UNIVERSITY OF CALIFORNIA

Los Angeles

A Quest for Visual Commonsense: Scene Understanding by Functional and Physical Reasoning

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Yibiao Zhao

2015

Abstract of the Dissertation

A Quest for Visual Commonsense: Scene Understanding by Functional and Physical Reasoning

by

Yibiao Zhao

Doctor of Philosophy in Statistics

University of California, Los Angeles, 2015

Professor Song-Chun Zhu, Co-Chair

Professor Ying Nian Wu, Co-Chair

Computer vision has made significant progress in locating and recognizing objects in recent decades. However, beyond the scope of this what is where challenge, it lacks the abilities to understand scenes characterizing human visual experience. Comparing with human vision, what is missing in current computer vision? One answer is that human vision is not only for pattern recognition, but also supports a rich set of commonsense reasoning about object function, scene physics, social intentions *etc*.

I build systems for real world applications and simultaneously pursuing a long-term goal of devising a unified framework that can make sense of an images and a scene by reasoning about the functional and physical mechanisms of objects in a 3D world. By bridging advances spanning fields of stochastic learning, computer vision, cognitive science, my research tackles following challenges:

(i) What is the visual representation? I develop stochastic grammar models to characterize spatiotemporal structures of visual scenes and events. The analogy of human natural language lays a foundation for representing both visual structure and abstract knowledge. I pose the scene understanding problem as parsing an image into a hierarchical structure of visual entities using the Stochastic Scene Grammar (SSG). With a set of production rules, the grammar enforces both structural regularity and flexibility of visual entities. Therefore, the algorithm is able to handle

enormous number of configurations and large geometric variations for both indoor scenes and outdoor scenes.

(ii) How to reason about the commonsense knowledge? I augment the commonsense knowledge about functionality, physical stability to the grammatical representation. The bottom-up and top-down inference algorithms are designed for finding a most plausible interpretation of visual stimuli.

Functionality refers to the property of an object or scene, especially man-made ones, which has a practical use for which it was designed, and it's deeper than geometry and appearance and thus is a more invariant concept for scene understanding. We present a Stochastic Scene Grammar (SSG) as a hierarchical compositional representation which integrates functionality, geometry and appearance in a hierarchy. This represents a different philosophy that views vision tasks from the perspective of agents, that is, agents (humans, animals and robots) should perceive objects and scenes by reasoning their plausible functions.

Physical stability assumption assumes objects in the static scene should be stable with respect to the gravity field. In other words, if any object is not stable on its own, it must be either grouped with neighbors or fixed to its supporting base. We pursue a physically stable scene understanding, namely "a parse tree", by inferring object stability in the physical world. The assumption is applicable to general scene categories thus poses powerful constraints for physically plausible scene interpretation and understanding.

(iii) How to acquire commonsense knowledge? I performed three case studies to acquire different kinds of commonsense knowledges: I teach the computer to learn affordance from observing human actions; to learn tool-use from single one-shot demonstration; and to infer containing relations by physical simulation without explicit training process. They provided some interesting perspectives on how to acquire and exploit commonsense knowledge. In general, the more prediction or simulation is performed, the less training data is needed. As a result, the acquired commonsense knowledge is more generalizable to new situations.

Such sophisticated understanding of 3D scenes enables computer vision to reason, predict, interact with the 3D environment, as well as hold intelligent dialogues beyond visible spectrum.

The dissertation of Yibiao Zhao is approved.

Keith Holyoak

Hongjing Lu

Demetri Terzopoulos

Ying Nian Wu, Committee Co-Chair

Song-Chun Zhu, Committee Co-Chair

University of California, Los Angeles

2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost I want to thank my advisor Dr. Song-Chun Zhu for his guidance and mentorship during my PhD study at UCLA. His passion and insight for research, expertise, and perspective has been an inspiration, and has helped me improve as a researcher, professional, and as a person. I am deeply grateful to Dr. Ying Nian Wu, co-chair of my committee, who gave helpful advice on research and career and taught me how to talk about vision seriously. I would also like to thank my committee Dr. Demetri Terzopoulous, Dr. Hongjing Lu, as well as Dr. Keith Holyoak for their insightful advice and discussions along the way. My appreciation also goes to Dr. Josh Tenenbaum and Dr. Katsushi Ikeuchi for their inspired discussions and encouraging support during our collaborations.

Many thanks also goes out to my many colleagues in the VCLA lab for their friendship, collaboration, and support throughout my time here. In particular, I thank Yixin Zhu, Siyuan Qi, Xiaobai Liu, Wei Liang, Ping Wei, Steven Holtzen for working closely with me during the past years, all the credit of my work also belongs to them. I also I extend my thanks to Tianfu Wu, Benjamin Yao, Brandon Rothrock, Zhangzhang Si, Mingtian Zhao, Wenze Hu, Zhi Han, Mingtao Pei, Jungseock Joo, Seyoung Park, Shuo Wang, Amy Fire, Maria Pavlovskaia, Jifeng Dai, Yang Lu, Joyce Meng, Bruce Nie, Kewei Tu, Dan Xie, Joey Yu, Bo Li, Weixin Li, Huanquan Lu, Yang Liu, Hang Qi, Tianmin Shu, Nishant Shukla, Nawin Waree, Caiming Xiong, Yuanlu Xu, Sam Freitas and Quanshi Zhang.

I am fortunate to make good friends and collaborate with Bo Zheng, Lap-Fai Yu, Tao Gao and Peter Battaglia. Our collaborations lead to an important research direction of ”vision meets cognition”, and a few interdisciplinary workshops we co-organized at CVPR and CogSci were wonderful.

Lastly, I cannot thank my parents and Xiaofei enough for their unwavering support.

Portions of this work were supported by DARPA MSEE project FA 8650-11-1- 7149, MURI ONR N00014-10-1-0933, and NSF IIS 1423305.

VITA

2011 - 2015	Ph.D. Candidate (Statistics), UCLA, Los Angeles, CA.
2015 Summer	CBMM Summer School: Brains, Minds and Machines, Woods Hole, MA.
2012 Summer	Visiting student, The Computational Cognitive Science Group, MIT, MA.
2011 Summer	IPAM Summer School: Probabilistic Models of Cognition, UCLA, CA.
2010 - 2011	Visiting student, UCLA, Los Angeles, CA.
2008 - 2009	Visiting student, Lotus Hill Institute, Hubei, China.
2003 - 2007	B.S. (Electronic commerce), Beijing Jiaotong University, Beijing, China.

PUBLICATIONS

[1] **Yibiao Zhao**, Song-Chun Zhu, *”Integrating Function, Geometry, Appearance for Scene Parsing”*, International Journal of Computer Vision, IJCV. (under revision)

[2] Bo Zheng, **Yibiao Zhao**, Katsushi Ikeuchi, Song-Chun Zhu, *”Scene Understanding by Reasoning Stability and Safety”*, International Journal of Computer Vision, IJCV. S.I. Scene Understanding, 2015

[3] Xiaobai Liu, **Yibiao Zhao**, Song-Chun Zhu, *”Attributed Grammar for Single-View 3D Scene Parsing”*, IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) (under revision)

[4] Ping Wei, **Yibiao Zhao**, Nanning Zheng, Song-Chun Zhu, *”Events and Objects in 4D Human-Object Interaction Space”*, IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) (under revision)

[5] Yixin Zhu, **Yibiao Zhao** (equal contribution), Song-Chun Zhu, *”Understanding Tool Use: a*

Task-oriented Vision Problem”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[6] Wei Liang, **Yibiao Zhao**, Yixin Zhu, Song-Chun Zhu, *”Evaluating Human Cognition of Containing Relations with Physical Simulation”*, Annual Conference of Cognitive Science Society (CogSci) 2015

[7] Bo Zheng, **Yibiao Zhao** (equal contribution), Katsushi Ikeuchi, Song-Chun Zhu, *”Detecting Potential Falling Objects by Inferring Human Action and Natural Disturbance”*, IEEE International Conference on Robotics and Automation (ICRA) 2014

[8] Jiajun Wu, **Yibiao Zhao** (equal contribution), Jun-Yan Zhu, Siwei Luo, Zhuowen Tu, *”MILCut: A Sweeping Line Multiple Instance Learning Paradigm for Interactive Image Segmentation”*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

[9] Xiaobai Liu, **Yibiao Zhao**, Song-Chun Zhu, *”Single-View 3D Scene Parsing by Attributed Grammar”*, IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2014

[10] **Yibiao Zhao**, Song-Chun Zhu, *”Scene Parsing by Integrating Function, Geometry and Appearance Models”*, IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2013

[11] Bo Zheng, **Yibiao Zhao**, Joey C. Yu, Katsushi Ikeuchi, Song-Chun Zhu, *”Beyond Point Clouds: Scene Understanding by Reasoning Geometry and Physics”*, IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2013

[12] Ping Wei, **Yibiao Zhao**, Nanning Zheng, Song-Chun Zhu, *”Modeling 4D Human-Object Interactions for Event and Object Recognition”*, IEEE International Conference on Computer Vision, (ICCV) 2013

[13] Ping Wei, Nanning Zheng, **Yibiao Zhao**, Song-Chun Zhu, *”Concurrent Action Detection with Structural Prediction”*, IEEE International Conference on Computer Vision, (ICCV) 2013

[14] **Yibiao Zhao**, Song-Chun Zhu, *”Image Parsing via Stochastic Scene Grammar”*, Advances on Neural Information Processing Systems, (NIPS) 2011

CHAPTER 1

Introduction

1.1 Motivation

We have entered the Big Data Era – the acquisition and sharing of vast quantities of data has never been easier. On Facebook alone, 350 million photos are uploaded every day. Computer vision is about to play a key role in peoples daily lives by creating systems that can see like humans to infer general principles and ongoing situations from imagery.

Over the past 20 years, researchers in computer vision and learning have paid a great deal of attention to appearance-based methods and have achieved remarkable progress in recognizing objects, actions, and scenes. However, computers still cannot perform many important tasks that are trivial for human vision. For example, in the ImageNet Challenge 2014, the average precision of the state-of-the-art object detection algorithm over 200 classes is 43.9%, and the rates are generally worse for functional objects, such as 23.4% for hammer, 26.8% for sofa, and 28.9% for table. Both the geometric approaches in the 1980-1990s and the appearance methods dominating the last 15-years have fundamental limits. This performance gap between people and computers seemingly cannot be filled by designing better features or better classifiers as people have been doing in recent years.

Image understanding is not only about the image itself but also the commonsense knowledge of the world. Humans recognize images so well because we know how the world works. In order to grant computers the same reasoning power, it is important to understand the learning and reasoning mechanism of vision and cognition in computational terms. What is the underlying representation that captures the complex visual structure of the world and also supports reliable generalization to novel situations? What is the core knowledge for understanding an image, and how can we build more powerful learning machines based on functional, physical, causal and social mechanisms of the world?

1.2 Overview of the Methodology

My research aims to close the gap between human vision and computer vision by exploring statistical methods for representing 3D visual world and exploiting commonsense knowledge about functionality, physics, intentionality and causality (FPIC) in ways that can make human-level performance possible in machines.

Firstly, I develop a stochastic grammar model to characterize spatial and temporal structures of visual scenes and events. I believe the inner representation for computer vision and robotics system has to be compositional and flexible like human natural languages for explaining the world and expressing thoughts.

Secondly, I augment the grammatical representation with commonsense knowledge. The high-level knowledge projects onto an image (or a sentence) thus forms contexts of the vision (or a language). My collaborators and I coined functionality and physics as principal determinants in our research to model how a visual scene is organized by human or by nature. And we also consider intentionality and causality as fundamental bases that drive human actions as well as temporal events. Such observation is ubiquitous in vision:

- Objects, especially man-made, are defined by their functions and actions that they are involved.
- Actions, especially rational ones, are intended to change the world and are defined by causal-effects.
- Scenes, especially man-made, are defined by activities and actions that they can provide space for.

Thirdly, I design algorithms for Bayesian inference in an analysis-by-synthesis fashion. A forward synthesis model constructs possible interpretations of the world, and then selects the one that best agrees with the visual evidence as well as high-level prior knowledge. For example, if someone asks, ”Can an iPhone crack nuts?” we may build a mental picture of a scene with a person smashing nuts with an iPhone. This mental picture allows us to reason, predict, answer questions, and hold intelligent dialogs beyond visible spectrum.

In effect, these components span a large space of image understanding. Computer vision systems can go beyond labeling what is where on an image to building a sophisticated understanding of a scenes three-dimensional organization over time. These abilities allow a computer to answer an almost limitless range of questions about an image using a finite and general-purpose model, which will finally be instrumental to designing computers that can understand images like humans and answer queries, such as:

”What is Obama doing? Why is he doing that?”

