute correlation. There are $2^{12} = 4096$ possible combinations, of which 393 occur in our data. Most attributes are uncorrelated according to the correlation coefficient ϕ, as seen in Fig. 6.4(b): mean correlation is $\phi = 0.20$ and 72% of pairs have $\phi < 0.2$. The two strong correlations ($\phi > 0.5$) are, unsurprisingly, (1) planarity and no planarity; and (2) emptiness and thinness.

6.4 Approach

We now describe the CNN architecture and loss functions that we use to learn the attribute predictors and shape embedding. We cast this as multi-task training and optimize directly for both. Specifically, the network is trained using a loss function over all attributes as well as an embedding loss that encourages instances of the same shape to have the same representation. The former lets us model the attributes that are currently labeled. The latter forces the network to learn a representation that can distinguish sculptures, implicitly modeling aspects of shape not currently labeled.

Network Architecture: We adapt the VGG-M architecture proposed in [22]. We depict the overall architecture in Fig. 6.6: all layers are shared through the last fully connected layer, fc7. After fc7, the model splits into two branches, one for attributes, the other for embedding. The first is an affine map to 12D followed by independent sigmoids, producing 12 separate probabilities, one per attribute. The second projects fc7 to a 1024D embedding which is then normalized to unit norm.

We directly optimize the network for both outputs, which allows us to obtain strong performance on both tasks. The first loss models all the attributes with a cross-entropy loss summed over the valid attributes. Suppose there are N samples and L attributes, each of which can be 1 or 0 as well as $\emptyset$ to indicate that the attribute is not labeled; the loss is

$$L(Y,P) = \sum_{i=1}^{N} \sum_{\substack{l=1 \\ Y_{i,l}\neq\emptyset}}^{L} Y_{i,l} \log(P_{i,l}) + (1 - Y_{i,l}) \log(1 - P_{i,l}), \tag{6.1}$$

for image i and label l, where we denote the label matrix as $Y_{i,l} \in \{0,1,\emptyset\}^{N,L}$ and the predicted probabilities as $P_{i,l} \in [0,1]^{N,L}$. The second loss is an embedding loss over triplets as in [145, 146, 163]. Each triplet i consists of an anchor view of one object x_i^a, another view of the same object x_i^p, as well as a view of a different object x_i^n. The loss aims to ensure that two images of the same object are closer in feature space compared to another object by a margin:

$$\sum_{i=1} \max\left(D(x_i^a, x_i^p) - D(x_i^a, x_i^n) + \alpha, 0\right) \tag{6.2}$$

where $D(\cdot,\cdot)$ is squared Euclidean distance. We generate triplets in a mini-batch and use soft-margin violaters [145].

We see a number of advantages to multi-task learning. It yields a network that can both name attributes it knows about and model the 3D shape space implicitly. Additionally, we found it to improve learning stability, especially compared to individually modeling each attribute.

Configurations: We explore two configurations to validate that we are really learning about 3D shape. Unless otherwise specified, we use the system described above, *Full*. However, to probe what is being learned in one experiment, we also learn a network that only optimizes the attribute Loss (6.1), which we refer to as *Attribute-Only*.

Implementation Details: *Optimization:* We use a standard stochastic gradient descent plus momentum approach with a batch size of 128. *Initialization:* We initialize the network using the model from [22] which was pre-trained on image classification [138]. *Parameters:* We use a learning rate of 10^{-4} for the pre-trained layers, and 10^{-3} and 10^{-2} for classification and embedding layers respectively. *Augmentation:* At training time, we use random crops, flips, and color jitter. At test time, we sum-pool over multiple scales, crops and flips as in [22].

6.5 Experiments

We describe a set of experiments to investigate both the performance of the learnt 3D shape attribute classifiers, *and* what has been learnt. We aim to answer two basic questions: (1) how well can we predict 3D shape attributes from a single image? and (2) are we actually predicting 3D properties or a proxy property that correlates with attributes in an image? To

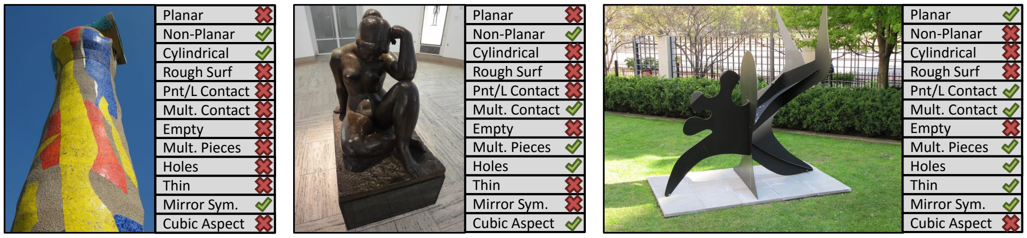
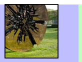
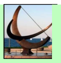
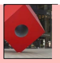
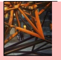


Figure 6.7: Predictions for all attributes on test images. The system has never seen these sculptures or ones by the artists who made them, but generalizes successfully.

Table 6.2: Area under the ROC curve. Higher is better. Our approach outperforms the baselines by a large margin and achieves strong performance on an absolute basis.

	Curvature				**Contact**		**Occupancy**						
													
Method	Plan	¬ Plar	Cyl	Rough	P/L	Mult	Emp	Mult	Hole	Thin	Sym	Cubic	Mean
[11] + [18]	64.1	63.4	51.2	61.3	61.1	61.6	66.5	52.8	56.0	63.5	56.2	55.7	59.4
[39] + [18]	64.6	61.0	50.6	60.6	57.5	60.9	65.2	55.7	52.4	65.7	57.2	51.2	58.5
[11] + [66]	70.0	64.4	53.1	63.9	63.6	64.8	73.7	56.4	54.1	69.7	60.2	56.2	62.5
[39] + [66]	67.5	61.8	51.9	64.8	58.5	64.8	71.5	57.8	52.4	67.7	59.4	56.1	61.2
Proposed	**82.8**	**77.2**	**56.9**	**76.0**	**74.4**	**76.4**	**87.0**	**60.4**	**69.3**	**85.8**	**60.8**	**60.3**	**72.3**

address (1) we evaluate the performance on the Sculpture Images Test set, and also compare to alternative approaches that first predict a metric 3D representation and then derive 3D attributes from that (Sec. 6.5.1). We probe (2) in two ways. First, by evaluating the learnt representation for a different task – determining if two images from different viewpoints are of the same object or not (Sec. 6.5.2); and, second, by evaluating how well the 3D shape attributes trained on the Sculpture images generalize to non-sculpture data, in particular to predicting shape attributes on PASCAL VOC categories (Sec. 6.5.3).

6.5.1 Attribute Prediction

We first evaluate how well 3D shape attributes can be estimated from images. Here, we report results for our full network. Since our dataset is large enough, the attribute-only network does similarly. We compare directly inferring holistic attributes to a number of baselines that are depth orientated, and start by computing a metric depth at every pixel.

Baselines: The baselines start by estimating a metric 3D map, and then attributes are extracted from this map. We use two recent methods for estimating depth from single images with code available: a CNN-based depth estimation technique [39] and an intrinsic images technique [11]. Since [11] expects a mask, we use the segmentation used for collecting the dataset (in Sec. 6.3.2). One question is: how do we convert these depthmaps into our attributes? Hand-designing a method is likely to produce poor results. We take a data-driven approach and treat it as a classification problem. We use two approaches that have produced strong performance in the past. The first is a linear SVM on top of kernel depth descriptors [18], which convert the depthmap into a high-dimensional vector incorporating depth configurations and image location. The second is the HHA scheme [66], which con-

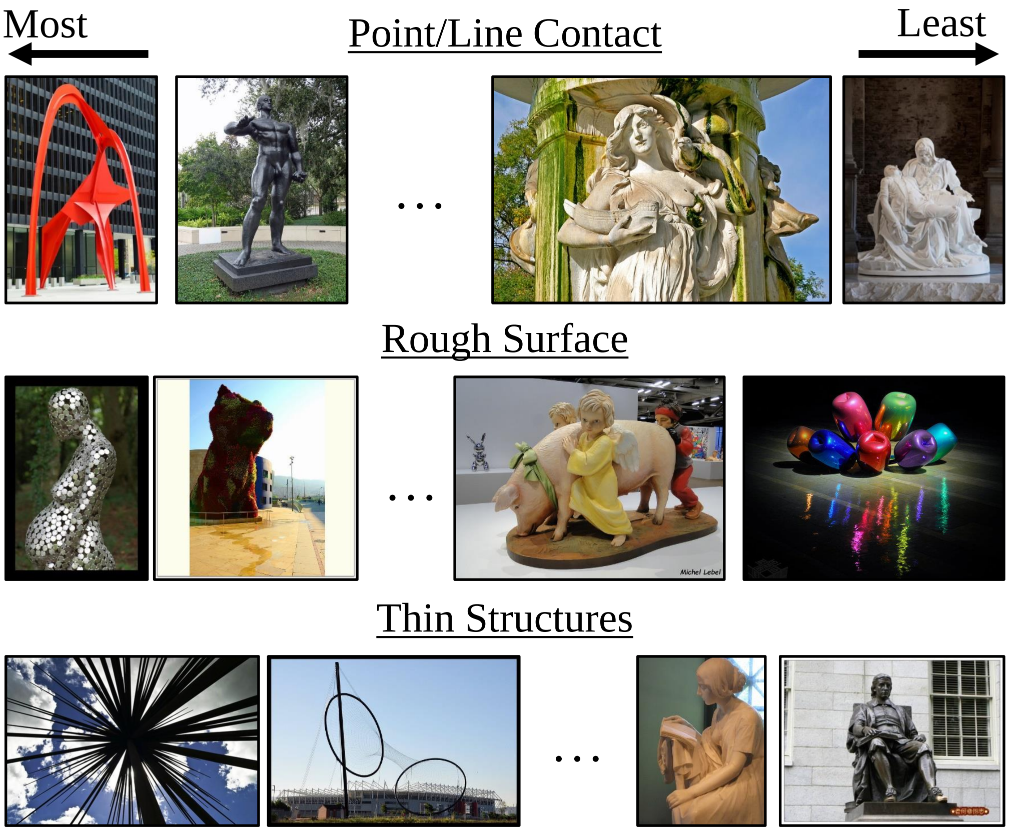


Figure 6.8: Test images sampled at the top, 95th, 5th percentiles and lowest percentile with respect to three attributes.

verts the depthmap into a representation amenable for fine-tuning a CNN; in this case, we learn the attribute network described in section 6.4.

Evaluation Criteria: Each method produces a prediction scoring how much the image has the attribute. We characterize the predictive ability of these scores with a receiver operator characteristic (ROC) over the Sculpture images test set. This enables comparison *across attributes* since the ROC is unaffected by class frequency. We summarize scores with the area under the curve.

Results: Fig. 6.7 shows predictions of all of the attributes on a few sculptures. To help visualize what has been learned, we show automatically sampled results in Fig. 6.8, sorted by the predicted presence of attributes. Additional results are given in the supplementary material.

We report quantitative results in Table 6.2. On an absolute basis, certain attributes, such as planarity and emptiness, are easier than others to predict, as seen by their average performance; the harder ones include ones based on symmetry and aspect ratio, which may require a global comparison across the image, as opposed to aggregation of local judgments.

In relative terms, our approach out-performs the baselines, with especially large gains

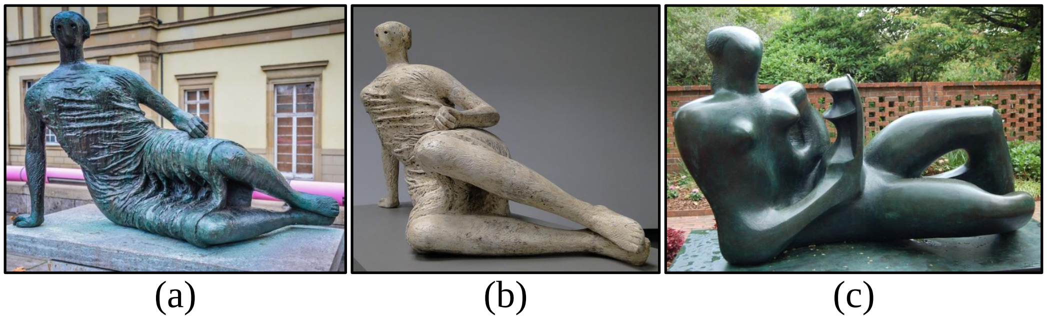


Figure 6.9: In mental rotation, the goal is to verify that (a) and (b) correspond and (a) and (c) do not. Roughness is a useful cue here.

on planarity, emptiness, and thinness. Note that reconstructing thin structures is challenging even with stereo pairs; an approach based on depth-prediction is likely to fail at reconstruction, and thus on attribute prediction. Instead, our system directly recognizes that the object is thin (e.g., Fig. 6.7 bottom). Fig. 6.8 shows that frequently, the instances that least have an attribute are the negation of the attribute: for example, even though many other sculptures are not rough, the least rough objects are especially smooth.