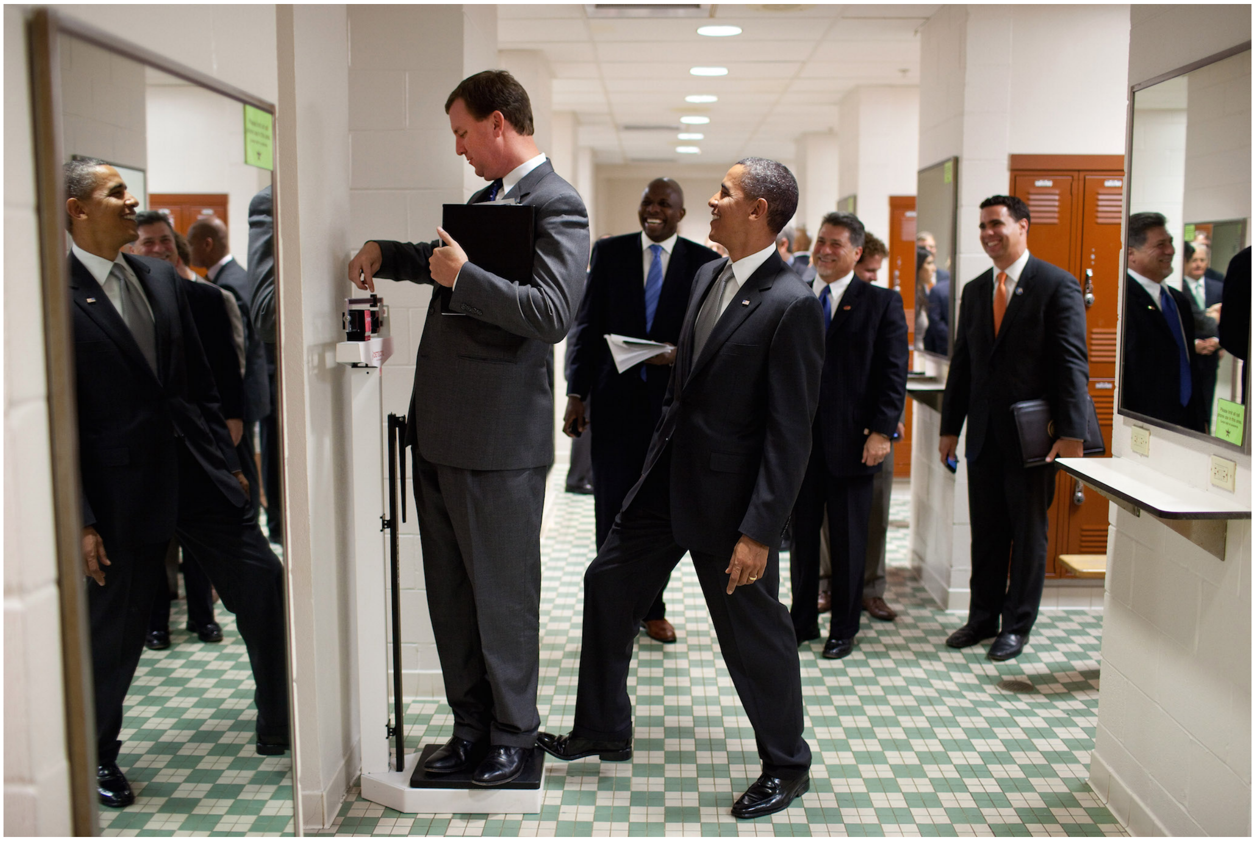


Figure 1.1: Scene understanding with the commonsense. (Image courtesy of Andrej Karpathy)

To answer this question, a computer not only has to know traditional what is where questions, but also needs to understand a huge amount of commonsense knowledge beyond image pixels:

- What is a scale used for?

- How does a scale work?

- What happens after Obama puts his toe on the scale?

- Where is the persons attention?

- What are Obamas intents and beliefs?

- What do other people predict the persons reaction will be as he reads off his over-weighted measurement?

1.3 Related work

Our method is related to five streams of research in the literature which we will briefly discuss in the following.

1.3.1 Scene Representation

There are five major scene representations in the vision literature.

(i) Representing scene as feature vectors for classification, such as the scene gist in [OT01], spatial pyramid matching (SPM) in [LSP06] and recent reconfigurable scene models by [POF12] and [WWZ12].

(ii) Region-based representations for semantic scene labeling. Conditional random fields [LMP01] are widely used to represent semantic relations between adjacent regions, such as {*inside, below, around, above* }. [CLT10] studied 2D context models that guide detectors to produce a semantically coherent interpretation of a scene. They showed that such 2D horizontal contexts are very sensitive to camera rotations.

(iii) Non-parametric representations for scene labeling, for example, label transfer by SIFT flow in [LYT11], SuperParsing in [TL13a, TL13b] and scene collage in [IL13] interpret a new scene by searching nearest neighbors from images in the scene dataset, and then transfer the label maps to the target through warping or contextual inference. Interestingly, [SLH12], [SH13] recently generalize the idea of nearest-neighbor search to the 3D scenes, so that their approach can recognize objects cross viewpoints. [LPT13, LKT14, DBK13] detected indoor objects by matching with fine-grained 3D CAD furniture models. [AME14, SX14] detects chairs by exemplar-SVM classifiers with a large set of synthetic training data, which rendered from 3D CAD models under various viewpoints.

(iv) 3D block world representation, which allows reasoning about the physical constraints within the 3D scene. [GEH10b] posed 3D objects as blocks and inferred their 3D properties such as occlusion, exclusion and stability in addition to surface orientation labels. They showed that a global 3D prior does improve 2D surface labeling. [HHF09, HHF10, HHF12], [WGK10], [LHK09, LGH10], [SU12, SHP12, SFP13] parameterized the geometric scene layout of the background and/or foreground blocks and trained their models by the structured SVM (or latent SVM).

(v) Deformable part-based models: [Hu12], [XRT12], [HR12], [XS12], [PGS12], [FDU12], [DR13] designed several new variants of the deformable part-based models to detect 3D entities under different view points.

1.3.2 3D Reconstruction from Single Image

Automatic 3D reconstruction from a single image was considered an ill-posed problem. In order to recover a meaningful 3D reconstruction, researchers make assumptions about the scene and use prior knowledge to regularize the solution. In this research stream, people used four types of assumptions.

(i) Sketch smoothness assumption:

[HZ04a] was the first tackling this problem by assuming the local sketch smoothness and global scene alignment for recovering 3D objects, like plant, tree and buildings from 2D single image.

(ii) Piece-wise smoothness assumption:

[SSN09] presented a fully supervised method to learn a mapping between informative features and depth values under a conditional random field framework. [PT11] proposed a joint model to recognize objects and estimate scene shape simultaneously.

(iii) Surface assumption:

[HEH09] recognized the geometric surface orientation and fit ground-line that separate the floor and objects in order to pop-up the vertical surface. [DLN07] proposed a dynamic Bayesian network model to infer the floor structure for autonomous 3D reconstruction from a single indoor image. [MZY11] extracted low rank textures of repeated patterns to construct surfaces like building facades. Recently, [FGH14] proposed the use of convex and concave edges for regularizing scene configurations like playing Origami.

(iv) Manhattan world assumption:

Recent studies on indoor scene parsing, including [HHF09, HHF10, HHF12], [WGK10], [LHK09, LGH10], [SHP12, SU12, SFP13], [ZZ11, ZZ13] and [DGB11, DBF12, DBK13] adopted the Manhattan world representation extensively. This assumption stated that man-made scenes are built on a cartesian grid and thus have regularities in the image edge gradient statistics. This enables us, from a single image, to determine the orientation of the viewer relative to the scene and also to recover scene structures which are aligned with the grid.

Most recently, a series of work, including [LFU13, CCP13, ZZ13, GH13, ZST14], proposed holistic approaches to exploits 2D semantic segmentation, 3D geometry, as well as 3D contextual relations in a joint framework.

1.3.3 Stochastic Grammar for Image Understanding

This stream of research started from “syntactic pattern recognition” by K. S. Fu and his school in the late 1970s to early 1980s. [Fu82] depicted an ambitious program of block world scene understanding using grammars. This stream was disrupted in the 1980s and suffered from the lack of an image vocabulary that is realistic enough to express real-world objects and scenes, and reliably detectable from images.

[TCY05] raised the notion of image parsing to the decomposition of an image into a hierarchical “parse graph” by a Data-Driven Markov Chain Monte Carlo sampling strategy. [ZM07] proposed an And-OR graph model to represent the compositional structures in vision. [HZ09] detected rectangular structures in man-made scenes by applying bottom-up / top-down grammar rules in a greedy manner. [PZ10] proposed a cluster sampling algorithm to parse aerial images by allowing for Markov chain jumping between competing solutions. A recent work [LCK14] studied a probabilistic grammar model for labelling 3D CAD scenes.

1.3.4 Object Functionality and Affordance

In computer vision, [SB91] pioneered the use of functional properties in 3D object recognition. They parsed an objects into a 3D geometric description, and recognized the object by searching potential functional elements. Both developmental psychologist [OM08] and computer vision researchers [YMF13] demonstrated that functionality is at least as important as appearance in recognizing objects.

More recently, numerous approaches have been proposed to detect functional objects based on human-object interactions in video. [WZZ13] and [JKS13] extracted human actions from RGBD video data and used the human actions as a prior to indirectly detect objects and label scenes. [BR06] and [GGG11] detected chairs by hallucinating agents in the 3D CAD data and depth data respectively. [GSE11] proposed an algorithm to infer the human workable space by adapting human poses to the scene.

[DFL12] and [FDG12] recovered the semantics and geometry of a scene by observing human activities in the room. [KCG14] learned an affordance model to predict a static pose that a person would need to adopt in order to use an object. [KS13] anticipated human activities using object affordance learned from RGBD videos [KS14].

1.3.5 Physical Reasoning and Intuitive Physics

The vision communities have studied the physical properties based on a single image for the ”block world” in the past three decades [BMR82, GEH10a, GSE11, **?**, LHK09, LGH10]). *e.g.*Biederman *et al.* [BMR82] studied human sensitivity of objects that violate certain physical relations. Our goal of inferring physical relations is most closely related to [GEH10a] who infer volumetric shapes, occlusion, and support relations in outdoor scenes inspired by physical reasoning from a 2D image, and Silberman *et al.* [NF12, JKS13, GH13] who infers the support relations between objects from a single depth image using supervised learning with many prior features. In contrast, our work is the first that defines explicitly the mathematical model for object stability. Without a supervised learning process, our method is able to infer the 3D objects with maximum stability.

The intuitive physics model is an important perspective for human-level complex scene understanding. However, to our best knowledge, there is little work that mathematically defines intuitive physics models for real scene understanding. [JGS13] adopts an intuitive physics model in [McC83], however this model lacks deep consideration on complex physical relations. In our recent work [**?**, ZZY14], we propose a novel intuitive physics model based on gravity potential energy transfer. In this paper, we extend this intuitive physics model by combining specific physical disturbance fields. While Physics engines in graphics can accurately simulate the motion of objects under the influence of gravity, it is computationally too expensive for the purpose of measuring object stability.

Recent psychology studies suggested that approximate Newtonian principles underlie human judgements about dynamics and stability [FBB10, HBT11] Hamrick *et al.* [HBT11, BHT13] showed that knowledge of Newtonian principles and probabilistic representations are generally applied for human physical reasoning. These intuitive models are studied for understanding human behaviors, not for vision robotics.

1.4 Thesis outline

The rest of the chapters of this dissertation are organized as follows:

Chapter 2 presents a stochastic grammar models to characterize 3D structures of visual scenes. The analogy of human natural language lays a foundation for representing both visual structure and abstract knowledge. I pose the scene understanding problem as parsing an image into a hierarchical structure of visual entities using the Stochastic Scene Grammar (SSG). With a set of production rules, the grammar enforces both structural regularity and flexibility of visual entities. Therefore, the algorithm is able to handle enormous number of configurations and large geometric variations for both indoor scenes and outdoor scenes.

Chapter 3 introduces our grammar model for integrating function, geometry and appearance. The motivation behind this is to infer the functionality, *i.e.*the property of an object or scene, especially man-made ones, which has a practical use for which it was designed. This represents a different philosophy that views vision tasks from the perspective of agents, that is, agents (humans, animals and robots) should perceive objects and scenes by reasoning their plausible functions. The bottom-up and top-down inference algorithms are designed for finding a most plausible interpretation of functional assignments and 3D scene layouts.

Chapter 4 describes an algorithm for scene understanding by physical reasoning. We pursue a physically stable scene understanding, namely “a parse tree”, by inferring object stability in the physical world. Physical stability assumption assumes objects in the static scene should be stable with respect to the gravity field. This assumption is applicable to general scene categories thus poses powerful constraints for physically plausible scene interpretation and understanding.

Chapter 5 describes three case studies for learning visual commonsense. I teach the computer to learn affordance from observing human actions; to learn tool-use from single one-shot demonstration; and to infer containing relations by physical simulation without explicit training process. They provided some interesting perspectives on how to acquire and exploit commonsense knowledge. In general, the more prediction or simulation is performed, the less training data is needed. As a result, the acquired commonsense knowledge is more generalizable to new situations.

Chapter 6 concludes the dissertation, and outlines our contributions.

CHAPTER 2

Stochastic Grammar for 3D Scene Understanding

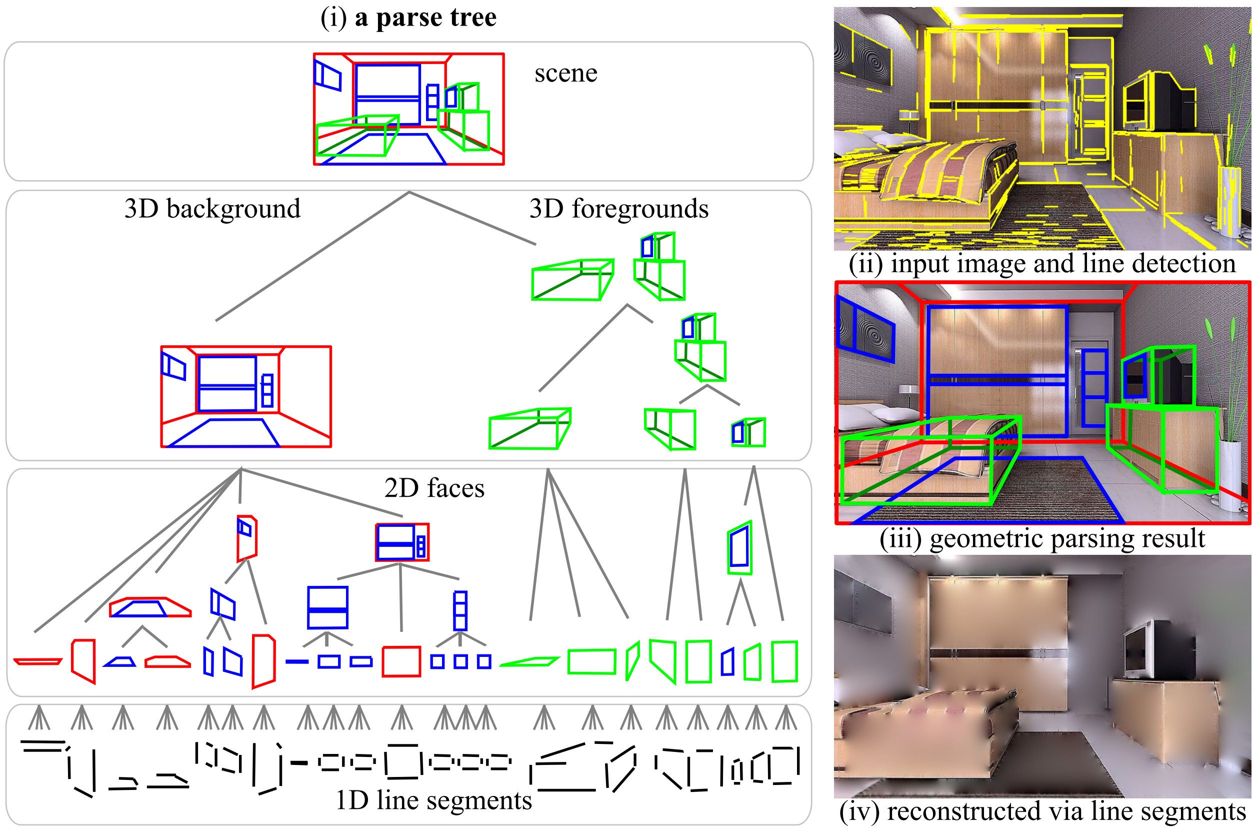

(i) a parse tree
scene
3D background
3D foregrounds
2D faces
1D line segments
(ii) input image and line detection
(iii) geometric parsing result
(iv) reconstructed via line segments

Figure 2.1: A parse tree of geometric parsing result.

Scene understanding is an important task in neural information processing systems. By analogy to natural language parsing, we pose the scene understanding problem as parsing an image into a hierarchical structure of visual entities (in Fig.2.1(i)) using the Stochastic Scene Grammar (SSG). With a set of production rules, the grammar enforces both structural regularity and flexibility of visual entities. Therefore, the algorithm is able to handle enormous number of configurations and large geometric variations, which are the major difficulties of image parsing.

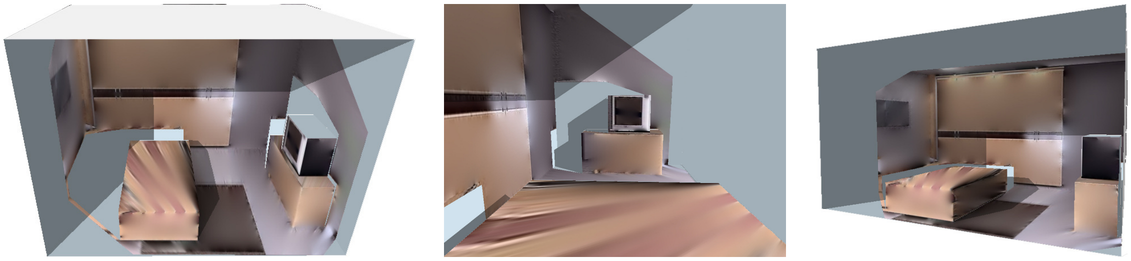


Figure 2.2: 3D synthesis of novel views based on the parse tree.

2.1 Language Grammar

A *Context-Free Grammar* is defined as $G = (S, V, R)$. $V = V^N \cup V^T$, and V^T is a finite set of terminal symbols, V^N is a finite set of non-terminal symbols (structures or sub-structures), $S \in V^N$ is a distinguished non-terminal called the start symbol, and R is a finite set of productions of the form $A \to BC$ or $A \to w$ in *Chomsky Normal Form* with no useless productions, where $A, B, C \in N$ and $w \in T$.

A set of all valid configurations C derived from production rules is called a *language*:

$$L(G) = \{C : S \overset{\{r_i\}}{\to} C, \{r_i\} \subset R, C \subset V^T\}. \tag{2.1}$$

A *Probabilistic Context-Free Grammar* (PCFG) is defined by a pair (G, P) consisting of a context-free grammar G and a real-valued vector P of length $|R|$ indexed by production ruless, where $P = P(\alpha \to \beta)$ is an expansion probability for each production rule $\alpha \to \beta \in R$. It is required that $P(\alpha \to \beta) \geq 0$ and $\sum_{(\alpha \to \beta) \in R} P(\alpha \to \beta) = 1$ for all nonterminals $\alpha \in V^N$. A *parse tree pt* is a set of nodes, each node has a chosen production rule $\alpha \to \beta$.

The probability of a parse tree is derived from the PCFG is defined as

$$P(pt|S) = \prod_{\alpha \in V^N} P(\alpha \to \beta) \tag{2.2}$$

2.2 Attributed Grammar and Context-Sensitive Grammar

The *Stochastic Scene Grammar* in this paper is designed for modeling 3D scene structures for parsing a 2D image. Different from traditional language parsing problem, the 3D scene parsing faces two major challenges: 3D geometry and context sensitivity. Therefore, we modified the traditional grammar model in two aspects accordingly:

I) Geometry: The complexity of 3D scene parsing problem comes from the explicit modeling of 3D geometric arrangement of objects, while the language grammar only need to handle the left-right order of words. We augment the nodes in the grammar with 3D geometric attributes, and thus extend it to attributed grammar. We represent each 3D object by a 3D cuboid with three geometric attributes: size (3 DoF), relative position (3 DoF) and relative orientation (1 DoF around gravity axis).

II) Context sensitivity: There are two kinds of contexts: physical exclusion and graphical occlusion. The physical exclusion means each grammar node (such as an object) should be physically collision-free with all the other objects in the 3D scene, and the graphical occlusion means that all the grammar nodes compete with each other for explaining the image pixels with respect to the depth order in an image formation process. Thus the grammar becomes Context-sensitive which breaks the probabilistic derivation in Eq. 2.2 in the way that the image data not only depends on its direct parents but also be constrained by all the other notes in the image formation process. In particular, we explicitly model the image formation process in the inference stage by an analysis-by-synthesis paradigm.

Therefore, the SSG is attributed and context sensitive, for which traditional inference algorithm, such as inside-outside algorithm, are no longer applicable. Inspired by probabilistic models of cognition [TKG11, BHT13, MKP13, GT14], we design the two context-sensitive modules in a probabilistic program. At each MCMC iteration, a probabilistic sample is evaluated by recreating the image formatting process, we reconstruct a volumetric 3D scene and re-render a 2D image with a depth buffer.

2.3 Stochastic Grammar for Indoor Scene Parsing

The Stochastic Scene Grammar (SSG) is defined as a four-tuple $G = (S, V, R, P)$, where S is a start symbol at the root (scene); $V = V^N \cup V^T$, V^N is a finite set of non-terminal nodes (structures or sub-structures), V^T is a finite set of terminal nodes (line segments); $R = \{r : \alpha \to \beta\}$ is a set of production rules, each of which represents a generating process from a parent node α to its child nodes $\beta = Ch_\alpha$. $P(r) = P(\beta|\alpha)$ is an expansion probability for each production rule $(r : \alpha \rightarrow \beta)$. A set of all valid configurations C derived from production rules is called a *language*: $L(G) = \{C : S \xrightarrow{\{r_i\}} C, \{r_i\} \subset R, C \subset V^T, P(\{r_i\}) > 0\}$.

Production rules. We define three types of stochastic production rules R^{AND},R^{OR},R^{SET} to represent the structural *regularity* and *flexibility* of visual entities. The regularity is enforced by the AND rule and the flexibility is expressed by the OR rule. The SET rule is a mixture of OR and AND rules.

(i) An AND rule $(r^{AND} : A \rightarrow a \cdot b \cdot c)$ represents the *decomposition* of a parent node A into three sub-parts a, b, and c. The probability $P(a, b, c|A)$ measures the compatibility (contextual relations) among sub-structures a, b, c. As seen Fig.2.11(i), the grammar outputs a high probability if the three faces of a 3D box are well hinged, and a low probability if the foreground box lays out of the background.

(ii) An OR rule $(r^{OR} : A \rightarrow a \mid b)$ represents the *switching* between two sub-types a and b of a parent node A. The probability $P(a|A)$ indicates the preference for one subtype over others. For 3D foreground in Fig.2.11(iii), the three sub-types in the third row represent objects below the horizon. These objects appear with high probabilities. Similarly, for the 3D background in Fig.2.11(iii), the camera rarely faces the ceiling or the ground, hence, the three sub-types in the middle row have higher probabilities (the higher the darker). Moreover, OR rules also model the discrete size of entities, which is useful to rule out the extreme large or small entities.

(iii) An SET rule $(r^{SET} : A \rightarrow \{a\}_k, k \geq 0)$ represents an *ensemble* of k visual entities. The SET rule is equivalent to a mixture of OR and AND rules $(r^{SET} : A \rightarrow \emptyset \mid a \mid a \cdot a \mid a \cdot a \cdot a \mid \cdots)$. It first chooses a set size k by ORing, and forms an ensemble of k entities by ANDing. It is worth noting that the OR rule essentially changes the graph topology of the output parse tree by changing its node size k. In this way, as seen in Fig.2.11(ii), the SET rule generates a set of 3D/2D entities which satisfy some contextual relations.

Contextual relations. There are two kinds of contextual relations, *Cooperative* "+" relations and *Competitive* "-" relations, which involve in the AND and SET rules.