The system's mistakes primarily occur on images where it is uncertain: sorting the images by attribute prediction and re-evaluating on the top and bottom 25% of the images yields a substantial increase to 77.9% mean AUC; using the top and bottom 10% yields an increase to 82.6%.

Throughout, we fix our base representation to VGG-M [22]. Switching to VGG-16 [151] gives an additional boost: the mean increases from 72.3 to 74.4 and $1/3$ of the attributes are predicted with AUCs of 80% or more.

6.5.2 Mental Rotation

If we have learned about 3D shape, our learnt representation ought to encode or *embed* 3D shape. But how do we characterize this embedding systematically? To answer this, we turn to the task of mental rotation [149, 157] which is the following: given two images, can we tell if they are different views of the same object or instead views of different objects? This is a classification task on the two presented images: for instance, in Fig. 6.9, the task is to tell that (a) and (b) correspond, and that (a) and (c) do not.

Note, the design of the dataset has tried to ensure that sculpture shape is not correlated with location by ensuring that images of a particular work come from different locations (since multiple instances of a work are produced) and different materials (e.g., bronze and stone in Fig. 6.9).

We report four representations: (i) the 1024D embedding produced by our full network; (ii) the 4096D fc7 layer of the full network; (iii) the 4096D fc7 layer of the attribute-only network; (iv) the attribute probabilities themselves from the full network. If our attribute network is using actual 3D properties, then the attribute network's activations ought to work well for the mental rotation task even though it was never trained for it explicitly. Additionally, the attributes themselves ought to perform well.

Baselines: We compare our approach to (i) the pretrained FC7 from the initialization of

Table 6.3: AUC for the mental-rotation task. Both variants of our approach substantially out-perform the baselines.

	Full Network			Attr. Only	Pretrained	IFV	
	Emb.	FC7	Attr	FC7	FC7	[118]	[4]
All	**92.3**	90.7	81.9	89.8	88.9	78.0	74.4
Hard	**86.9**	84.1	76.4	82.5	80.0	57.3	61.9

the network and to (ii) IFV [129] over the BOB descriptor [4] that was used to create the dataset and dense SIFT [118]. The pre-trained FC7 characterizes what has been learned; the IFV representations help characterize the effectiveness of the attribute predictions on their own. We use the cosine distance throughout.

Evaluation Criteria: We adopt the evaluation protocol of [84] which has gained wide acceptance in face verification: given two images, we use their distance as a prediction of whether they are images of the same object or not. Performance is measured by AUROC, evaluated over 100 million of the pairs, of which 0.9% are positives. Unlike [84], positives in the same viewpoint cluster are ignored: these are too easy decisions.

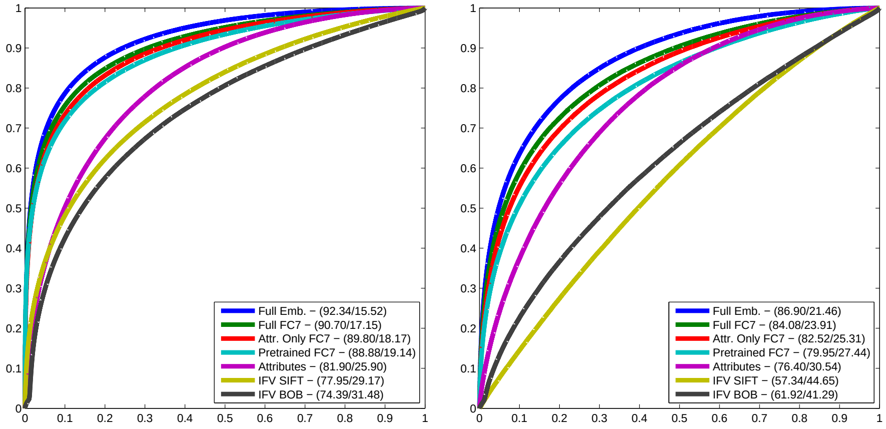
We further hone in on difficult examples by automatically finding and removing easy positives which can be identified with a bare minimum image representation. Specifically, we remove positive pairs with below-median distance in a 512-vocabulary bag-of-words over SIFT representation. This yields a more challenging dataset with 0.3% positives. As mentioned in Sec. 6.3 artists often produce work of a similar style, and the most challenging examples are often pairs of images from the same artist (which may or may not be of the same work). We call the standard setting *Easy* and the filtered setting with only hard positives *Hard*.

Results: Table 6.3 and Fig. 6.10 show results for both settings. By themselves, the 12D attributes produce strong performance, 3-4% better than IFV representations. The attribute-only network improves over pretraining (by 0.9% in easy, 2.5% in hard), suggesting that it has learned the shape properties needed for the task. The full system does best and substantially better than any baseline (by 3.4% in easy, 6.9% in hard). Relative performance compared to the initialization consistently *improves* for both the full system and the attribute-only system when going from Easy to Hard settings, providing further evidence that the system is indeed modeling 3D properties.

6.5.3 Object Characterization

Our evaluation has so far focused on sculptures, and one concern is that what we learn may not generalize to more everyday objects like trains or cats. We thus investigate our model's beliefs about these objects by analyzing its activations on the PASCAL VOC dataset [41]. We feed the windows of the trainval set of VOC-2010 to our shape attribute model, and obtain a prediction of the probability of each attribute. We probe the representation by sorting class members by their activations (i.e., "which trains are planar?") and sorting the classes by their mean activations.

Per-image results: The system forms sensible beliefs about the PASCAL objects, as we show in Fig. 6.11. Looking at intra-class activations, cats lying down are predicted to have single, non-point contact as compared to ones standing up; trains are generally planar, except for



(a) Easy Setting

(b) Hard Setting

Figure 6.10: Mental rotation ROCs for easy and hard settings. In the legend, we report the AUC and EER for each method.

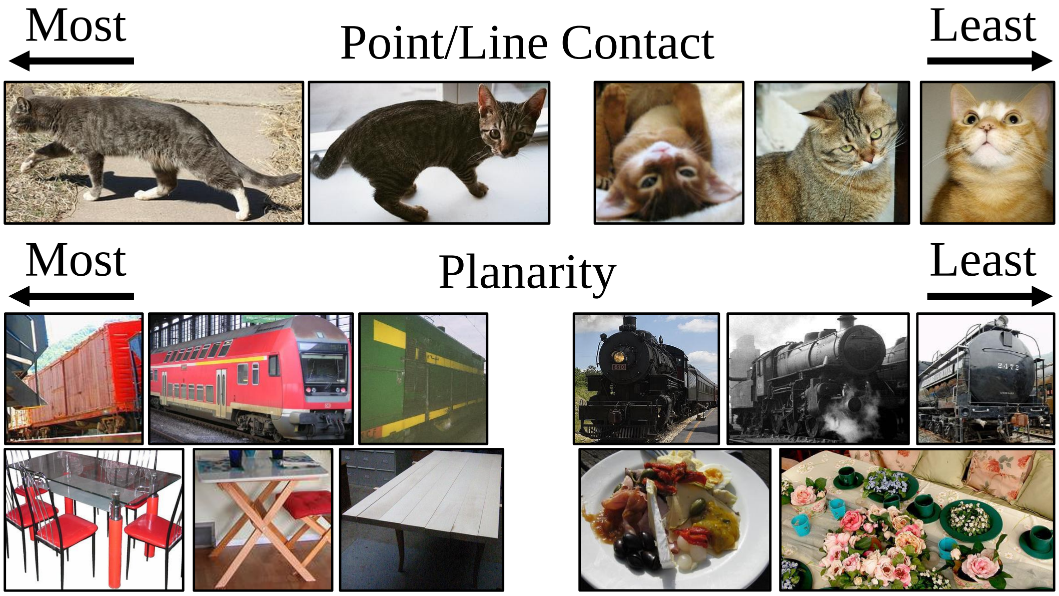


Figure 6.11: The top activations on PASCAL objects for Planarity and Point/Line Contact.

older cylindrical steam engines. Similarly, the non-planar dining tables are the result of occlusion by non-planar objects.

Per-category results: The system performs well at a category-level as well. Note that averaging over windows characterizes how objects *appear* in PASCAL VOC, not how they are prototypically imagined: e.g., as seen in Fig. 6.11, the cats and dogs of PASCAL are frequently lying down or truncated. The top 3 categories by planarity are bus, TV Monitor, train; and the bottom 3 are cow, horse, sheep. For point/line contact: bus, aeroplane, car

are at the top and cat, bottle, sofa are at the bottom. Finally, sheep, bird, and potted plant are the roughest categories in PASCAL and car, bus, and aeroplane the smoothest.
Discriminating between classes: It ought to be possible to distinguish between the VOC categories based on their 3D properties, and thus we verify that the predicted 3D shape attributes carry class-discriminative information. We represent each window with its 12 attribute probabilities and train a random forest classifier for two outcomes in a 10-fold cross-validation setting: a 20-way multiclass model and a one-vs-rest. The multiclass model achieves an accuracy of 65%, substantially above chance. The one-vs-rest model achieves an average AUROC of 89%, with vehicles performing best.

6.6 Discussion

We have shown that 3D shape attributes can be inferred directly from images at quite high quality. These provide a complementary view of 3D understanding to the surface normal-based approaches presented in the rest of the thesis: for instance, if we want to understand a chain-link fence, the particular normals are less important (and more difficult to estimate) compared to the fact that the fence is a thin structure that obstructs passage of a body but not hands. Additionally, these attributes, if localized, could serve as a way to constrain the surface normal estimation techniques: our approaches produce either unconstrained continuous predictions or constrained discretized predictions. By determining which objects in the scene are planar, 3D shape attributes could produce better reconstructions.

Chapter 7

Physical Constraints for 3D Structure

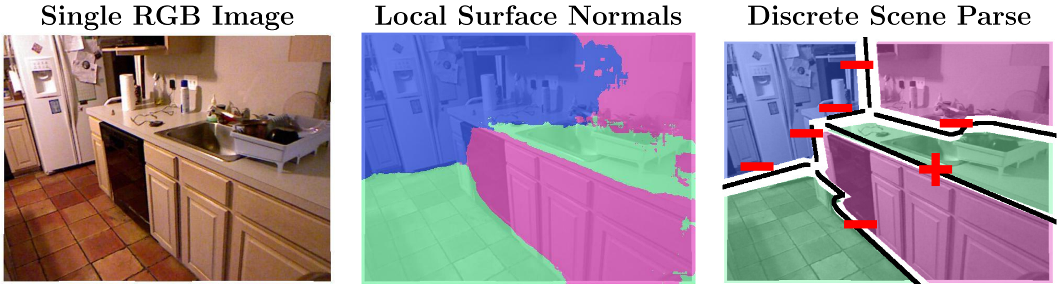


Figure 7.1: We propose the use of mid-level constraints from the line-labeling era and a parameterization of layout to "unfold" an interpretation of the scene as large planar surfaces and the edges that join them. In contrast to per-pixel normals, we return a discrete parse in terms of surfaces and the edges between them in the style of Kanade's Origami World .

Chapters 4 and 5 introduced work on finding primitives that can provide local evidence of 3D structure. Unfortunately, looking at local image evidence can only take one so far: it is generally impossible, for instance, to recognize the orientation of a white wall from purely local evidence alone. Instead, one must recognize things featureless things like white walls and beige countertops by first recognizing the easy to-recognize parts of the scene and then propagating information via constraints.

Most past efforts in geometric reasoning have used either local domain-agnostic constraints or global high-level constraints based on domain knowledge. For instance, CRF-based approaches include the constraint that regions with similar appearance should have similar orientation. These end up, however, enforcing little more than smoothness. This has led to high-level top-down constraints given by domain-specific knowledge. For example, most indoor approaches assume a Manhattan world in which the surface normals lie on three principal directions [26]. Second only to the Manhattan-world constraint is the cuboidal room constraint, in which the camera is assumed to be inside a cube and inference becomes predicting the cube's extent [74]. While this has been enormously influential, the camera-inside-a-box representation leaves the interior and most interesting parts of the scene, for instance furniture, uninterpreted. Recent work has aimed at overcoming this by finding volumetric primitives inside scenes, conventionally cuboids [63, 111, 147, 167], and in simple scenes such as the UIUC dataset of [74], cuboid representations have definitely increased robustness of 3D scene understanding. Nonetheless, cuboid-based object reasoning is fundamentally limited by the appearance based likelihood models and it is not clear that these approaches generalize well to highly cluttered scenes.