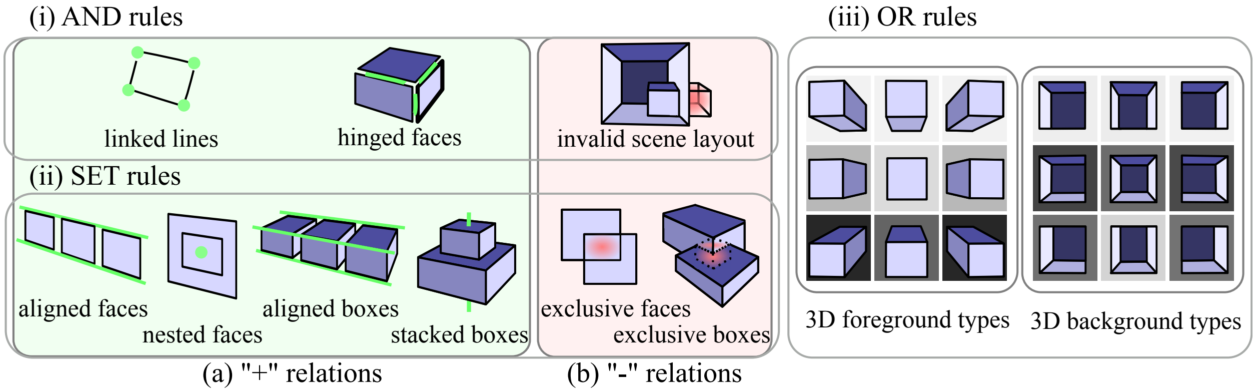


Figure 2.3: Three types of production rules: AND (i), SET (ii) OR (iii), and two types of contextual relations: cooperative "+" relations (a), competitive "-" relations (b).

(i) The cooperative "+" relations specify the *concurrent* patterns in a scene, e.g. hinged faces, nested rectangle, aligned windows in Fig.2.11(a). The visual entities satisfying a cooperative "+" relation tend to bind together.

(i) The competitive "-" relations specify the *exclusive* patterns in a scene. If entities satisfy competitive "-" relations, they compete with each other for the presence. As shown in Fig.2.11(b), if a 3D box is not contained by its background, or two 2D/3D objects are exclusive with one another, these cases will rarely be in a solution simultaneously.

The *tight structures* vs. the *loose structure*: If several visual entities satisfy a cooperative "+" relation, they tend to bind together, and we call them *tight structures*. These tight structures are grouped into clusters in the early stage of inference (Sect.3.3). If the entities neither satisfy any cooperative "+" relations nor violate a competitive "-" relation, they may be loosely combined. We call them *loose structures*, whose combinations are sampled in a later stage of inference (Sect.3.3). With the three production rules and two contextual relations, SSG is able to handle an enormous number of configurations and large geometric variations, which are the major difficulties in our task.

2.3.1 Bayesian Formulation of the Grammar

We define a posterior distribution for a solution (a parse tree) pt conditioned on an input image I. This distribution is specified in terms of the statistics defined over the derivation of production rules.

$$P(pt|I) \propto P(pt)P(I|pt) = P(S) \prod_{v \in V^N} P(Ch_v|v) \prod_{v \in V^T} P(I|v) \tag{2.3}$$

where I is the input image, pt is the parse tree. The probability derivation represents a generating process of the production rules $\{r : v \to Ch_v\}$ from the start symbol S to the nonterminal nodes $v \in V^N$, and to the children of non-terminal nodes Ch_v. The generating process stops at the terminal nodes $v \in V^T$ and generates the image I.

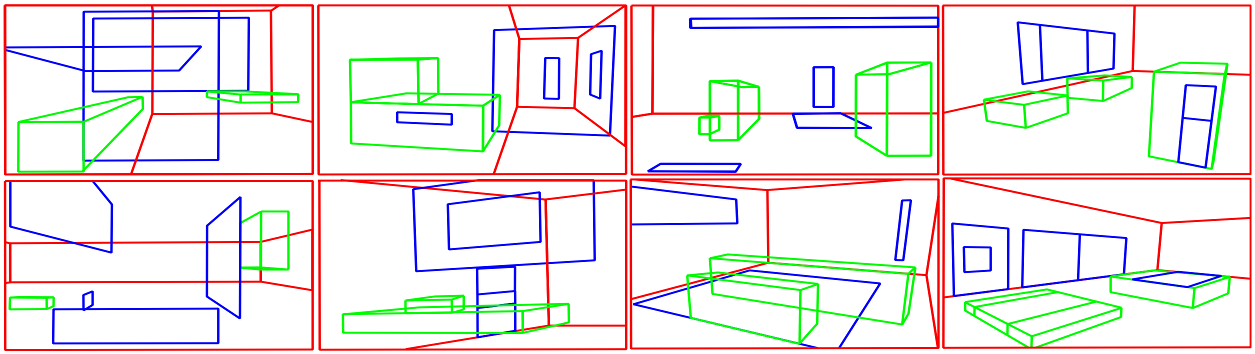
We use a probabilistic graphical model of AND/OR graph to formulate our grammar. The graph structure $G = (V, E)$ consists of a set of nodes V and a set of edges E. The edge define a parent-child conditional dependency for each production rule. The posterior distribution of a parse graph pt is given by a family of Gibbs distributions: $P(pt|I; \lambda) = 1/Z(I; \lambda) \exp\{-E(pt|I)\}$, where $Z(I; \lambda) = \sum_{pt \in \Omega} \exp\{-E(pt|I)\}$ is a partition function summation over the solution space Ω.

The energy is decomposed into three potential terms:

$$E(pt|I) = \sum_{v \in V^{OR}} E^{OR}(A_T(Ch_v)) + \sum_{v \in V^{AND}} E^{AND}(A_G(Ch_v)) + \sum_{\Lambda_v \in \Lambda_I, v \in V^T} E^T(I(\Lambda_v)) \tag{2.4}$$

(i) **The energy for OR nodes** is defined over ”type” attributes $A_T(Ch_v)$ of ORing child nodes. The potential captures the prior statistics on each switching branch. $E^{OR}(A_T(v)) = -\log P(v \to A_T(v)) = -\log\{\frac{\#(v \to A_T(v))}{\sum_{u \in Ch(v)} \#(v \to u)}\}$. The switching probability of foreground objects and the background layout is shown in Fig.2.11(iii).

(ii) **The energy for AND nodes** is defined over ”geometry” attribute $A_G(Ch_v)$ of ANDing child nodes. They are Markov Random Fields (MRFs) inside a tree-structure. We define both “+” relations and “-” relations as $E^{AND} = \lambda^+ h^+(A_G(Ch_v)) + \lambda^- h^-(A_G(Ch_v))$, where $h(*)$ are



(i) initial distribution (ii) with cooperative(+) relations (iii) with competitive(-) relations (iv) with both (+/-) relations

Figure 2.4: Learning to synthesize. (a)-(d) Some typical samples drawn from Stochastic Scene Grammar model with/without contextual relations.

sufficient statistics in the exponential model, λ are their parameters. For 2D faces as an example, the “+” relation specifies a quadratic distance between their connected joints $h^{+}(A_G(Ch_v)) = \sum_{a,b \in Ch_v}(X(a) - X(b))^2$, and the “-” relation specifies an overlap rate between their occupied image area $h^{-}(A_G(Ch_v)) = (\Lambda_a \cap \Lambda_b)/(\Lambda_a \cup \Lambda_b)$, $a, b \in Ch_v$.

(iii) **The energy for Terminal nodes** is defined over bottom-up image features $I(\Lambda_v)$ on the image area Λ_v. The features used in this paper include: (a) surface labels of geometric context [HHF09], (b) a 3D orientation map [LHK09], (c) the MDL coding length of line segments [VJM10]. This term only captures the features from their dominant image area Λ_v, and avoids the double counting of the shared edges and the occluded areas.

We learn the context-sensitive grammar model of SSG from a context-free grammar. Under the learning framework of minimax entropy [ZWM98], we enforce the contextual relations by adding statistical constraints sequentially. The learning process matches the statistics between the current distribution p and a targeted distribution f by adding the most violated constraint in each iteration. Fig.2.4 shows the typical samples drawn from the learned SSG model. With more contextual relations being added, the sampled configurations become more similar to a real scene, and the statistics of the learned distribution become closer to that of target distribution.

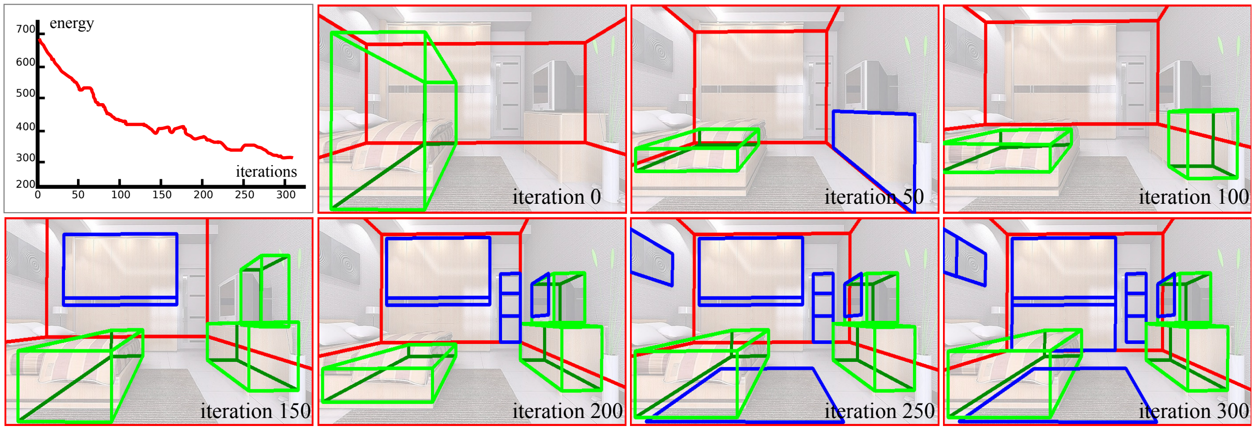


Figure 2.5: The hierarchical cluster sampling process.

2.3.2 Inference with Hierarchical Cluster Sampling

We design a hierarchical cluster sampling algorithm to infer the optimal parse tree for the SSG model. A parse tree specifies a configuration of visual entities. The combination of configurations makes the solution space expand exponentially, and it is NP-hard to enumerate all parse trees in such a large space.

In order to detecting scene components, neither sliding window (top-down) nor binding (bottom-up) approaches can handle the large geometric variations and an enormous number of configurations. In this paper we combine the bottom-up and top-down process by exploring the contextual relations defined on the grammar model. The algorithm first perform a bottom-up clustering stage and follow by a top-down sampling stage.

In the clustering stage, we group visual entities into clusters (tight structures) by filtering the entities based on cooperative “+” relations. With the low-level line segments as illustrated in Fig.3.9.(iv), we detect substructures, such as 2D faces, aligned and nested 2D faces, 3D boxes, aligned and stacked 3D boxes (in Fig.2.11(a)) layer by layer. The clusters Cl are formed only if the cooperative “+” constraints are satisfied. The proposal probability for each cluster Cl is defined as

$$P_+(Cl|I) = \prod_{v \in Cl^{OR}} P^{OR}(A_T(v)) \prod_{u,v \in Cl^{AND}} P_+^{AND}(A_G(u), A_G(v)) \prod_{v \in Cl^T} P^T(I(\Lambda_v)). \tag{2.5}$$

Clusters with marginal probabilities below threshold are pruned. The threshold is learned by a probably approximately admissible (PAA) bound [23]. The clusters so defined are enumerable.

In the sampling stage, we performs an efficient MCMC inference to search in the combinational space. In each step, the Markov chain jumps over a cluster (a big set of nodes) given information of ”what goes together” from clustering. The algorithm proposes a new parse tree: $pt^* = pt + Cl^*$ with the cluster Cl^* conditioning on the current parse tree pt. To avoid heavy computation, the proposal probability is defined as

$$Q(pt^*|pt, I) = P_+(Cl^*|I) \prod_{u \in Cl^{AND}, v \in pt^{AND}} P_-^{AND}(A_G(u)|A_G(v)). \tag{2.6}$$

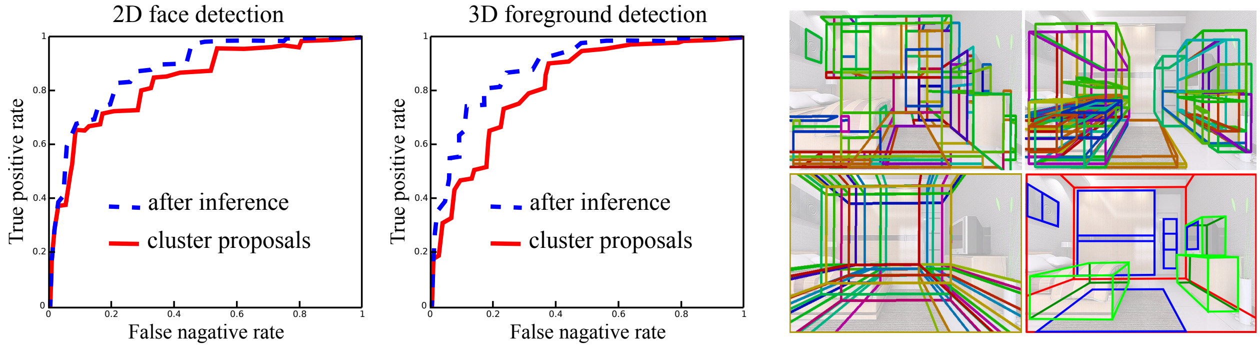
The algorithm gives more weights to the proposals with strong bottom-up support and tight “+” relations by $P_+(Cl|I)$, and simultaneously avoids the exclusive proposals with “-” relations by $P_-^{AND}(A_G(u)|A_G(v))$. All of these probabilities are pre-computed before sampling. The marginal probability of each cluster $P_+(Cl|I)$ is computed during the clustering stage, and the probability for each pair-wise negative “-” relations $P_-^{AND}(A_G(u)|A_G(v))$ is then calculated and stored in a look-up table. The algorithm also proposes a new parse tree by pruning current parse tree randomly.