In this chapter, we introduce an alternate idea: while there have been great efforts and great progress in both low and high-level reasoning, we believe that one missing piece is mid-level constraints. Reasoning is not a one-shot process, and it requires constraints at different levels of granularity. For cluttered and realistic scenes, before we can go to cuboids, we need to a way to piece together local interpretations into large planar surfaces and link them together via edges. This work aims to address this problem.

These mid-level constraints linking together planes via convex and concave edges have been extensively studied in the past. There is a vast line-labeling literature (e.g., classic works [24, 85] as well as later ones [121, 154]); among these works, we were principally inspired by Kanade's landmark Origami World paper [94], which reasoned directly about surfaces as first-class objects and the edges between them rather than use volumetric objects. As systems, line-labeling efforts failed due to weak low-level cues and a lack of probabilistic reasoning techniques; however, they hold a great deal of valuable insight.

Inspired by these pioneering efforts, we introduce mid-level constraints based on convex and concave edges and show how these edges help link multiple surfaces in a scene. Our contributions include: (a) a generic framework and novel parameterization of superpixels that helps to incorporate likelihoods and constraints at all levels of reasoning; (b) the introduction of mid-level constraints for 3D scene understanding as well as methods for finding evidence for them; (c) a richer mid-level interpretation of scenes compared to local normal likelihoods that can act as a stepping-stone for subsequent high-level volumetric reasoning.

7.1 Background

Determining the 3D layout of a scene has been a core computer vision problem since its inception, beginning with Robert's ambitious 1963 "Blocks World" thesis [135]. Early work such as [24, 85] often assumed a simple model in which the perceptual grouping problem was solved and there were no nuisance factors. Thus, the 3D layout problem could be posed as constraint satisfaction over the visible lines. These methods, however, failed to pan out in natural images because the actual image formation process is much more noisy than was assumed.

After many decades without success, general 3D layout inference in relatively unconstrained images began making remarkable progress [31, 80, 81, 141] in the mid-2000s, powered by the availability of training data. This sparked a renaissance during which progress started being made on a variety of long-standing 3D understanding problems. In the indoor world, a great deal of effort went into developing constrained models for the prediction of room layout [74] as well as features [51, 112, 133] and effective methods for inference [30, 111, 147, 148]. While these high-level constraints have been enormously successful in constrained domains (e.g., less cluttered scenes with visible floors such as the datasets of [74, 171]), they have not been successfully demonstrated on highly cluttered scenes such

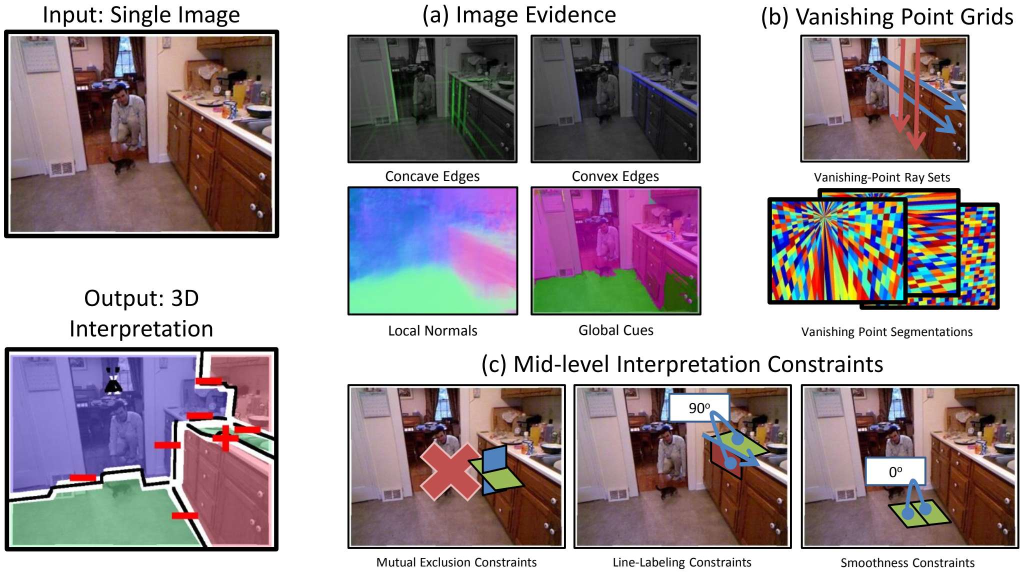

Input: Single Image
(a) Image Evidence
(b) Vanishing Point Grids
Concave Edges
Convex Edges
Vanishing-Point Ray Sets
Local Normals
Global Cues
Vanishing Point Segmentations
Output: 3D Interpretation
(c) Mid-level Interpretation Constraints
90º
0º
Mutual Exclusion Constraints
Line-Labeling Constraints
Smoothness Constraints

Figure 7.2: **Overview of the proposed approach.** (Left) We seek to take a single image of an indoor scene and produce a 3D interpretation of it in terms of surfaces and the edges (Convex: +, Concave: −) that join them. (Right) We accumulate evidence in the form of inferred surface normal discontinuities in the scene (convex blue, concave green) and local surface normals and room layout. (b) We formulate the problem as assembling a coherent interpretation of the scene from a collection of vanishing-point aligned grids. (c) This interpretation must respect constraints from observed edges that join segments as well as constraints such as mutual exclusion and smoothness.

as the NYU v2 Depth Dataset [150]. Indeed, on these cluttered scenes, it turns out to be difficult *even with depth* to find a variety of simple primitives such as cuboids [90,91,173], support surfaces [62], or segmentations [65]. We believe that one missing ingredient is effective mid-level constraints, and in this work propose such constraints in the form of line-labels. We emphasize that our goal is to complement prior high-level constraints: we envision a system in which all of these cues cooperate to produce an understanding of the scene at a number of levels of grouping in a hierarchical but feedback manner. This work acts as a stepping stone towards this vision, and introduces a framework for layout estimation that we demonstrate can easily integrate constraints from a variety of levels.

Other work in single-image layout prediction has drawn inspiration from classic line-labeling approaches, but has focused on the task of finding occlusion boundaries [83, 89, 116]. While important, occlusion boundaries only provide a 2.1D sketch (e.g., like [125]) and provide no information in a single image about the surface orientation without their complementary normal discontinuity labels. In this work, we focus on this other class of labels from classic line-labeling approaches, namely convex and concave edges. These have been applied in the context of stereo [168], but have largely been ignored in the single-image layout community, apart from work on shape recovery [97] using hand-marked folds.

7.2 Approach

In this work, our goal is: given a single image, group pixels into planes, and infer the orientations of these planes, the convex/concave nature of edges and how the planes link together via edges to form objects. Similar to previous indoor scene understanding approaches, we assume a Manhattan world, which restricts the orientation of planes to three principal directions. Generally, this constraint is implicitly encoded by first grouping pixels into regions and then solving the surface normal problem as a 3-way classification problem. Our key idea is to reformulate the problem and solve the grouping and classification problem jointly by using top-down superpixels. Given the estimated vanishing points, we determine three possible grids of superpixels aligned with these vanishing points and the problem of classification becomes finding the "active" grid cell at every pixel.

However, inferring the active grid cells using image evidence, like most single image 3D understanding tasks, is a severely underconstrained problem. Therefore, we include a variety of constraints based on (a) mutual exclusion (b) appearance and smoothness; (c) convex/concave edge likelihoods; (d) global room layout constraints. Some constraints are enforced as a unary, while constraints such as (c) are binary in nature and therefore our objective is a quadratic with mutual exclusion constraints. Additionally, the superpixel variables must be integer if one wants a single interpretation of each pixel (1 corresponding to active, 0 to non-active) and therefore the quadratic problem is NP-hard. We propose to optimize both the integral and relaxed problems: while discrete solutions are themselves rich inferences about the scene, we believe the relaxed solutions can act as inputs to higher-level reasoning processes.

We formalize our parameterization of the problem in Section 7.3 and discuss how we combine all the available evidence into one model in Section 7.4. Finally, we introduce an approach to finding surface normal discontinuities in Section 7.5.

7.3 Parameterization

The first step in our approach is estimating the vanishing points and creating grid cells in 3 principal directions. These act as superpixels defined by geometry rather than appearance. These grids are generated by sweeping rays from pairs of vanishing points, as shown in Fig. 7.3. The orientation of cells in the grids is defined by the normal orthogonal to the two generating vanishing points. Thus, a cell not only defines a grouping but also an orientation. Therefore any interpretation of the scene in terms of Manhattan-world surface normals that respects this grid can be represented as a binary vector $\mathbf{x}$ encoding which grid cells are active. To illustrate this, we show the likelihoods for the ground truth over grid cells in Fig. 7.3 (c). This formulation generalizes many previous parameterization of the 3D layout problem, for instance the parameterization proposed in [74, 147]. As we demonstrate with our potentials, our parameterization enables the easy arbitration between beliefs about layout encoded at every pixel such as [51, 98] and beliefs encoded parameterically, such as room layouts or cuboids [74, 111, 147]. Note that our grids overlap, but only one grid cell can be active at each pixel location; we enforce this with a mutual exclusion constraint.

The first step in our approach is to estimate the vanishing points and create grid cells in 3 principal directions which act as superpixels defined by geometry rather than appearance. These grids are generated by sweeping rays from a pair of vanishing points, as shown in Fig. 7.3. The orientation of segments in the grids is defined by the normal orthogonal to the

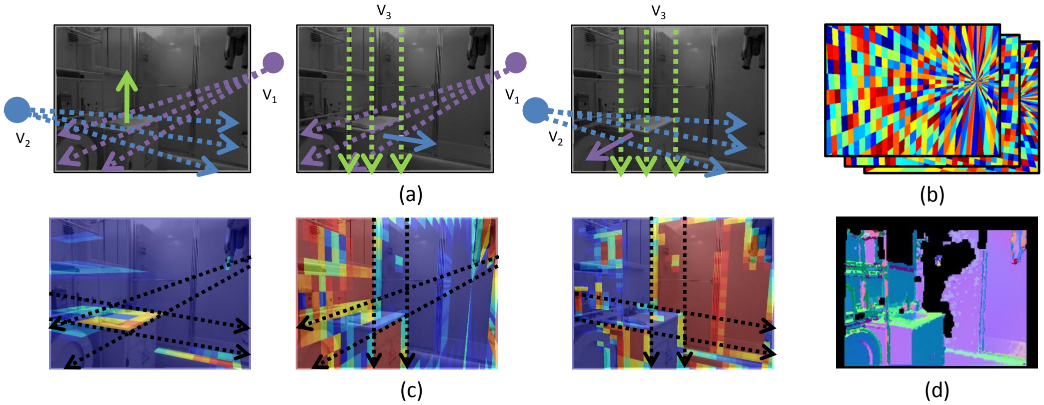


Figure 7.3: Parameterization of the method. (a) We sweep rays (dotted lines) from each vanishing point, defining a pencil of lines. The intersection of two pencils of lines defines a superpixel with normal (solid line) perpendicular to the normals of the generating vanishing points. (b) We represent layout by activations of these superpixel grids. (c) We show the likelihoods on each grid cell for the ground truth surface normals (d).

two generating vanishing points. Thus, a grid cell not only defines a grouping but also gives us an orientated plane. Therefore any coarse interpretation in terms of Manhattan-world surface normals can be represented as a binary vector $\mathbf{x} \in \{0,1\}^n$ encoding which grid cells are active, and thus this formulation generalizes many previous parameterization of the 3D layout problem inference, for instance the parameterization proposed in [74, 147]. Because we have overlapping grids, only one grid cell can be active at each pixel location; this is enforced as a mutual exclusion constraint. To illustrate the parameterization, we show the likelihoods in (c) over the grid cells for the ground truth in Fig. 7.3.

7.4 Unfolding an Interpretation

We now present how we combine our image evidence to find an interpretation of the scene. We first explain how we obtain surface normal likelihoods and use them as unaries to provide evidence for grid cells in Sec. 7.4.1. We then explain how we can enforce pairwise constraints on these grid cells given edge evidence in Sec. 7.4.2. Finally, we introduce a binary quadratic program that arbitrates between these cues and constraints to produce a final interpretation of the scene in Sec. 7.4.3.

7.4.1 Unary Potentials

The first kind of cue for finding whether grid cell i is active ($x_i = 1$) or not ($x_i = 0$) is local evidence at the grid cell location. In this work, we use two complementary cues based on techniques for inferring evidence of surface normals and transform them into potentials that capture how much we should prefer x_i to take the value 1. Recall that every grid cell represents not only a grouping but also an orientation, and therefore one can easily convert between likelihoods for orientation at a location and likelihoods of each grid cell being activated. This relationship is illustrated qualitatively in Fig. 7.4.

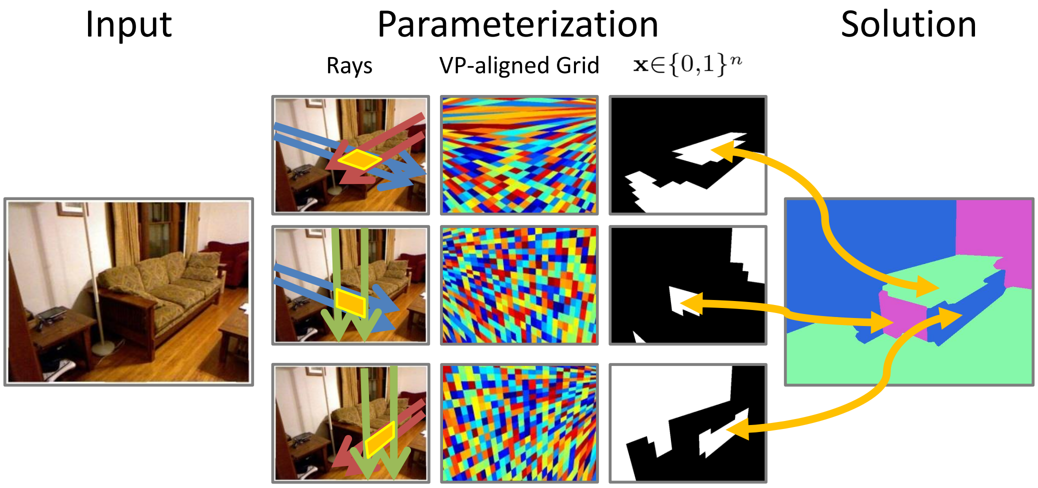


Figure 7.4: Given a set of vanishing point grids, our variable encodes a vanishing-point-aligned orientation map in terms of which cells in which grid are active.

Local evidence: A wide variety of approaches have been proposed for estimating local surface normal likelihoods from image evidence. We adopt the top-performing approach for doing this, Data-driven 3D Primitives [51] (3DP), which builds a bank of detectors that are associated with local surface configurations. At test time, the method convolves the bank with the image at multiple scales and transfers the associated local surface configuration to the test image wherever each detector has a high response. We soft-assign each pixel to each grid, producing a probability map of each orientation over the image. The local evidence potential of a grid cell i $\phi_{\text{local}}(i)$ is the probability of its orientation averaged over its support.
Global room-fitting evidence: Global room fitting constraints have been important and successful in the single image 3D understanding community. These seek to model the room as a vanishing-point-aligned box. We run the room-fitting method of Hedau et al. [74], which produces a ranked collection of 3D cuboid room hypotheses, where each wall corresponds to one of our grids' directions. At every pixel, we build a histogram of each direction, weighted by the inverse of the rank of the room and suppressing pixels predicted as clutter. The room-fitting evidence potential of a grid cell i $\phi_{\text{room}}(i)$ is the frequency of its orientation averaged over its support.

7.4.2 Binary Potentials

The second kind of cue for whether a cell i is active or not comes from considering it in conjunction with its neighbors. We use binary potentials characterizing preferences for pairs of grid cells. These operate within the same grid (i.e., on cells of the same orientation) and across grids (i.e., on cells with different orientations). These allow us to probabilistically integrate mid-level constraints via convex and concave edges. In this section, we describe our potential for achieving this as well as a standard appearance smoothness potential.
Line-labeling: The presence of a convex or concave edge tells us not only that a discontinuity may exist, but also what sorts of labels can occur on either side. For instance, in

Manhattan-world scenes, a convex edge at top of a counter tells us there is a horizontal surface above and a vertical surface below. Because this edge constrains the labels of adjoining surfaces, it is much more powerful than a simple smoothness term, which would only permit a labeling change at the edge.

We therefore include a potential that combines inferred convex and concave edges with a dictionary of surface configurations to reward interpretations of the scene in terms of grid cell activations that match our assumptions and available image evidence. We present a basic method for obtaining evidence of convexity and concavity in Section 7.5, but our potential is agnostic to the source of evidence.

We build an explicit enumeration of arrangements in the image plane that satisfy observing a scene formed by continuous Manhattan-world aligned polyhedra, e.g., a concave edge joining two vertical surfaces with the rightwards facing surface on the left. Some scenes may not satisfy our assumptions about the world and our image evidence may be wrong, and we therefore do not make hard decisions as in past line-labeling work [24, 85, 94], but instead form a potential encouraging interpretations that agree with our beliefs. Specifically, given two grid cells with different orientations, we can determine what edge we expect to see in our dictionary, and reward the mutual activation of the two grid cells if we see that edge. We use the potential $\psi_{\text{line}}(i,j) = \exp(-\beta_{line} e_{i,j}^2)$ where $e_{i,j}$ is the inferred probability of that edge from image evidence (i.e., mean image evidence over the edge joining two superpixels). We compute this potential over adjacent pairs of grid cells (i.e., sharing a vanishing point ray) but with different orientations. We compute this separately for convex and concave edges, letting the learning procedure decide their weight.

Smoothness: Adjacent and similar looking parts of the scene should generally have similar labels. As is common in the segmentation literature, we use a Potts-like model: we compute color histograms over LAB space (10 bins per dimension) for grid cells i and j, yielding histograms h_i and h_j; the potential is $\psi_{\text{smooth}}(i,j) = \exp(-d(h_i,h_j)^2)$, where d is the χ^2 distance. We compute the potential over adjacent grid cells with the same orientation, rewarding similarly colored regions for having similar orientation.

7.4.3 Inference

We need to resolve possibly conflicting potentials and infer the best interpretation of the scene given the available evidence. Mathematically, we formulate this as an optimization over a vector $\mathbf{x} \in \{0,1\}^n$, where each x_i represents whether grid cell i is active and where $\mathbf{x}$ contains the grid cells from all grids.

Our unary potentials $\{\mathbf{u}_i\}$ and binary potentials $\{\mathbf{B}_j\}$ are collated as a vector $\mathbf{c} = \sum_k \lambda_k \mathbf{u}_k$ and matrix $\mathbf{H} = \sum_l \alpha_l \mathbf{B}_l$ respectively, where c_i and $H_{i,j}$ respectively represent the costs of turning grid cell i on and the cost of turning both grid cell i and j on. Since two active overlapping cells imply that their pixels have two interpretations, we add a mutual-exclusion constraint. This is enforced on cells i and j that are on different grids and have sufficient overlap ($|\cap|/|\cup| \geq 0.2$ in all experiments). This can be formulated as a linear constraint $x_i + x_j \leq 1$. Finally, since our output is in the image plane and our cells are not all the same size, we weight the unary potentials by their support size and binaries by the minimum size of the cells involved.

Our final optimization is a binary quadratic program,

$$\underset{\mathbf{x} \in \{0,1\}^n}{\arg\max}\, \mathbf{c}^T \mathbf{x} + \mathbf{x}^T \mathbf{H} \mathbf{x} \quad \text{s.t. } \; \mathbf{A}\mathbf{x} \leq \mathbf{1}, \tag{7.1}$$

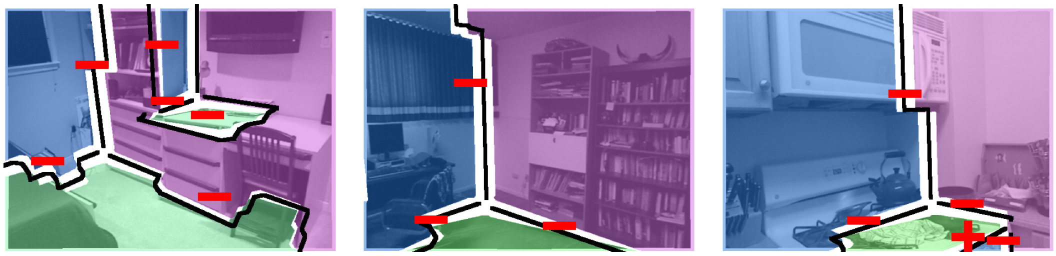


Figure 7.5: Selected surface connection graphs of scenes automatically obtained from a single image. The scene is interpreted as a collection of large vanishing-point-aligned regions with edges between them (convex: $+$, concave $-$).

where $\mathbf{A}$ stacks the mutual-exclusion linear constraints. Inference of this class of problems is NP-hard; we obtain a solution with the Gurobi solver, which first solves a continuous relaxation of Eqn. 7.1 and then performs a branch-and-bound search to produce an integral solution. The relaxed solution also acts as an updated belief about the scene, and may serve as a cue for the next layer of scene reasoning. We learn trade-off parameters $\{\lambda_k\}$, $\{\alpha_l\}$ by grid-search over a grid for each of the five potentials on a reduced set of images in the training set.

7.4.4 Surface Connection Graphs

Our approach can also produce surface connection graphs (SCGs) in the style of Kanade's Origami World [94]. Specifically, we take our interpretation according to Eqn. 7.1, take the resulting scene fragments (i.e., via connected components) and decode the relationship (convex/concave) between pairs of adjacent fragments. To do this, we use our scene formation assumption (contiguous Manhattan-world polyhedra). Parsing failures occur at configurations that are impossible in our dictionary (e.g., vertical-atop-other-vertical, caused by occlusion) or large pieces that are physically impossible. Jointly reasoning about all classic labels $+/-/\!\!\rightarrow\!\!\!-/$ remains a topic for future research in the single-view case. We note that it is, in principle, possible to similarly attempt to decode the Manhattan-world interpretations of the 3D Primitives; however, these results have many artifacts, and single large surfaces are fragmented into many disjoint and improperly shaped regions. The fragments of these SCGs can act as more useful information for future scene reasoning than simple per-pixel probabilities of orientation. We show a number of such results in Fig. 7.5.

7.4.5 Implementation Details

We now present a few important details on the origami world model:
Avoiding $\mathbf{x} = \mathbf{0}$: In this formulation, one must be careful to ensure that the optimal $\mathbf{x}$ is not the trivial solution $\mathbf{x} = \mathbf{0}$ (i.e., no segment is on). For the sorts of potentials used in this work, this occurs when the unary rewards for turning on a segment (e.g., because there is good local evidence for it) are outweighed by the incompatibility penalties. In practice, this occurs rarely.
Relaxed solutions: Our chosen solver, Gurobi, does not give programmatic access to the continuous relaxation it uses internally to initialize its branch-and-bound search. We instead use its continuous solver (solving for $\arg\min$); due to programmatic constraints, we

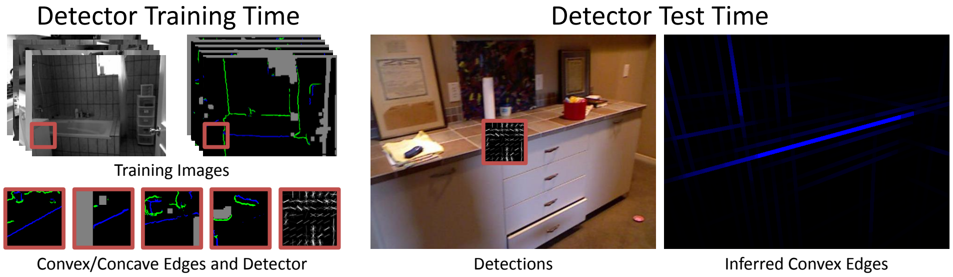


Figure 7.6: An illustration of our approach for finding surface normal discontinuities. At training time, we automatically build a collection of surface normal discontinuity maps (convex blue, concave green, missing data gray). At test time, we run a bank of 3D primitive detectors in the image; these detectors are trained to recognize a set of patches. We transfer the convex and concave edge patches associated with each detector at high responses. These noisy responses are smoothed and globalized over our vanishing-point aligned grids.

pass in $(\mathbf{H}+d\mathbf{I})$ for some small d to ensure a PSD Hessian and $(\mathbf{c}+d\mathbf{1})$ to ensure $\mathbf{x} \neq \mathbf{0}$. The resulting relaxed solution $\mathbf{x} \in [0,1]^n$ is difficult to interpret directly due to low dynamic range. We instead compute preferences for each direction at every pixel, subtract off the minimum, and normalize the values to sum to 1.

Grid holes: Readers might notice that the grid has holes where segments of three directions meet. We fill these with nearest-neighbor interpolation post-inference.

Number of grid cells: Throughout the work, our grid cells are generated by 32 rays from exterior vanishing points and 64 interior vanishing points.

Vanishing points: we use the vanishing point detector introduced in [74].

Implausible cells: some grid cells near vanishing points and the horizon line correspond to unlikely viewed surfaces (e.g., an enormous plane at just the right angle). Just as in classic work [94,154], we assume no accidental views and softly suppress these with an additional unary potential on the aspect ratio of the cell and the ratio of cell to bounding box, converted into a per-image quantile and then passed through $\exp(-x^2/\sigma^2)$ for $\sigma = 2\%$.