By applying the Metropolis-Hastings acceptance probability $\alpha(pt \to pt*) = min\{1, \frac{Q(pt|pt*, I)}{Q(pt*|pt, I)} \cdot \frac{P(pt*|I)}{P(pt|I)}\}$, the Markov chain search satisfies the detailed balance principle, which implies that the Markov chain search will converge to the global optimum in Fig.2.5.

2.3.3 Experiments

We evaluate our algorithm on both the UIUC indoor dataset [HHF09] and our own dataset. The UIUC dataset contains 314 cluttered indoor images, of which the ground-truth is two label maps of background layout with/without foreground objects. Our dataset contains 220 images which cover six indoor scene categories: bedroom, living room, kitchen, classroom, office room, and corridor. The ground-truths are hand labeled segments for scene components for each image. Our algorithm usually takes 20s in clustering, 40s in sampling, and 1m in preparing input features.

Qualitative evaluation: The experimental results in Fig.2.7 is obtained by applying different production rules to images in our dataset. With the AND rules only, the algorithm obtains rea-



(a) (b) (c)

Figure 2.6: Quantitative performance of 2D face detection (a) and 3D foreground detection (b) in our dataset. (c) An example of the top proposals and the result after inference.

sonable results and successfully recovers some salient 3D foreground objects and 2D faces. With both the AND and SET rules, the cooperative "+" relations help detect some weak visual entities. Fig.2.8 lists more experimental results of the UIUC dataset. The proposed algorithm recovers most of the indoor components. In the last row, we show some challenging images with missing detections and false positives. Weak line information, ambiguous overlapping objects, salient patterns and clustered structures would confuse our algorithm.

Quantitative evaluation: We first evaluate the detection of 2D faces, 3D foreground objects in our dataset. The detection error is measured on the pixel level, it indicates how many pixels are correctly labelled. In Fig.2.6, the red curves show the ROC of 2D faces / 3D objects detection in clustering stage. They are computed by thresholding cluster probabilities given by Eq.2.5. The blue curves show the ROC of final detection given a partial parse tree after MCMC inference. They are computed by thresholding the marginal probability given Eq.2.4. Using the UIUC dataset, we compare our algorithm to four other state-of-the-art indoor scene parsing algorithms, Hoiem et al. [HEH07], Hedau et al. [HHF09], Wang et al. [WGK10] and Lee et al. [LHK09]. All of these four algorithms used discriminative learning of Structure-SVM (or Latent-SVM). By applying the production rules and the contextual relations, our generative grammar model outperforms others as shown in Table.2.1.

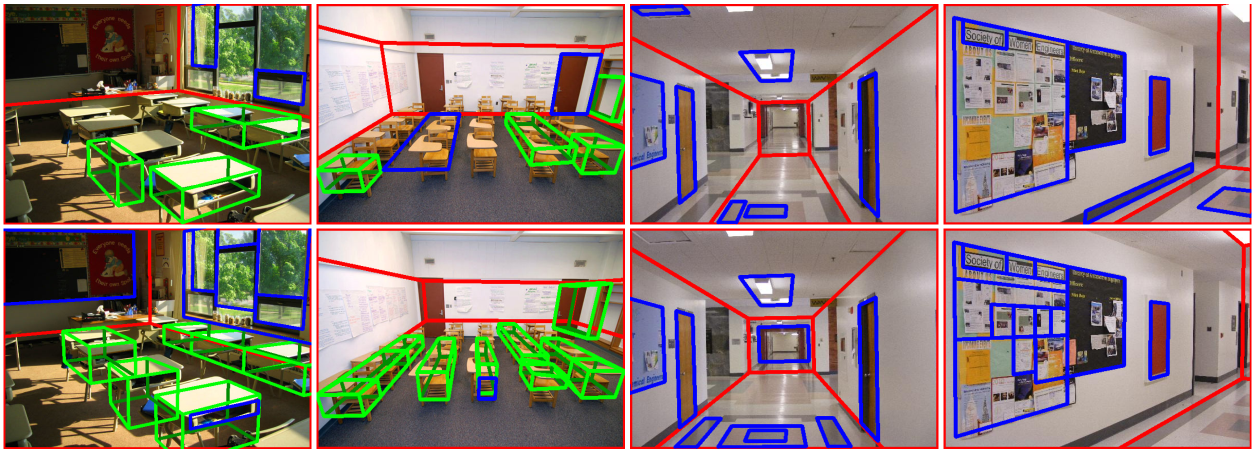


Figure 2.7: Experimental results by applying the AND/OR rules (the first row) and applying all AND/OR/SET rules (the second row) in our dataset

Table 2.1: Segmentation precision compared with Hoiem et al. 2007 [HEH07], Hedau et al. 2009 [HHF09], Wang et al. 2010 [WGK10] and Lee et al. 2010 [LHK09] in the UIUC dataset [HHF09].

Segmentation precision	[HEH07]	[HHF09]	[WGK10]	[LHK09]	Our method
Without rules	73.5%	78.8%	79.9%	81.4%	80.5%
With 3D “-” constraints	-	-	-	83.8%	84.4%
With AND, OR rules	-	-	-	-	85.1%
With AND, OR, SET rules	-	-	-	-	85.5%

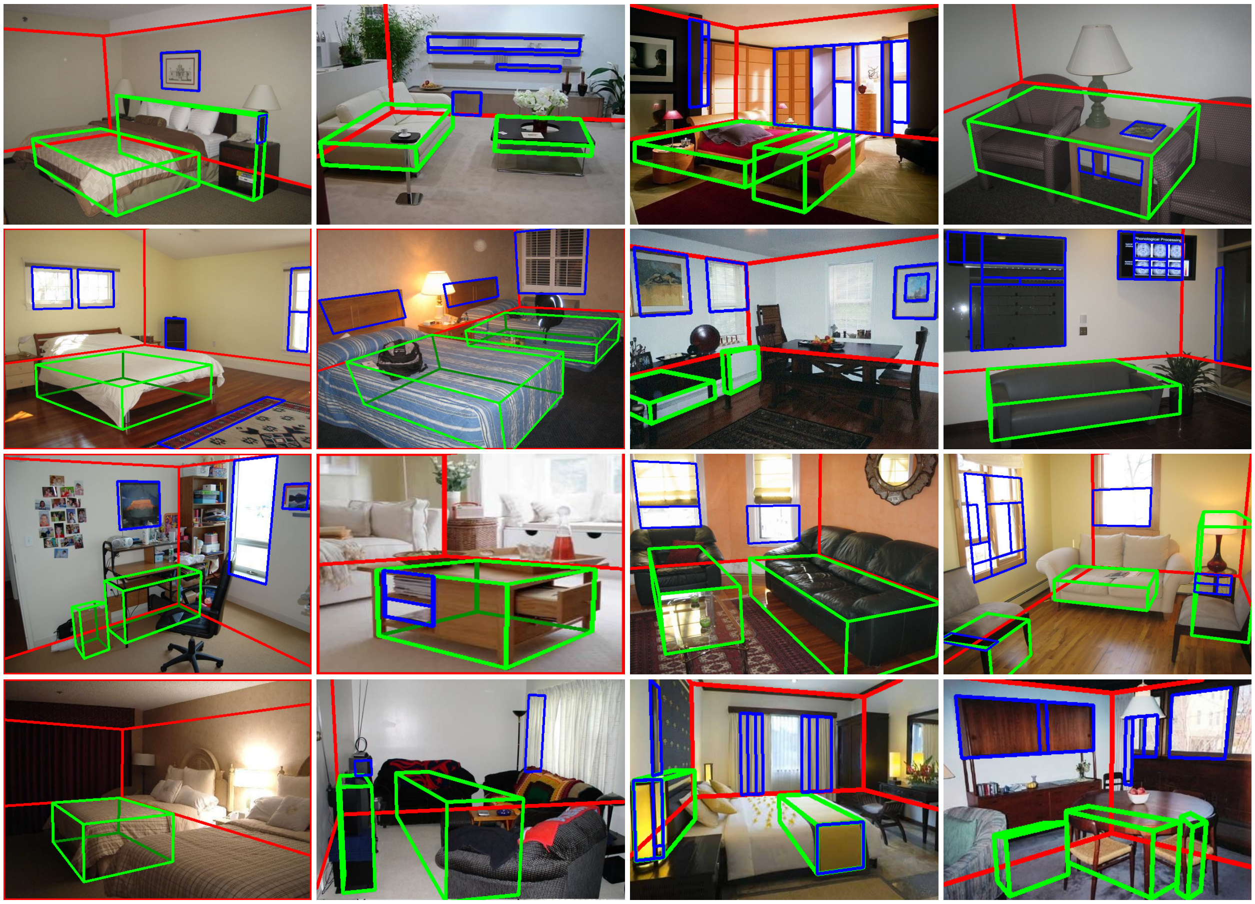


Figure 2.8: Experimental results of more complex indoor images in UIUC dataset [HHF09]. The last row shows some challenging images with missing detections and false positives of proposed algorithm.

2.4 Stochastic Grammar for Outdoor Scene Parsing

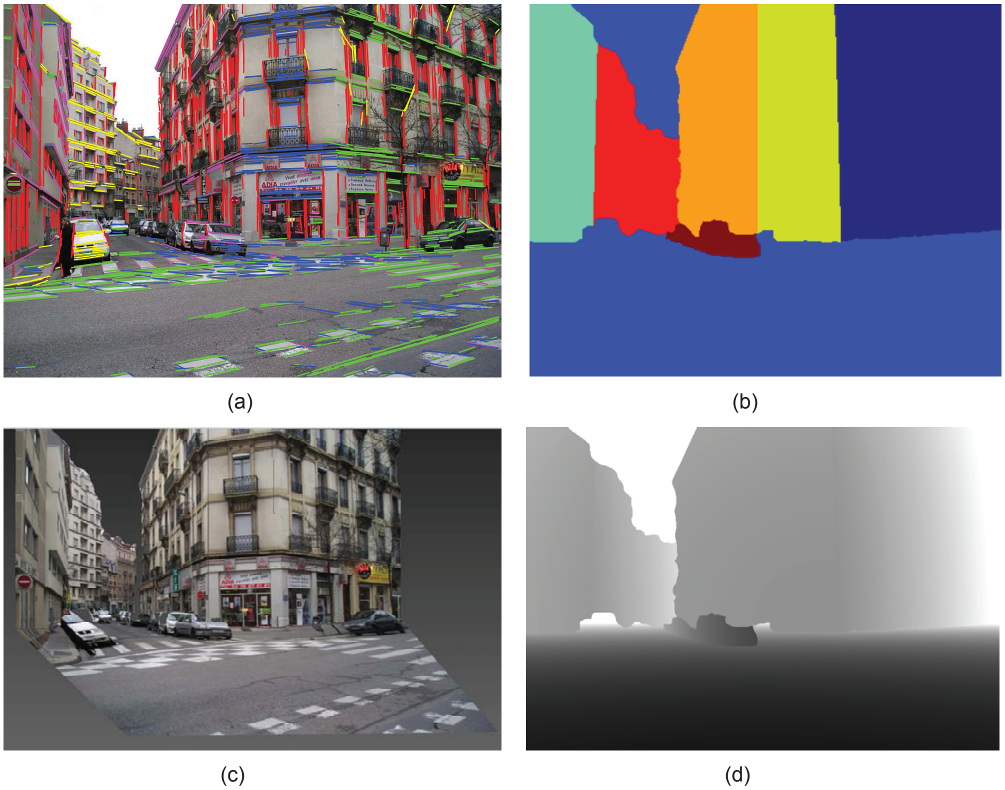


Figure 2.9: A typical result of our approach. (a) input image overlaid detected parallel lines; (b) segmentation of scene layout; (c) synthesized image from a novel viewpoint; d) recovered depth map (darker pixels means closer).