7.5 Finding Line Labels

Our method needs a source of convex and concave edges in a scene. In this section, we describe a simple method for obtaining them. We produce surface normal discontinuity maps from depth data and adapt the 3D Primitives approach [51] to transfer oriented surface normal discontinuities.

We begin by generating surface normal discontinuity labels for our method, adopting a procedure similar to [65]. We sweep a half-disc similar to [6] at 8 orientations at 7 scales over the image to get cross-disc normal angles at every pixel and orientation. These are noisy at each scale, but the stable ones (i.e., low variance over all scales), tend to be high quality. Example labels are shown in Fig. 7.6.

Given this data, we use a simple transfer method to infer labels for a new scene with a bank of 3D primitive detectors from [51]. Each detector is associated with a set of bounding boxes corresponding to the locations on which the detector was trained. In the original approach, the surface normals in these boxes were transferred to new images. Instead of transferring surface normals, we transfer the normal discontinuity label, separating by type (convex/concave) and orientation (8 orientations). Thus edge probabilities only accumulate if the detections agree on both type and orientation. At every pixel, the empirical frequency of normal discontinuity labels gives a probability of each edge type at each orientation. This is complementary to the local evidence unary: at the corner of a room, for instance, while a set of detectors may not agree on the specific surface normal configuration, they might agree that there is a concave edge.

7.6 Experimental Evaluation

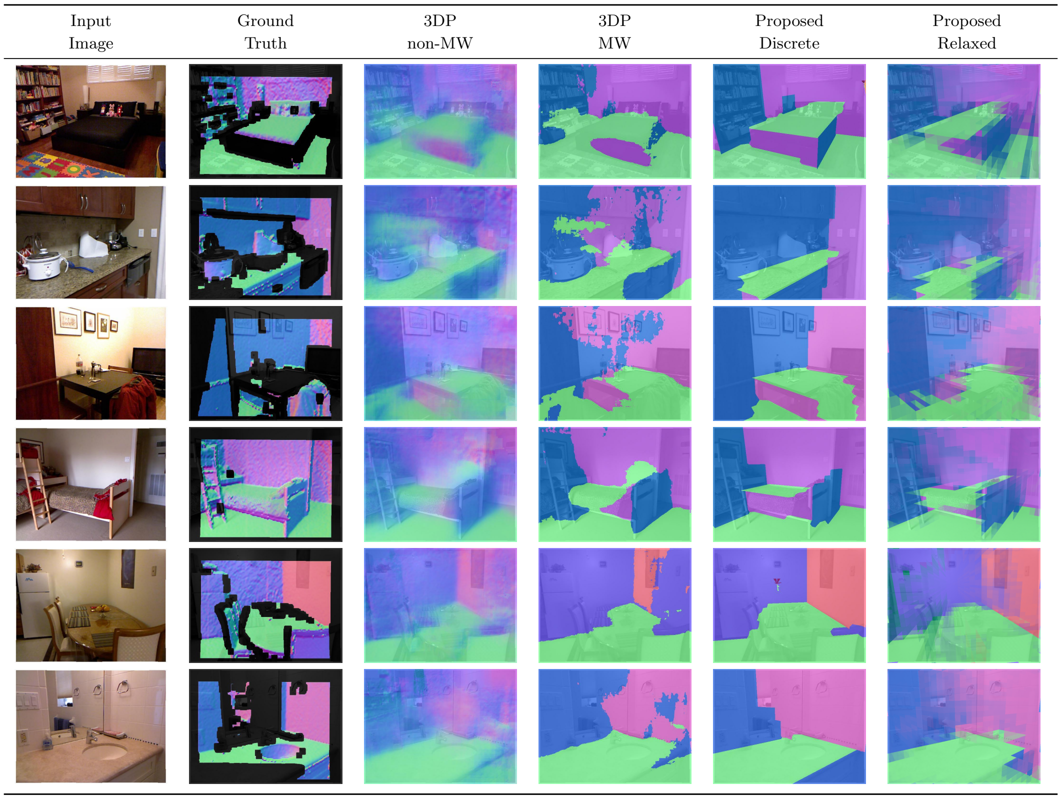


Figure 7.7: Selected results on the NYU Dataset comparing our approach to the 3D Primitive local likelihood model in non-Manhattan-world and Manhattan-world surface normal prediction. To help visualize alignment, we blend the predicted normals with the image.

Our output space is a richer interpretation of images compared to per-pixel prediction of surface normals. Evaluating this output in terms of line labelings or linkages of planes is not possible since there are no baseline approaches, ground-truth labels, or established evaluation methodologies. We therefore evaluate one of the outputs of our approach, surface normals, for which there exist approaches and methodologies. We adopt the setup used by the state-of-the-art on this task [51], and evaluate on the challenging and highly cluttered NYU v2 dataset [150].

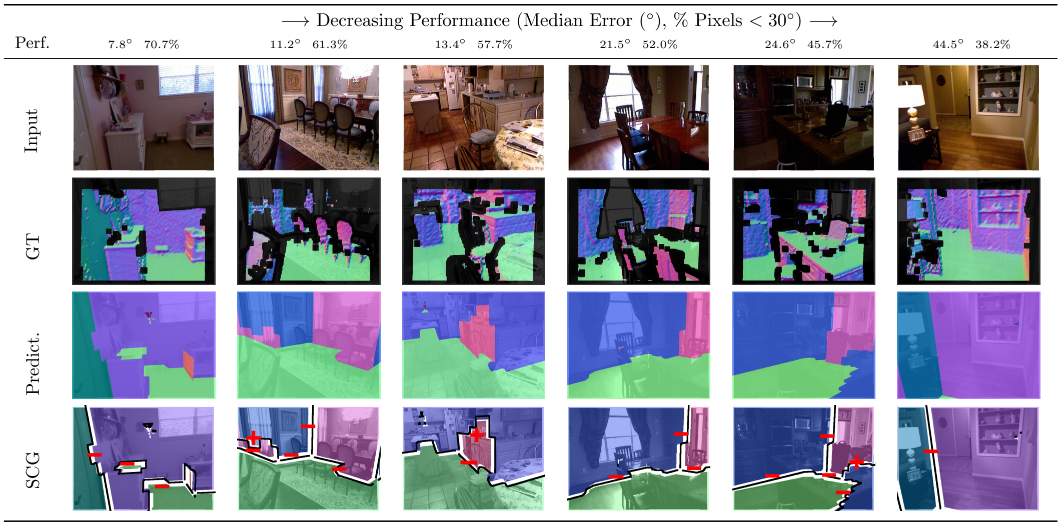


Figure 7.8: Results automatically sampled across the method's performance range. Performance reported as median error and % Pixels $< 30°$. Results were sorted by mean rank over all criteria and six results were automatically picked to evenly divide the list.

Table 7.1: Results on NYU v2 for per-pixel surface normal prediction. Our approach improves over Manhattan-world methods in every evaluation metric.

	Summary Stats. (Lower Better)			% Good Pixels (Higher Better)		
	Mean	Median	RMSE	11.25°	22.5°	30°
	Manhattan-world Techniques					
Proposed	**35.1**	**19.2**	**48.7**	**37.6**	**53.3**	**58.9**
Fouhey et al. [51]	36.0	20.5	49.4	35.9	52.0	57.8
Hedau et al. [74]	40.0	23.5	54.1	34.2	49.3	54.4
Lee et al. [112]	43.3	36.3	54.6	25.1	40.4	46.1

7.6.1 Experimental Setup

Training and testing: We adopt the setup used in Chapter 4.

Quantitative Metrics: We quantitatively evaluate the surface normal aspect of our approach using the same metrics introduced in Chapter 4. However, we strongly believe that the existing per-pixel quantitative metrics for this task are sometimes misleading. For instance, in Fig. 7.7, row 1, our method does worse than [51] on mean and median error, even though it conveys the cuboidal nature of the bed more precisely and segments it into three faces. Note that each metric characterizes a different aspect of performance, not all of which are equally desirable.

Baselines: Our primary point of comparison is 3DP, which we introduced in [51] and Chapter 4. In particular, the informative comparison to make is with the Manhattan-world version of 3DP as comparisons between non-Manhattan-world and Manhattan-world methods are difficult to interpret. This is because Manhattan-world methods generally produce re-

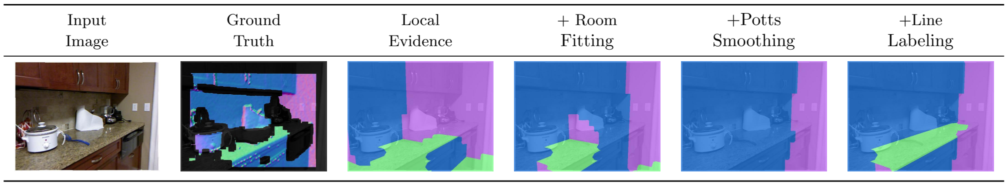


Figure 7.9: Qualitative analysis of the method components. Right-to-left: We start with local, then global unaries, followed by smoothness then line-labeling binaries.

Table 7.2: Component-wise analysis on NYU v2 [150]: we report results with parts of the full system removed to analyze the contributions of each method to overall performance.

	Mean	Median	RMSE	11.25°	22.5°	30°
Unaries Only	36.0	19.9	49.6	36.8	52.4	58.0
Smoothness Only	35.4	19.7	48.9	37.4	53.0	58.5
Full Method	**35.1**	**19.2**	**48.7**	**37.6**	**53.3**	**58.9**

sults that are nearly correct (correct vanishing point) or off by 90° (incorrect one). This results in error distributions that are qualitatively different; however, when one projects the distribution to a summary statistic, this information is lost. Unfortunately then, the preferred method comes down to implicit preferences in the metric used: the Manhattan-world's bi-modal distribution is rewarded by some metrics (% Good Pixels) and penalized by others (mean, RMSE).

7.6.2 Results

Predicting Surface Normals: We show selected qualitative results in Fig. 7.7 that illustrate the contributions of our method, as well as an automatically selected performance range in Fig. 7.8. Our method can often accurately capture object boundaries, especially in comparison with the normal-snapping approach described in [51], which produces noticeable spotting or bending artifacts. Our top-down parameterization mitigates this issue by constraining the space of interpretations, resulting in more plausible explanations. Our mid-level constraints help with the recovery of hard-to-see surfaces, such as surfaces on top of counters or beds (rows 1, 4) or sides of cabinets (row 6). These small surfaces are frequently smoothed away by the Potts model, but are recovered when our line labeling potentials are used.

We report quantitative results in Table 7.1. Our method outperforms the state-of-the-art for Manhattan-world prediction in every metric. This is important since each metric captures a different aspect of performance. It also does better than the non-Manhattan-world methods in all metrics except the ones that heavily penalize Manhattan-world techniques, mean and RMSE; nonetheless, even on mean error, it is second place overall. Although our system outperforms the state-of-the-art, we stress that per-pixel metrics must be considered carefully.

Qualitative scene parses: We show some surface connection graphs in Fig. 7.5, 7.8. The SCGs characterize the overall shape of the scene well: note the countertops and the overall

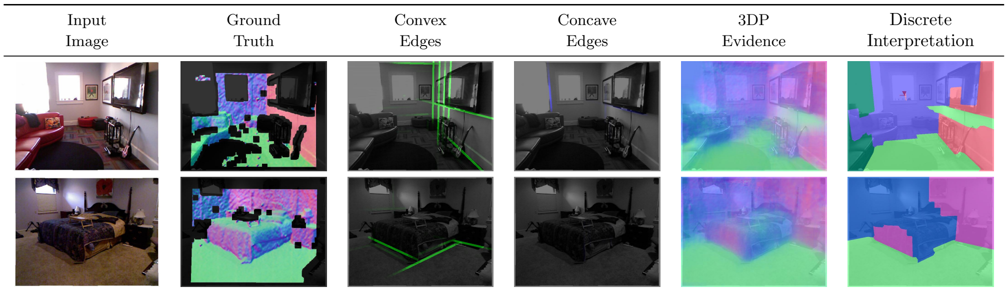


Figure 7.10: **Failure modes and limitations:** (Top) Local evidence can be misleading. An edge is seen below the TV, and our method "folds" its interpretation accordingly. (Bottom) We model only unary and binary relationships between cells; higher order reasoning may allow the recognition of the top of the bed from its sides.

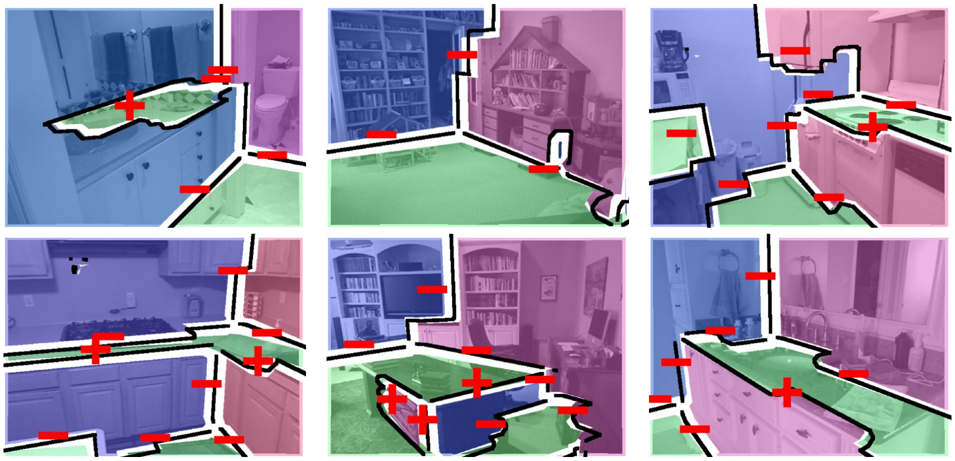


Figure 7.11: Additional surface connection graphs produced automatically by our method.

decrease in spotting artifacts compared to 3DP. Of course, some details are lost in the interpretation and because occlusion is not modeled, some planar pieces are actually multiple planes (bottom, middle).

Ablative analysis: We now describe experiments done to assess the contributions of each component. We show an example in Fig. 7.9 that characterizes how each part tends to change the solution: an initial shape is captured by the local evidence potential and is improved by the room fitting potential. Smoothness potentials remove noise, but also tend to remove small surfaces. The line-labeling potentials can enable the better recovery of the counter top by counterbalancing the smoothness. We show quantitative results in Table 7.2: each step contributes; our line-labeling potentials reduce the median error the most.

Failure modes and limitations: We report some failure modes and limitations in Fig. 7.10. Our primary failure mode is noisy evidence from inputs. These tend to correspond to mistaken but confident interpretations (e.g., the fold preferred by our model in the first row).

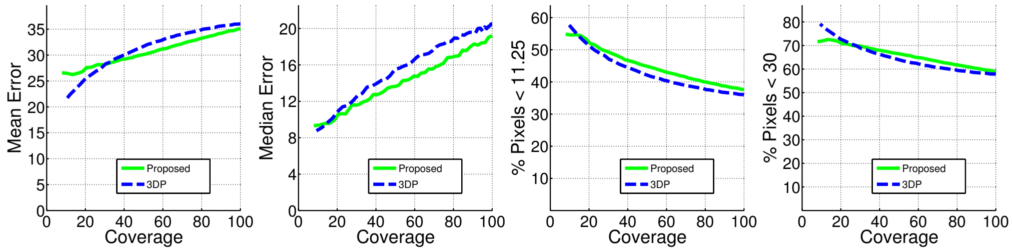


Figure 7.12: Performance vs. coverage curves for the system and the next-best method, 3DP. Each plot quantifies accuracy in a metric against fraction of pixels predicted.

Sometimes layouts inferred by our system violate high-level constraints: in the second row, for instance, our interpretation is unlikely. By reasoning about the proposed pieces, we can reject it without the surface on top necessary to make it plausible. This is consistent with our vision of the 3D inference process: mid-level constraints are valuable, not that they are the end of the scene interpretation story. Rather than solve all problems at once, we must pass updated evidence to subsequent reasoning.

7.7 Discussion

This chapter presented a method that infers a discrete parse of a previously unseen scene. This method offers a way to organize a scene into large planar surfaces and the edges between them, and improves the state-of-the-art in Manhattan-world surface normal prediction. Additionally, we presented a technique for obtaining discontinuities such as convex and concave Irrespective of the particular cues we have used, this parameterization and formulation enables the easy arbitration among constraints and cues at a variety of levels.

Chapter 8

The Broader Context

Chapters 4-7 form the contributions of our dissertation. However, they are best described in the context of a larger research agenda aimed at understanding scenes and objects in terms of physical properties, or the concrete properties of the scene, as well as functional properties, or what an agent can do in the scene.

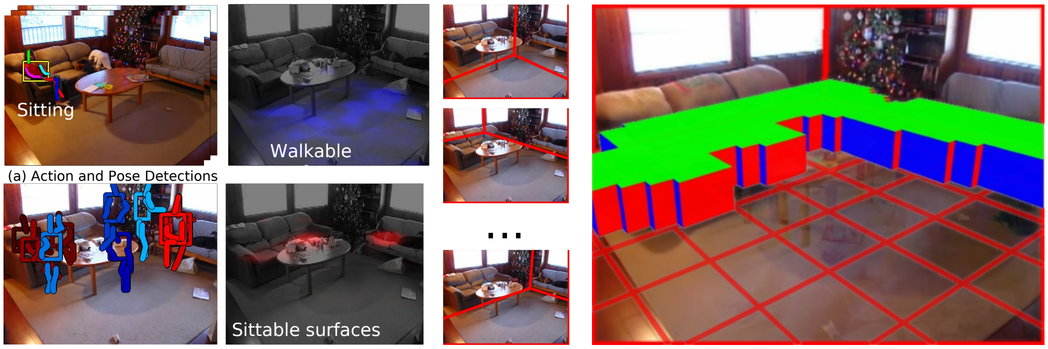
8.1 Functional Constraints on 3D Structure

This dissertation focused on physical objects and demonstrated how to recover their 3D structure. However, the world most of us live in is not just a *physical* space, but also a *human-centric* space: the layouts and construction of our spaces are designed specifically with humans in mind. Many chapters of this dissertation have relied on this implicitly: the cues explored in Chapters 4 and 5 depend on the regularity of human scenes in terms of 3D structure and style and the constraints introduced in Chapter 7 leverage the structure of human scenes in terms of 3D structure.

This interaction between humans and their environments is part of a larger picture of understanding the world in terms of agents and how they can interact with it. Broadly, this is motivated by the work of Gibson [60] which sought to frame the understanding of the world in these terms. However, we only discuss our use of its implications to 3D here.

We have also explicitly taken advantage of the human-centric nature of scenes by exploring the idea that humans can act as a cue for single view geometry in [49,50]. In particular, our approach observed the poses of humans performing their daily behavior in scenes and used them as a cue for a single-view layout model related to the ones we have discussed in Chapters 4 and 7. This is achieved by reasoning about the 3D configurations implied by detected humans and using these to softly eliminate incompatible 3D hypotheses. For instance, if we are unsure about the extent of a room, seeing a person gives a lower bound on the extent of the room.

Our approach, illustrated in Fig. 8.1, takes as input a time-lapse video from a fixed camera and converts it into a 3D understanding of a scene. Our 3D understanding consists of a layout similar to [74,111], or a vanishing-point-aligned cuboid, as well as an estimate of occupied voxels within the cuboid. Our focus on time-lapses comes from their sampling of a wide variety of human poses over a short period of time: a 3 minute time-lapse may cover



(a) Action and Pose Detections

(b) Poses (potentially aggregated over time)

(c) Estimates of functional surfaces

(d) 3D room geometry hypotheses

(e) Final 3D scene understanding

Figure 8.1: **Overview of the functional constraints for 3D.** Given an input image or set of input images taken by a fixed camera, we estimate human poses in each image (a), yielding a set of human-scene interactions (b), which we aggregate over time (for time-lapses). We use these to infer functional surfaces (c) in the scene: sittable (red), walkable (blue). We simultaneously generate room hypotheses (d) from appearance cues alone. We then select a final room hypothesis and infer the occupied space in the 3D scene using both appearance and human action cues.

3 hours of human activity. The approach proceeds in three main stages: estimating human poses and accumulation of functional evidence; box-layout estimation; and space carving by humans.

Functional map estimates: For each frame, we detect where humans are performing each action from a set of 3 actions: walking, sitting, and reaching. We use an articulated pose estimator [169] as well as a deformable parts model [43] to obtain detections of humans and their poses. These are accumulated over time to yield a map of functional properties of the scene (i.e., where someone can walk, where someone can sit). Contemporary work [32], lead by Vincent Delaitre, showed how to use similar maps of human poses as a cue for scene semantics (i.e., this is a coffee table/couch/etc.).

Layout estimation: These functional maps place constraints on the overall layout of space: all true detections of people should be contained in the room. Of course, not all of our detections are correct and not all spaces are modeled perfectly by cuboids. We thus take a soft-approach and linearly weight three terms: (1) the learned appearance term of [74]; (2) a functional constraint penalty that tries to ensure that the room contains the detected humans; (3) a regularization term that accounts for the fact that people can only expand rooms.

Space carving: Once a room is estimated, we can then more precisely carve out the 3D space. Our core observation is that certain constrains imply different constraints: walking implies free space in the area that is being walked in; sitting implies occupied space beneath the pelvic joint. We backproject the estimated functional maps and do a simple voting scheme to estimate a probabilistic occupancy map.

Results: We show two results in Fig. 8.2 of processing time-lapses from a dataset of time-lapses collected from Youtube. We detect sitting people primarily on sofas, and walking

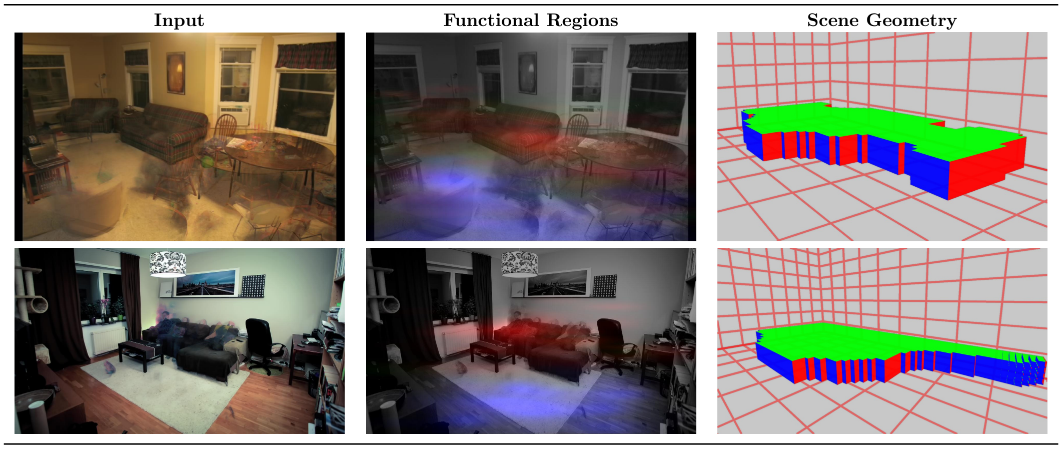


Figure 8.2: Selected results of the functionally constrained method. After estimating the functional regions, we combine them with appearance cues to obtain a 3D scene understanding

people in other parts of the scene. By making these observations, we can improve the layout and obtain an estimate of the volumetric occupancy of the scene. Quantitatively, we observed a improvement: on the extended time-lapse dataset of [32], using humans as a cue gave a 3% improvement, but more importantly almost always resulted in as-good-or-better results. Even in the case of single images containing as little as one human, we found that humans could consistently reliably performance.

Functional constraints without humans: One natural question might be: do we actually need real physical humans to use functional reasoning to improve 3D understanding? In [55], we demonstrated that this was not the case. After generating synthetic training on top of the NYU dataset by analyzing the point-clouds of NYUv2 [150], we showed that one could directly estimate the affordance properties such as “sittable” from images. These properties are useful on their own since they provide a more complete description of the scene. They also can also be used to produce better estimates of 3D structure: plugging in the estimates into the time-lapse system produced results within 1% of the system using actual humans. Additionally, using the predicted functional regions yielded a 3.1% gain in determining horizontal surfaces compared to the approach presented in Chapter 4.

8.2 Deep Learning for Surface Normals

One of the overarching themes in this dissertation has been to use physical knowledge in order to improve our recovery of 3D structure. This can take a number of forms. Typically this is in implicit forms that do not get in the way of learning: for instance, in the choice of output representation, such as normals, or in pre-and-post-processing steps, such as the rectification in Chapter 5, which apply to any learning technique. However, physical knowledge frequently plays an active role in the learning formulation and in providing constraints as in Chapter 7. One large change in computer vision during the dissertation has been the transition from shallow learning techniques with many design decisions but few parameters to deep learning techniques with fewer design decisions but many more pa-

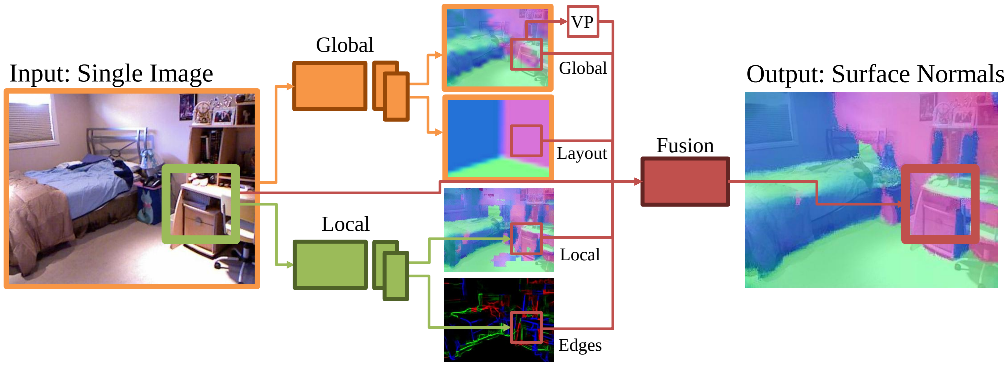


Figure 8.3: Overview of the deep learning approach

rameters. The overall philosophy of these approaches has tended to be to learn everything end-to-end, apparently leaving little room for these constraints.