Automatically creating high-quality 3D model from single view is valueless to image understanding and other high-level vision tasks, e.g. human activities recognition. This is a challenging problem due to its ill-posed nature. However, for the image of man-made outdoor scenes, human can recognize 3D structure of the scene effortlessly. We conjecture that human make 3D inference with some commonsense knowledge, such as most of the objects placed on the ground due to gravity, building are most standing uprightly, man-made scenes usually have Manhattan type structure [CY03], or parallel lines in the words merge at vanishing points in images. Recently, researchers also tried to use the physics law to guide the 3D reconstruction [GEH10a]. Integrating these cues can definitely improve the system performance whereas an open problem is how to select the most useful knowledge during the inference.

In this paper, we present a simple attributed grammar for single view3D scene parsing for man-made scenes. The basic observation is, like language where a large number of sentences are generated by a small set of words through a few of grammar rules, the visual patterns in the scene can be decomposed hierarchically into primitives through a few grammar rules. The grammar uses the superpixels as terminal nodes.

Given one image, our goal is to build a hierarchical parse graph where each nonterminal node corresponds to a production rule. Our grammar uses attributes as switch variables to introduce constraints on nodes. Figure 2.11(a) illustrates a hierarchical representation of an outdoor scene. In this parse tree, the vertical links show the decomposition of the scene or one node into their components, and the horizontal links specify the spatial relationship between components. Both vertical and horizontal relationships are regularized by the local attributes of the nodes and the global attributes of the whole scene. The global attributes include the camera parameters, e.g. focal length, and multiple Cartesian coordinate systems (CCS). Each CCS includes three or two orthogonal families of parallel lines. In contrast with the Manhattan world which partitions all the parallel lines into three orthogonal families or one single CCS, we allow one scene have multiple CCS and further assume all surfaces in one scene should belong to one of this CCS. Two CCSs may share at most one parallel family. These attributes are associated with the root node and will be inherited by all the nodes in the hierarchical parse graph. Every node, however, has its own constrain equations, which may use parts of these attributes.

We formulate the problem of constructing parse graph as maximizing a posterior probability and develop an efficient cluster sampling algorithm for inference. This algorithm is able to exploit various grammar rules to make proposals either by bottom-up detections or top-down predictions.

Local Manhattan World

The image of man-made scenes is full of families of parallel lines, e.g. the boundary of a road, the edges of facades, etc. These families of parallel lines are often, but not necessarily, perpendicular to each other. This observation goes beyond the Manhattan assumption [CY03] [LHK09] which assume all parallel lines in a scene shall form one Cartesian coordinate system (CCS). In fact, one scene usually includes more than 3 parallel families and thus there would be multiple CCSs each including two or three orthogonal families of parallel lines. We call

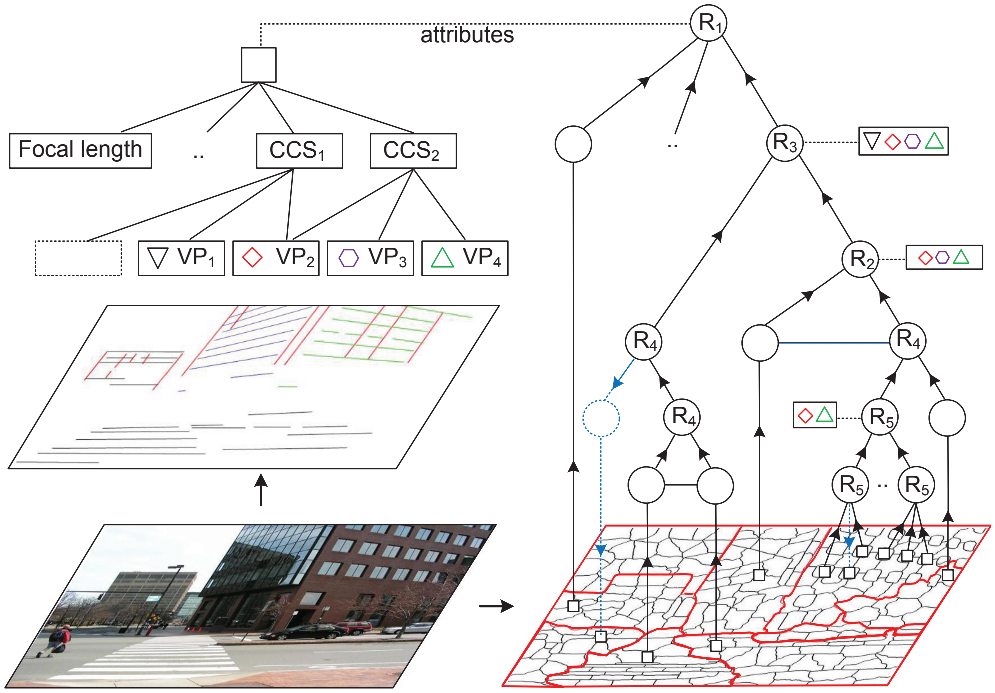


Figure 2.10: Parsing images using grammar rules.

this kind of scenes as *Local Manhattan World*, because only local regions, instead of the whole scene, follow the Manhattan world assumption. We further assume that i) all the CCSs in the same scene share the same vertical axis, i.e. the vertical vanishing point (VP), and ii) every surface in the scene belongs to one and only one CCS. Thus, we could use VP to indicate the surface orientation.

We adopt a simple procedure to discover the CCSs for the input image. Firstly, we use the method by Tretyak et al in [EBK12] to detect the families of parallel lines and their associated vanishing points (VPs). This method also identifies one of the VPs as the vertical VP. All other VPs are considered as horizontal VPs. Second, based on the orthogonality between the vertical VP and horizontal VPs in the world space, we can estimate the camera focal length using the technique in [CDR99]. Last, we adopt the proof by contradiction strategy to check if two horizontal VPs are orthogonal (see experiment for details).

Attribute Image Grammar We first introduce the mathematic definition of attribute grammar used in our work. It is first proposed by Han et al. in [HZ09] for image parsing and we extend it to integrate 3D spatial relationships between nodes. A attributed grammar is specified by a 5-tuple: $G = (V_N, V_T, S, R, P)$, where V_N is the set of non-terminal nodes, V_T is the set of terminal nodes, S is the initial root node for the whole scene, R is a set of production rules for spatial relationships, and P is the probability for the grammar. A non-terminal node is denoted by capital letter, $A_1, A_2 \in V_N$, and a terminal node is denoted by a lowercase letter , $a, b, c \in V_T$. Both non-terminal and terminal nodes have one vector of attributes $X(A)$ or $X(a)$ respectively.

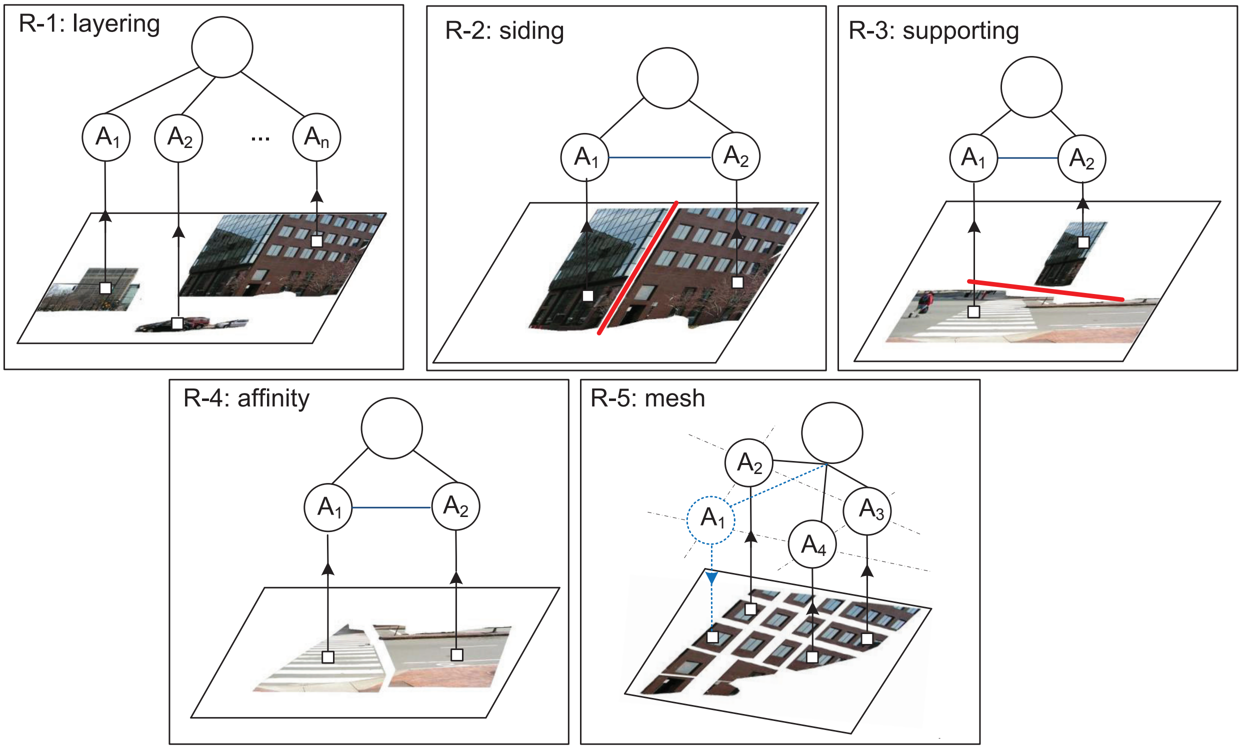


Figure 2.11: Illustration of the proposed five grammar rules.

We partition the input image into a set of superpixels and use them as the terminal nodes of the proposed attributed grammar, denoted as $V_T = \{(a, X(a) : X(a) \in \Omega_a\}$ The attributes of a terminal node is defined as: $X(a) = (u, v, h)$ where (u, v) is the central location im image, h is visual features extracted from this local region. We use the method by Ren et al. [RM03] to partition the image pixels into superpixels so each corresponds with only one geometric surface. There are about 200-300 suerpixels generated for each image.

Production Rules The parse graph consists of one root node S for the whole scene, and essentially it is a graph-structured representation expanded from S by a sequence of production rules. As aforementioned, there are five rules in our generative grammar including: R_1, the layering rule; R_2 the siding rule, R_3, the supporting rule; R_4, the appearance rule; and R_5, the mesh rule. Every non-terminal node in the parse graph can be decomposed into children nodes or grouped with other nodes to form parent nodes by applying the above grammar rules. We denote all the non-terminal nodes as $V_N = \{(S, X(S)), (A, X(A)) : X(S), X(A) \in \Omega_A\}$ which includes the root node S for the whole scene, A denotes the non-terminal node and $X(A)$ the attributes of node A.

The **layering rule** $R_1 : S \to (A_1, A_2, ...)$ generates the scene node S into m independent objects. Herein, one object indicates either an superpixels or a non-terminal nodes generated by other rules. The attributes of S include the focal length f of the camera, the camera height h (i.e. the distance from the camera center to the ground), and n Local Manhattan World (LMW). Each LMW includes two or three VPs that are orthogonal to each other in the 3D world. The attribute of this node is defined as: $X(S) = (f, h, n, \{LMW_1, LMW_2, ..\})$ The layering rule is a loose grammar which does not generate any constraints equations.

The **siding rule** $R_2 : A \to (A_1, A_2)$ states two surfaces stand side by side in the 3D world. They usually belong to the same object (e.g. building). Since we assume all objects or surfaces in the scene stand on the ground, in the image, most of the siding surfaces are separated by a line passing through the vertical VP. Therefore, the attributes of A include: $X(A) = (u, v, \vec{l})$ where $\vec{l}$ is the parameters of the contact line between A_1 and A_2. The constraints for this rule include: 1) 1) A_1 and A_2 are spatially connected in image; 2) the normal direction of A_1 and A_2 are orthogonal to the vertical VP and 3) the contact line $\vec{l}$ should go through the vertical VP in image. Notice that this rule does not require the normals of these children surfaces to be orthogonal, in contrast with the work in [GEH10a].

The **supporting rule** $R_3 : A \to (A_1, A_2)$ states the node A_1 is supporting A_2. The attributes of A is same as that of R_2. The constraints for R_3: 1) the contact line $\vec{l}$ in the image should go though the horizontal VP to which the normal of A_2 is orthogonal; 2) the normal of A_1 should be parallel to the vertical VP in 3D world.