In an effort led by Xiaolong Wang, we explored the extent to which deep-learning-based approaches can be reconciled with what we already know about 3D. The idea is that we incorporate our knowledge via multiple types of intermediate supervision and through constraints that we can apply to these intermediate results. This way, the CNN can do what it does best – map images to labels – and our knowledge is used, but does not get in the way of learning.

Our approach is summarized in Fig. 8.3, and we briefly describe each component below. Throughout we follow the quantization approach of Ladicky et al. [109], which reduces the constrained regression problem of normal estimation to a classification one.

Local Network: This network takes a small patch in the image as input, and predicts local normals similar to the approach of Chapter 4 as well as the classic line-labeling edges, similar to the method presented in Chapter 7. This network is applied in a sliding-window fashion and thus tends to be accurate in highly textured areas and inaccurate in poorly textured areas.

Global Network: This network takes as input the entire image and predicts a coarse layout in terms of normals as well as categorizing the image as one of a number of different types of rooms similar to [74]. This tends to give a good global sense of the scene but miss out on high-frequency details.

Fusion Network: Each network produces a collection of potentially conflicting interpretations and evidence. Additionally, we can apply constraints on top: for instance, we take the normal predictions of the global network and snap them to the nearest scene direction estimated by vanishing points. In order to reconcile all of the differing predictions, we treat their predictions as images. We train a final network to locally look at all the predicted evidence as well as the original input image and make a final prediction. The idea is that this network should learn rules like: if the region is textureless and the local and global regions disagree, propagate the predictions of the global network or the room layout.

Results: We trained the network from scratch on the video version of the NYU v2 [150] dataset, consisting of hundreds of thousands of video frames. Our deep learning approach produced qualitatively strong results, as can be seen in Fig. 8.4. In particular, by fusing the multiple forms of evidence, the approach was able to produce more detailed interpretations of the scene compared to past work.

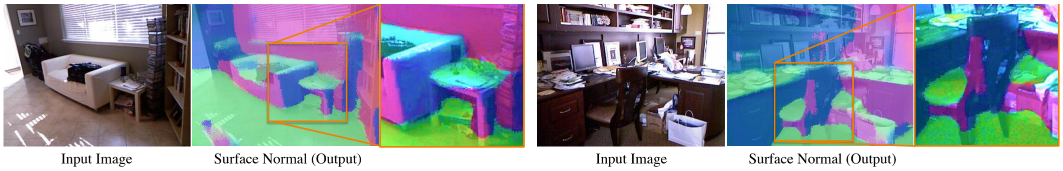


Figure 8.4: Some results from the method

More importantly though, when we performed ablative analysis, we found that our intermediate representations in terms of edges, room layouts, and vanishing points led to a substantial improvement in the most strict metric used (% pixels $< 11.25°$). This suggests that knowledge of physical structure is *complementary* to data-driven techniques, not in competition.

Chapter 9

Future Directions

Much work remains to be done in terms of seeing the full 3D world from images. In this final chapter, we take a step-back from the work and discuss future directions towards this goal.

9.1 Further factoring scenes

The introduction of this thesis argued that scenes were the product of a viewpoint-specific 3D structure with texture placed on top of it. There are a number of factors that have to be similarly separated out in order to obtain a really deep understanding. We see two that are of particular importance.

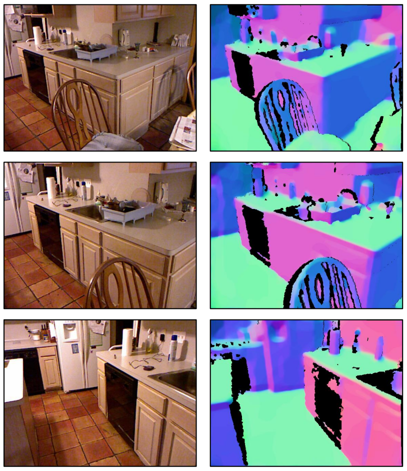


Figure 9.1: Fixing the 3D scene, changing the viewpoint.

Viewpoint-free 3D: The first is that what we call 3D structure in this thesis is actually the product of two factors itself: an underlying viewpoint-free 3D scene and a viewpoint. Together, the 3D scene and viewpoint produce the 3D structure we see. This 3D scene is truly 3D as opposed to a 2.5D representation where each pixel has some property (e.g., depth or normal). When we look at this 3D scene at a particular viewpoint, the fully 3D relations are compressed down into a 2.5D map. For instance, as the camera moves in the scene in Fig. 9.1, both the texture and the normals change. This thesis has implicitly used the fact that the two change in a consistent way to recover the normals with respect to the camera axes from a single image. However, the underlying scene does not change: things come into and out of view as the camera moves and the normals are rigidly transformed as the camera's axes change. In particular, the *relationships* between objects remain fixed: even though the normals on the fridge on the left and the cabinets on the right change, the 90° angle between them does not change.

This distinction does not make a large difference for the task of surface normal estimation – after all, the goal of this task is to predict a viewpoint-specific map for the image. However, the ultimate goal of a 3D understanding system ought to be to produce this full 3D scene. We believe that understanding the intrinsic redundancy in the learning data in the three images in Fig. 9.1 will be important going forward. In work [61] led by Rohit Girdhar, we have taken initial steps towards this in for the case of objects, but it is not clear how to extend this to a scene level.

Lighting: This dissertation has mainly ignored light. This is because the primary cues to shape in the scenes and scales we have worked at are based on texture. However, going forward, understanding light will be of increasing importance: a proper understanding of light greatly improves our understanding and is crucial for textureless objects.

As a concrete example of the importance of light, consider distinguishing convex and concave corners. How should we explain the relative success of our approaches to this problem as well as others'? It is often said, that this ought to be *impossible* due to a variety of ambiguities: for instance, the well-known Bas-Relief ambiguity [13]. If we focus only on texture, our answer has been to point to the regularity present in the data: if the edge joining the counter to the cabinets in Fig. 9.1 were concave, the texture combination would be extraordinarily unlikely. Notably, this does not explain untextured room corners. However, if we properly model light, not only should we not be surprised, but we should expect to distinguish convex and concave edges: these ambiguities go away under more complex lighting models [131] and under inter-reflections [21, 48, 124]. Thus, modeling light properly enables us to understand *why* an approach ought to work and gives us hope that it will work on general data.

As a consequence of ignoring light, this dissertation has also largely ignored textureless objects – on which the only cue is from shading via light – due to the relative frequency in the data and the difficulty of getting even the general shape of the scene (e.g., walls) correct at the start of our work. Nonetheless, as our approaches increasingly start getting the large parts of the scene correct, we will start needing to focus on these objects.

9.2 Broader applications of style

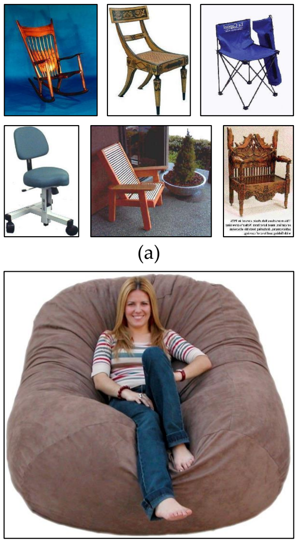
We proposed the idea of the factorization in the context of 3D scene understanding, understanding, but the notion of style, or viewpoint-canonicalized texture, has much broader applications in other scene understanding problems.

The idea of viewpoint-normalization/canonicalization/augmentation has already been used by others in discriminative problems: for instance, in [64, 86, 87] used it for text-recognition and [77, 86, 87] applied it to more general detection and classification. One distinction is that these approaches typically take an augmentation approach [64, 77] in which the model is exposed to all the possible views, or do a learned transformation for normalization [87]. In contrast, our style (as well as [86]'s work) uses geometric knowledge to analytically account for perspective effects. This can typically only be applied in scenes where the geometry is understood, but greatly reduces the search space. We note that while our approach has used vanishing points to generate possible canonical forms, this is not necessarily the case: one can imagine using the output of a viewpoint-specific normal predictor to canonicalize regions up to a much reduced set of transformations.

We believe the problem where style will shine is in generative problems. Already, others have shown that this idea of 3D structure and style can produce more plausible generative models of images [164] by factoring $P(\text{image})$ into $P(\text{image}|\text{normals})$ and $P(\text{normals})$. We believe it can go even further by explicitly leveraging knowledge of geometry: texture synthesis approaches produce substantially more plausible results if perspective effects are removed, then re-applied once the texture is generated [14].

9.3 Reuniting with semantics

The gap between the various forms of "3D understanding" is dwarfed by the gap between 3D and semantics. Future work ought to try to close this gap. The steadily increasing performance of techniques in both areas suggest that the fields may be finally be approaching the point at which it makes sense to start thinking again about their union.


(a)

(b)

Figure 9.2: (a) chairs (b) a bean-bag chair.

From semantics to 3D: It was argued in the background section that 3D succeeds when bottom-up techniques are fused with strong top-down knowledge. So far, we have mainly focused on constraints (Chapter 7), geometric knowledge (Chapters 4, 5), or predictable higher-order properties (Chapter 6). One obvious missing component is what one might call "semantic" objects, such as chairs, sofas, bowls, and cars.

These can act as a powerful top-down cue. For instance, a far-away object may be only a few pixels, but we can resolve its shape in great detail using top-down knowledge. There has been considerable work in CAD model alignment (e.g., [115]), but how to fully integrate this semantic knowledge with bottom-up techniques remains an open question.

From 3D to semantics: Semantic objects are often defined in terms of an underlying higher order 3D property. For instance, chairs permit a person to sit with back support, but not two people. There are an immense number of forms that satisfy this property and naturally, they look very different, for instance the chairs in Fig. 9.2. Now consider generalizing from the chairs in Fig. 9.2(a) to the chairs in Fig. 9.2(b). The six chairs in Fig. 9.2(a) have little in common with each other and even less in common with the bean-bag chair in Fig. 9.2(b). Thus, based on appearance alone, we have little hope. However, if we accept a definition of chair based on 3D, we at least have a chance.

9.4 Reuniting the 3Ds

There has historically been a divide between two types of 3D understanding: one based on intrinsically multi-view cues (e.g., stereo, structure-from-motion), the other based on intrinsically single-view cues (e.g., shape-from-X, 3D structure prediction, CAD alignment). These are typically further subdivided into much more specific sub-cases. However, in recent years, as discussed earlier in Section 3.2, there have been a number of approaches for integrating monocular cues, including the ones presented in this thesis, into traditionally multi-view systems.

We believe that the time is right for the reunion of the various 3D techniques because the full challenge of understanding a scene requires using all the available cues and, *equally importantly*, understanding when each cue ought to be used. The approaches we present in this thesis probably do not generalize to shape-from-shading problems (although recent work does treat this problem with discriminative learning [134]); and it seems unlikely that the problem of understanding the underlying structure of a full street scene is best done via shape-from-shading approaches. Each of these scenarios is working with enough accuracy that we can start to dreaming of tackling the full problem itself.

Chapter 10

Conclusion

This dissertation presented the idea of factoring scenes into 3D structure and style. At the heart of the dissertation is the idea that we should think of monocular 3D understanding as at the intersection of learning-based recognition and physical modeling. We cannot ignore the overwhelming amount of data-driven regularity present in images due to style, but we also cannot simply treat 3D understanding as yet another pixel-labeling problem. Instead, we must both think about the physical aspects of the problem as well as embrace the data driven regularity and biases in human scenes.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, F. P., and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2281, 2012.

[2] A. Ames. *The Ames Demonstrations in Perception*. Hafner Publishing, 1952.

[3] R. Arandjelović and A. Zisserman. Efficient image retrieval for 3D structures. In *BMVC*, 2010.

[4] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *ICCV*, 2011.

[5] R. Arandjelović and A. Zisserman. Name that sculpture. In *ACM ICMR*, 2012.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5), 2011.

[7] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[8] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *CVPR*, 2014.

[9] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2D-3D model alignment via surface normal prediction. In *CVPR*, 2016.

[10] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. *ECCV*, 2012.

[11] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015.

[12] H. Barrow and J. Tenenbaum. Recovering Intrinsic Scene Characteristics from Images. In *Computer Vision Systems*, pages 3–26. Academic Press, 1978.

[13] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *IJCV*, 35(1), 1999.

[14] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Opensurfaces: A richly annotated catalog of surface appearance. In *SIGGRAPH*, 2013.

[15] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[16] T. Binford. Visual perception by computer. In *IEEE Conference on Systems and Controls*, 1971.

[17] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.

[18] L. Bo, X. Ren, and D. Fox. Depth Kernel Descriptors for Object Recognition. In *IROS*, 2011.

[19] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *ICCV*, 2009.

[20] R. Brooks, R. Creiner, and T. Binford. The ACRONYM model-based vision system. In *IJCAI*, 1979.

[21] M. K. Chandraker, F. Kahl, and D. J. Kriegman. Reflections on the generalized bas-relief ambiguity. In *CVPR*, 2005.

[22] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[23] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2009.

[24] M. Clowes. On seeing things. *Artificial Intelligence*, 2:79–116, 1971.

[25] A. Concha, W. Hussain, L. Montano, and J. Civera. Incorporating scene priors to dense monocular mapping. *Autonomous Robots*, 39(3):279–292, 2015.

[26] J. Coughlan and A. Yuille. The Manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *NIPS*, 2000.

[27] A. Criminisi and A. Zisserman. Shape from texture: homogeneity revisited. In *BMVC*, 2010.

[28] J. E. Cutting and P. M. Vishton. Perceiving layout and knowing distances: The interaction, relative potency, and contextual use of different information about depth. In W. Epstein and S. Rogers, editors, *Perception of space and motion*, pages 69–117, 1995.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[30] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. L. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *CVPR*, 2012.

[31] E. Delage, H. Lee, and A. Y. Ng. A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image. In *CVPR*, 2006.

[32] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Efros, and A. Gupta. Scene semantics from long-term observation of people. In *ECCV*, 2012.

[33] S. Dickinson and I. Biederman. Geons. In *Encyclopedia of Computer Vision*. 2013.

[34] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *Neural Information Processing Systems (NIPS)*, 2013.

[35] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013.

[36] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *SIGGRAPH*, 31(4), 2012.

[37] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *CoRR*, abs/1411.4734, 2014.

[38] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[39] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014.

[40] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *CVPR*, 2007.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[42] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[43] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9), 2010.

[44] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.

[45] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.

[46] S. Fidler, S. Dickinson, and R. Urtasun. 3D object detection and viewpoint estimation with a deformable 3D cuboid model. In *NIPS*, 2012.

[47] D. Forsyth. Shape from texture without boundaries. In *ECCV*, 2002.

[48] D. Forsyth and A. Zisserman. Reflections on shading. *TPAMI*, 13(7), 1991.

[49] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *ECCV*, 2012.

[50] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. *IJCV*, 110(3):259–274, 2014.

[51] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3D primitives for single image understanding. In *ICCV*, 2013.

[52] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *ECCV*, 2014.

[53] D. F. Fouhey, A. Gupta, and A. Zisserman. 3D shape attributes. In *CVPR*, 2016.

[54] D. F. Fouhey, W. Hussain, A. Gupta, and M. Hebert. Single image 3D without a single 3D image. In *ICCV*, 2015.

[55] D. F. Fouhey, X. Wang, and A. Gupta. In defense of the direct perception of affordances. *CoRR*, abs/1505.01085, 2015.

[56] W. Freeman and P. Viola. Bayesian model of surface perception. In *NIPS*, 1998.

[57] S. Galliani and K. Schindler. Just look at the image: viewpoint-specific surface normal prediction for improved multi-view reconstruction. In *CVPR*, 2016.

[58] W. L. Gehringer and E. Engel. Effect of ecological viewing conditions on the ames' distorted room illusion. *Journal of Experimental Psychology: Human Perception and Performance*, 12(2):181–185, 1986.

[59] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[60] J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.

[61] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016.

[62] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *ICCV*, 2013.

[63] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[64] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016.

[65] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013.

[66] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014.

[67] A. Guzman. Decomposition of a visual scene into three-dimensional bodies. In *Proceedings Fall Joint Computer Conference*, volume 33, pages 291–304, 1968.

[68] S. Hadfield and R. Bowden. Exploiting high level scene cues in stereo reconstruction. In *ICCV*, 2015.

[69] C. Häne, L. Ladicky, and M. Pollefeys. Direction matters: Depth estimation with a surface normal classifier. In *CVPR*, 2016.

[70] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[71] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[72] T. Hassner and R. Basri. Example based 3D reconstruction from single 2D images. In *CVPR Workshop: Beyond Patches*, 2006.

[73] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[74] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009.

[75] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010.

[76] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2014.

[77] M. Hejrati and D. Ramanan. Categorizing cubes: Revisiting pose normalization. In *WACV*, 2016.

[78] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80, 2008.

[79] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. In *IJCV*, 2007.

[80] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. In *SIGGRAPH*, 2005.

[81] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[82] D. Hoiem, A. Stein, A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007.

[83] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *IJCV*, 91(3):328–346, 2011.

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

[85] D. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 8:475–492, 1971.

[86] W. Hussain, J. Civera, L. Montano, and M. Hebert. Dealing with small data and training blind spots in the manhattan world. In *WACV*, 2016.

[87] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks. In *NIPS*, 2015.

[88] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-D object dataset: Putting the Kinect to work. In *Workshop on Consumer Depth Cameras in Computer Vision (with ICCV)*, 2011.

[89] Z. Jia, A. Gallagher, Y.-J. Chang, and T. Chen. A learning based framework for depth ordering. In *CVPR*, 2012.

[90] Z. Jia, A. Gallagher, A. Saxena, and T. Chen. 3D-based reasoning with blocks, support, and stability. In *CVPR*, 2013.

[91] H. Jiang and J. Xiao. A linear approach to matching cuboids in RGBD images. In *CVPR*, 2013.

[92] A. Johnston and P. J. Passmore. Independent encoding of surface orientation and surface curvature. *Vision Research*, 34(22):3005 – 3012, 1994.

[93] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013.

[94] T. Kanade. A theory of origami world. *Artificial Intelligence*, 13(3), 1980.

[95] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015.

[96] K. Karsch, K. Sunkavalli, S. Hadap, N. Carr, H. Jin, R. Fonte, M. Sittig, and D. Forsyth. Automatic scene inference for 3d object compositing. *ACM Trans. Graph.*, 33(3), 2014.

[97] K. Karsch, Z. Liao, J. Rock, J. T. Barron, and D. Hoiem. Boundary cues for 3D object shape recovery. In *CVPR*, 2013.

[98] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *ECCV*, 2012.

[99] J. J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.

[100] J. J. Koenderink. *Solid Shape*. MIT Press, 1990.

[101] J. J. Koenderink. Pictorial relief. *Philosophical Transactions of the Royal Society of London*, (356), 1998.

[102] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *IVC*, 10(8):557 – 564, 1992.

[103] J. J. Koenderink and A. J. Van Doorn. Relief: Pictorial and otherwise. *Image and Vision Computing*, 13(5):321–334, 1995.

[104] J. J. Koenderink, A. J. Van Doorn, and A. M. Kappers. Surface perception in pictures. *Perception & Psychophysics*, 52(5):487–496, 1992.

[105] J. J. Koenderink, A. J. Van Doorn, and A. M. Kappers. Pictorial surface attitude and local depth comparisons. *Perception & Psychophysics*, 58(2), 1996.

[106] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[107] N. Kumar, A. Berg, P. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[108] L. Ladický, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014.

[109] L. Ladický, B. Zeisl, and M. Pollefeys. Discriminatively trained dense surface normal estimation. In *ECCV*, 2014.

[110] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009.

[111] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010.

[112] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009.

[113] Y. Lee, A. Efros, and M. Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *ICCV*, 2013.

[114] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

[115] J. J. Lim, A. Khosla, and A. Torralba. Fpm: Fine pose parts-based model with 3d cad models. In *ECCV*, 2014.

[116] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014.

[117] A. Lobay and D. Forsyth. Shape from texture without boundaries. *IJCV*, 67, 2006.

[118] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[119] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. Learning a confidence measure for optical flow. *TPAMI*, 35(5), 2013.

[120] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *IJCV*, 23, 1997.

[121] J. Malik. Interpreting line drawings of curved objects. *IJCV*, 1(1), 1987.

[122] D. Marr. *Vision*. Freeman, 1982.

[123] J. L. Mundy. Object Recognition in the Geometric Era: A Retrospective. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 3–28. Springer Berlin Heidelberg, 2006.

[124] S. Nayar, K. Ikeuchi, and T. Kanade. Shape from interreflections. *IJCV*, 6(3), 1991.

[125] M. Nitzberg and D. Mumford. The 2.1D sketch. In *ICCV*, 1990.

[126] J. F. Norman and J. T. Todd. The discriminability of local surface structure. *Perception*, 25:381–398, 1996.

[127] A. Owens, J. Xiao, A. Torralba, and W. T. Freeman. Shape anchors for data-driven multi-view reconstruction. In *ICCV*, 2013.

[128] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[129] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[130] J. Ponce and M. Hebert. On image contours of projective shapes. In *ECCV*, 2014.

[131] E. Prados and O. Faugeras. Shape from shading: a well-posed problem? In *CVPR*, 2005.

[132] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *CVIR*, 2008.

[133] S. Ramalingam, J. Pillai, A. Jain, and Y. Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *CVPR*, 2013.

[134] S. R. Richter and S. Roth. Discriminative shape from shading in uncalibrated illumination. In *CVPR*, 2015.

[135] L. Roberts. Machine perception of 3D solids. In *PhD Thesis*, 1963.

[136] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3D object shape from one depth image. In *CVPR*, 2015.

[137] C. Rother. A new approach to vanishing point detection in architectural environments. *IVC*, 20, 2002.

[138] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015.

[139] S. Satkin, M. Rashid, J. Lin, and M. Hebert. 3DNN: 3D nearest neighbor data-driven geometric scene understanding using 3D models. *IJCV*, 111, 2015.

[140] S. Satkin, J. Lin, and M. Hebert. Data-driven scene understanding from 3D models. In *BMVC*, 2012.

[141] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[142] A. Saxena, J. Schulte, and A. Ng. Depth estimation using monocular and stereo cues. In *IJCAI*, 2007.

[143] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *TPAMI*, 30(5):824–840, 2008.

[144] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.

[145] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[146] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2004.

[147] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box In the Box: Joint 3D Layout and Object Reasoning from Single Images. In *ICCV*, 2013.

[148] A. G. Schwing and R. Urtasun. Efficient Exact Inference for 3D Indoor Scene Understanding. In *ECCV*, 2012.

[149] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171:701–703, 1971.

[150] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012.

[151] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[152] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[153] F. Srajer, A. G. Schwing, M. Pollefeys, and T. Pajdla. MatchBox: Indoor Image Matching via Box-like Scene Estimation. In *3DV*, 2014.

[154] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, 1986.

[155] M. Tappen, E. Adelson, and W. Freeman. Estimating intrinsic component images using nonlinear regression. In *CVPR*, 2006.

[156] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. In *NIPS*, 2003.

[157] M. J. Tarr and H. H. Bülthoff. Image-based object recognition in man, monkey and machine. *Cognition*, 67(1-2):1–20, 1998.

[158] J. Tenenbaum and W. Freeman. Separating style and content. In *NIPS*, 1997.

[159] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[160] D. Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*. McGraw-Hill, 1975.

[161] H. Wang, Y. Wexler, E. Ofek, and H. Hoppe. Factoring repeated content within and among images. *SIGGRAPH*, 27(3), 2008.

[162] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015.

[163] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[164] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016.

[165] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D model matching with viewpoint-invariant patches (VIP). In *CVPR*, 2008.

[166] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, 2012.

[167] J. Xiao, B. Russell, and A. Torralba. Localizing 3D cuboids in single-view images. In *NIPS*, 2012.

[168] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. In *ECCV*, 2012.

[169] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *CVPR*, 2011.

[170] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, 2007.

[171] S. X. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *Workshop on Perceptual Organization*, 2008.

[172] A. Zaheer, M. Rashid, and S. Khan. Shape from angular regularity. In *ECCV*, 2012.

[173] J. Zhang, K. Chen, A. G. Schwing, and R. Urtasun. Estimaing the 3D Layout of Indoor Scenes and its Clutter from Depth Sensors. In *ICCV*, 2013.

[174] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. Tilt: Transform-invariant low-rank textures. *IJCV*, 99, 2014.

[175] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. *NIPS*, 2014.

[176] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, 2015.

[177] A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *IVC*, 7(1):38–42, 1989.