The **affinity rule** $R_4 : A \to (A_1, A_2)$ states that two nodes have similar appearance and thus are likely to belong to the same surfaces. The attributes of A is defined as $X(A) = (u, v, h, \theta)$ where θ indicates the VP which is orthogonal to the normal of A. The default valus of θ is unknown. This rule requires that: 1) A_1 and A_2 are spatially connected in image; 2)A_1 and A_2 have the same surface normal except the normal of A_1 or A_2 is labeled as unknown;

The **mesh rule** $R_5 : A \to (A_1, A_2, A_3, ...)$ states that multiple nodes are arranged in a mesh structure. One mesh can be described by two orthogonal VPs, denoted as θ_1, θ_2. Thus, the attributes of A include: $X(A) = (u, v, \theta_1, \theta_2)$. The constraints equations for this rule include: 1) any children node should be spatially connected to at least one of other children nodes; 2) children nodes should take one of the VPs as their attribute θ.

Among of the above rules, R_2,R_4 and R_5 can be applied recursively while the constraints equations are satisfied. For example, for three nodes A_1, A_2, A_3, we could apply R_3 to A_2 and A_3 to obtain node A_4, and further apply the same rule to A_1 and A_4 to obtain another node. Fig. 2.11(a) illustrates these five rules and Fig. 2.11(b) shows one parse graph that generates the input image. Overall, the rules R_1, R_2 and R_3 describe the 3D spatial relationship between nodes while the rules R_4 and R_5 describe the 2D spatial relationships between nodes.

This simple grammar can generate a large number of parse graphs for generic scenes. Every graph determines one layout segmententation by clustering the superpixels together according to the nodes of R_4, R_5 in the constructed parse graph. Fig. 2.11(b) shows one example of one layout segmententation in the bottom line. In addition, for two surfaces applied by the siding or the supporting rule, the union image regions of these two surfaces will be partitioned by the contact line into two parts each corresponding to one of the two surfaces. This projection from the parse graph to the configuration would help reducing the errors produced by the pre-step of superpixel partition. To obtain the optimal parse graph, we need to enforce constraints over the attributes of the parent nodes and those of the children nodes. In the next section, we introduce a Bayesian treatment for this problem to maximize a posterior probability.

Bayesian Formulation Given an input image, our goal is to solve its optimal parse graph ans its associated layout segmentation in the 2D image. This hierarchical parse graph along with its geometric attributes are able to derive a full 3D model for the input image. Let G denote the parse graph to solve, C the planar configuration $C = C(G)$ produced by G. We can formulate the above target in a Bayesian framework to maximize a posterior probability:

$$G^* = \arg\max p(I|C)p(G)p(C) \tag{2.7}$$

We shall discuss the prior model $p(G)$ and $p(C)$ and the likelihood model $p(I|C)$ in the rest of this section.

Prior Model

The probability $p(C)$ is used to encourage the typical Ising/Potts prior used in the grouping problem [BZ05], i.e., two neighbor sites tend to be grouped together. Let $\mathbf{c}(a)$ denote the region index or the color of a terminal node a in the layout segmentation, we define $p(C)$ as,

$$p(C) = \frac{1}{Z} exp\{\beta \sum_{<a,b>} \mathbf{1}(\mathbf{c}(a) = \mathbf{c}(b))\} \tag{2.8}$$

where $\mathbf{1}(\mathbf{c}(a) = \mathbf{c}(b)) = 1$ if $\mathbf{c}(a) = \mathbf{c}(b)$ for two adjacent superpixels otherwise it is zero. The highest probability is achieved when all vertices are the same color, or being merged into one single region. β and Z are constants both of which can be determined from the training data.

The other prior probability $p(G)$ is defined over the non-terminal nodes of the parse graph G. Let $\ell(A)$ denote the grammar rule associated with the node A, $X(A)$ the attributes of A, and $ch(A)$ the children nodes of A. The probability $p(G)$ can be factorized as,

$$p(G) = \prod_{A \in V_N} p(\ell(A))p(ch(A)|X(A), \ell(A)) \tag{2.9}$$

where $\ell(A)$ is a switch variable for selecting one of the grammar rules. The probabilities for the five rules sum to one: $\sum_{\ell=1}^{5} p(\ell(A)) = 1$. The probability term $p(ch(A)|X(A), \ell(A))$ is deterministic when A is the the root rule R_1, siding rule R_2, supporting rule R_3 and meshing rule R_4. It is set to be 1 if the rule A is selected and the associated attributes equations are all satisfied (see Section 2) otherwise it is zero. We set $p(ch(A)|X(A), \ell(A)) = 1$ if $\ell(A)$ is the affinity rule.

Likelihood Model

The likelihood model are defined on the layout segmentation of terminal nodes (i.e. superpixels) $p(I|C)$. We adopt the supervised model in [HEH] to classify regions to be one of the three geometric labels: ground, sky or vertical. Let j index the color of semantic region in C (clusters of superpixels), h_j the region features, v_j the region label determined by the method in [HEH], $\mathbf{c}(a)$ the color of the superpixel a, h_a the superpixel features. The probability $p(I|C)$ can be factorized

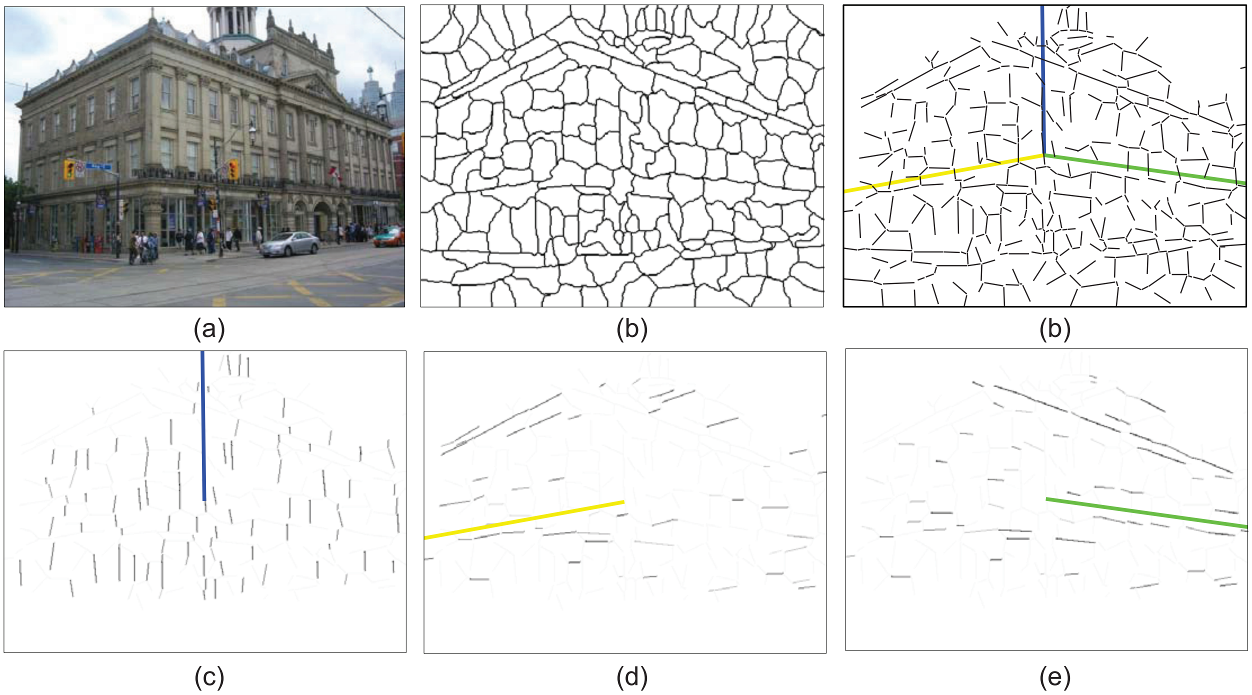


Figure 2.12: Augmented adjacent graphs for the Mesh Rule.

as,

$$P(I|C) = \prod_{j} p(v_j|h_j) \prod_{\mathbf{c}(a)=j,\mathbf{c}(b)=j} p(\mathbf{c}(a)=j,\mathbf{c}(b)=j|h_a,h_b) \tag{2.10}$$

where $p(v_j|h_j)$ is the label confidence in the geometric label v_j, $p(\mathbf{c}(a)=j,\mathbf{c}(b)=j|h_a,h_b)$ is the probability that superpixels a and b have the same geometric label, i.e. homogeneity likelihood. We follow the work in [HEH] to implement these two terms.

Inference

Given one image, our goal is to construct an optimal parse graph by sequentially applying the grammar rules to maximize a posterior probability as defined in the last section. This inference problem is challenging because: a) the parse graph does not have pre-defined structure; b) the attributes of graph nodes have to be passed to enforce the attributes constraint between parent-children nodes. We introduce an efficient bottom-up and top-down iterative procedure to construct the parse graph on the fly.

Our algorithm bases on the Swendsen-Wang Cut algorithm [BZ05] by Barbu and Zhu, which is essentially a cluster sampling procedure for graph coloring problem. It works on a adjacent graph iteratively following the MCMC design. At each step, it generates a cluster of connected component (CCP) by turning off the edges probabilistically, selects one CCP and changes the colors of the nodes in the selected CCP. The changes will be accepted with probability that usually integrates both the posterior probability and the proposal probability. The keys to the success of this clustering sampling algorithm include how to design proper adjacent graphs and how to make the informative solution proposals effectively .

We compare out method to two previous works, including the geometric parsing method by Hoiem et al. [HEH10], the method by Gupta et al. in [GEH10a]. Both methods can recover the three main geometric classes and the five vertical subclasses, whereas only [HEH10] ever reported the results of 3D reconstruction extensively. We use the default parameter configuration in their source codes

We first illustrate how to check the orthogonality between two horizontal VPs through one experiment on the image in Figure 2.9 where one vertical VP and four horizontal VPs are detected. Figure 2.14(a) plots the focal length (vertical direction) estimated from the vertical VP and each of the four horizontal VPs. We can observe the estimated focal lengths are roughly the same and the average focal length is 510 (unit). Figure 2.14(b) plots the focal length estimated from pairs of horizontal VPs by assuming they are orthogonal VPs. One could observe that none of them are close to the true focal length (i.e. 510 in this example). Therefore, for any pairs of horizontal VPs, we can estimate the focal length the method in [CDR99] and then check if it is close enough to the previously estimated value.

Qualitative Evaluations We first compare the 3D models recovered by our method to that by [HEH10]. Fig. 2.13 shows the results for two image in the CMU dataset [HEH10] and dataset LMW-A. As shown in Fig. 2.13 (c), the far-right region of building in orange is occluded by the vehicles and tree, and none of the previous methods can tell where is the contact line between this facade and the ground. In our method, however, parallel lines in this region suggest the contact line is likely to go through the VP in Green (shown in Fig. 2.13(b)) following local Manhattan world assumption. Thus, by reasonably assuming the contact line locates between these two regions in the image, we could simply perform an one-dimensional search for the optimal contact line that

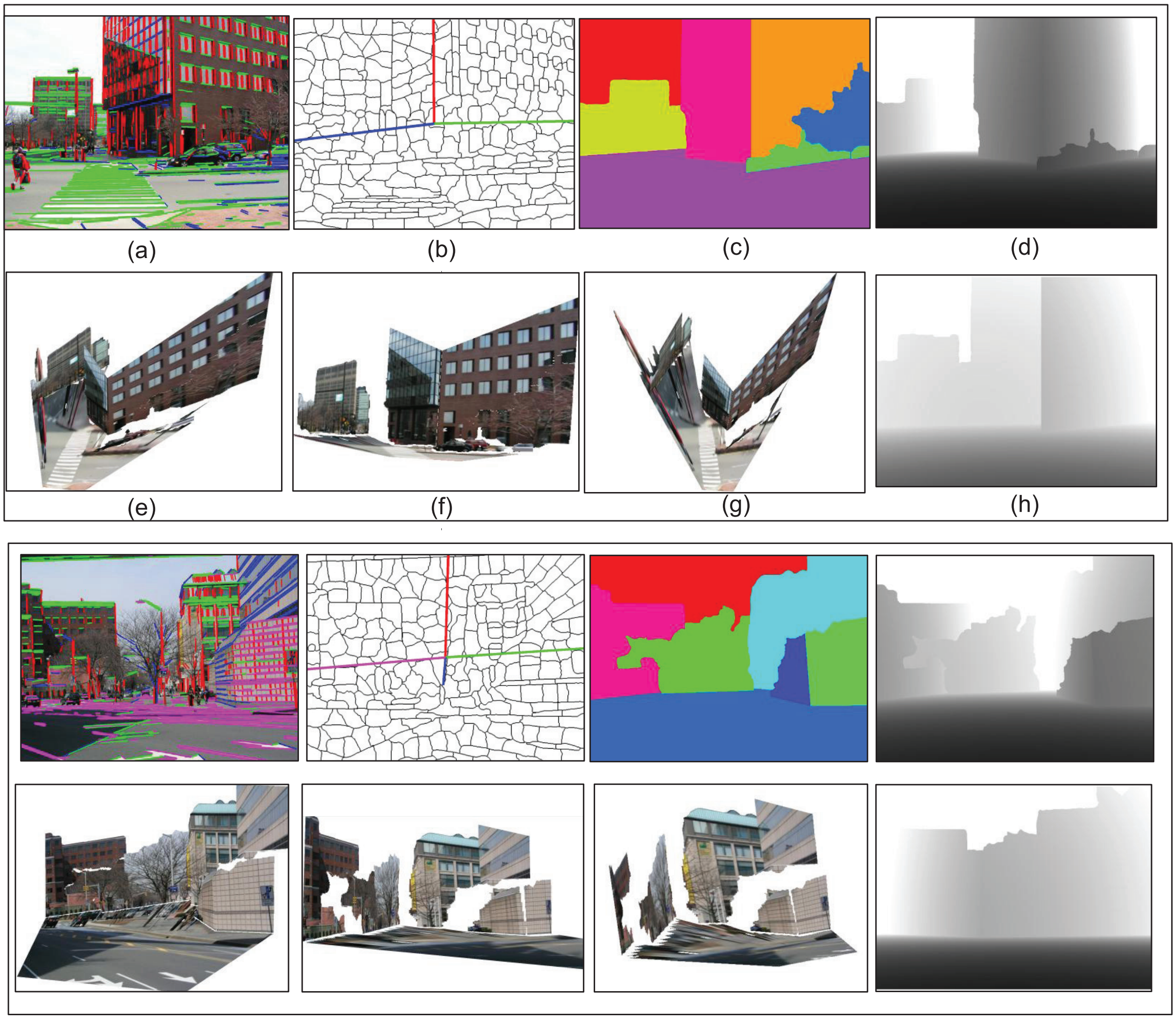


Figure 2.13: Experiment results of our method and Gupta et al. [GEH10a]. The image is from the CMU dataset [HEH10] (top block) and the dataset LMW-A (bottom block). Each block shows our results of: (a) original image overlaid parallel lines; (b) superpixels patition over laid lines linking image center and VPs; (c) layout segmentation; (d) depth map; and (e)-(g): three synthesized views, and the Result of [HEH10]: (h) depth map .

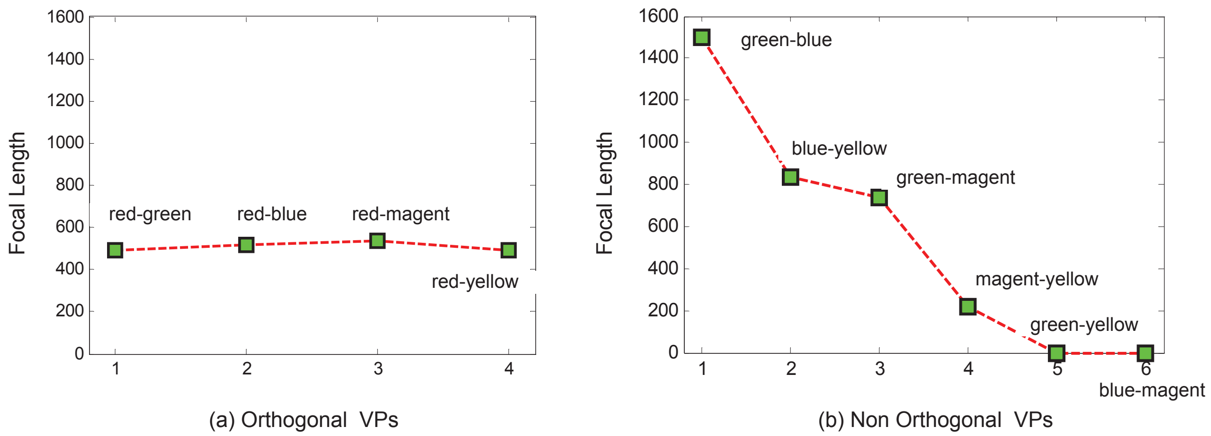


Figure 2.14: Focal length estimation for the image shown in Fig. 2.9. See texts for details.

maximizes the posterior probability. One can observe the estimated contact line in Fig. 2.13 (c) is very accurate. In this way, the 3D reconstructions of the 2D regions are well regularized by the Local Manhattan World assumption.

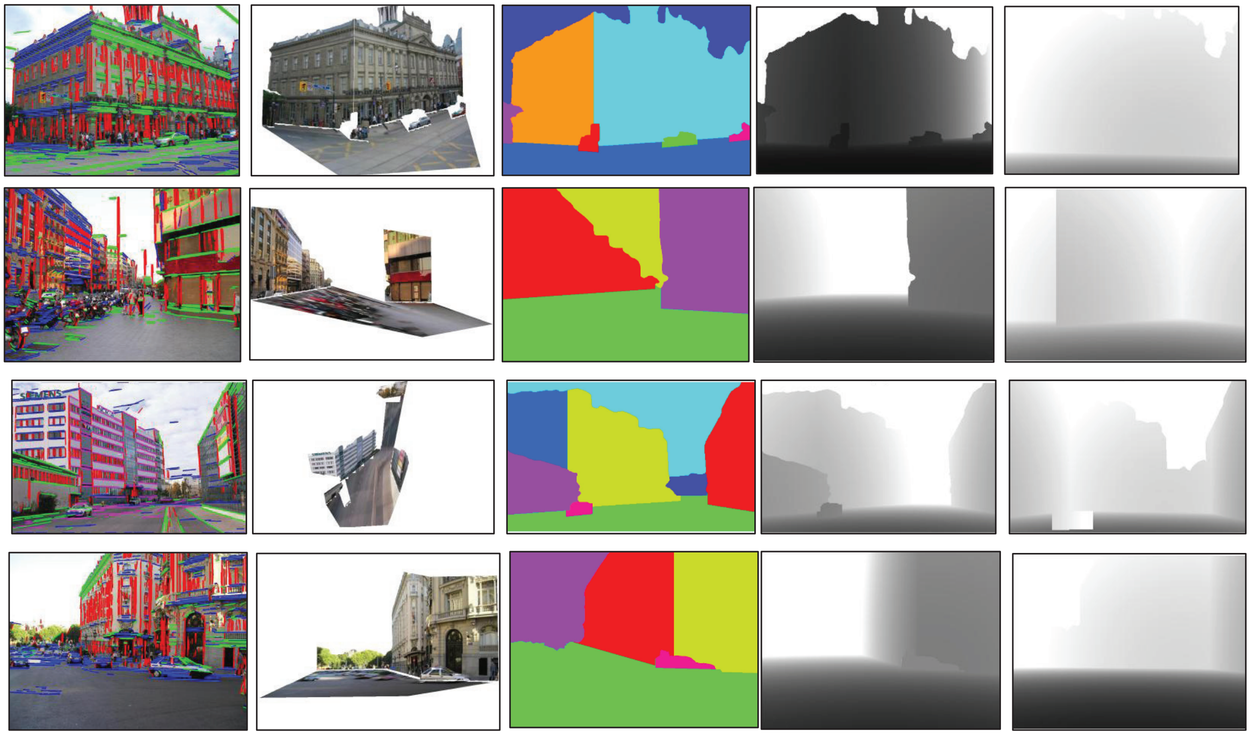


Figure 2.15: More experiment results on the CMU dataset(first row) and the LMW-A dataset (othe rows). Columns 1-4 show our results, including: families of parallel lines; newly synthesized view; layout segmentation; and depthmap. Column-5 shows the depthmap by Hoiem et al. [HEH10] .

In Fig. 2.15, we analyze how our method and [HEH10] perform on Manhattan type images and Local-Manhattan type images. As shown, the image in the firt row follows the typical Manhattan World assumption, while other images only follow the Local Manhattan World assumption as they contain more then 2 horizonal VPs or the horizontal VPs are not orthogonal with each other. For the first image, both [HEH10] and our method can produce reasonable depth maps. For the other images, [HEH10] tends to assign the same depth to the surfaces of 'vertical', whereas our method, in contrast, can still produce high-quality depth maps. These exemplar results well demonstrate how the LMW knowledge propagate through the grammar rules to create accurate 3D models.

Quantitative Results We further report the numerical comparisons between various methods in term of surface orientation estimation and region segmentation. For surface orientation estimation, we use the metric of accuracy, calculated by the percentage of pixels that have the correct label and averaged over the test images. Since both our method and the two baselines can achieve high performance on the estimation of main geometric classes ('ground','vertical', and 'sky'), we focus on the vertical subclasses, like [GEH10a]. We discard the superpixels belonging to ground and sky and evaluate the performance of all methods. Table 2.2 reports the numerical comparisons. The method by Gupta et al. [GEH10a] has an average performance of 73.72%, whereas ours performs at 76.34% on the dataset by Hoiem et al [HEH10]. On the other two datasets that have accurate surface orientation annotations, the improvements by our method are even more, i.e. 5.18 percentages and 4.1 percentages respectively. As stated by Gupta et al. [GEH10a] , improving vertical subclass peformance is known to be hard. Our method, however, can improve these two baselines with large margins.

We also evaluate the segmentation performance on the three datasets. We use the best spatial support metric in [GEH10a], which first estimates the best overlap score of each ground truth segment and then averages it over all ground-truth segments. Table 2.3 report the numerical comparisons on the three datasets. Our method improves the method [GEH10a] with the margins of 3.86,5.24,4.86 percentages on the three datasets, respectively.

	dataset in [HEH10]	LMW-A	LMW-B
Our method	76.34 %	67.9 %	64.3 %
Gupta et al. [GEH10a]	73.72 %	62.21 %	59.21 %
Hoiem et al. [HEH10]	68.8 %	56.3 %	52.7 %

Table 2.2: Numerical comparisons on surface orientation

	dataset in [HEH10]	LMW-A	LMW-B
Our method	72.71%	66.45%	65.14 %
Gupta et al. [GEH10a]	68.85%	59.21%	60.28%
Hoiem et al. [HEH10]	65.32 %	58.37%	57.7 %

Table 2.3: Numerical comparisons on segmentation

CHAPTER 3

Image Understanding by Integrating Function, Geometry, and Appearance

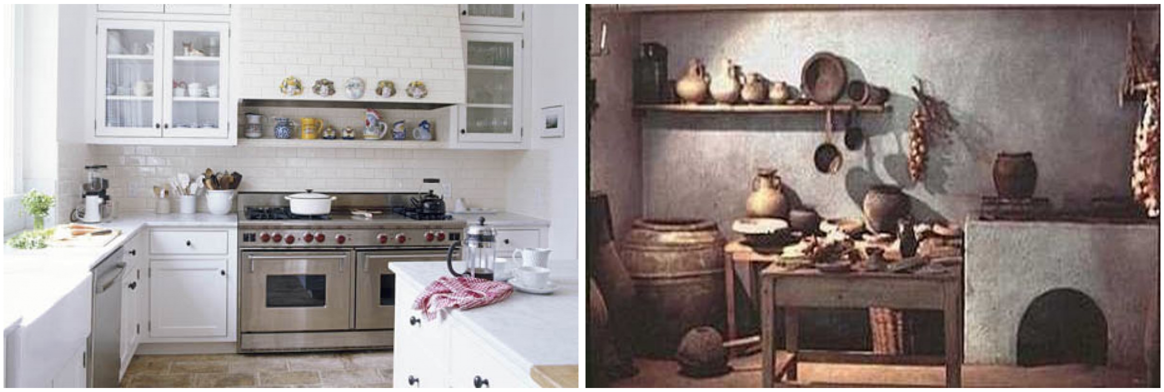


Figure 3.1: A modern kitchen and an ancient kitchen with similar functions but drastically different geometry and appearances.

3.1 Introduction

3.1.1 Motivation and Objective

In the past 15 years, a prevailing approach in the vision literature has been posting scene recognition as a classification problem – classifying scene categories, recognizing scene attributes, and detecting objects through appearance-based features, machine learning techniques, and large training examples. Such approach essentially memorizes the typical examples in each scene or object categories, does not "understand" the real meanings of objects and scenes, and thus is known to have difficulties in generalizing and extrapolating into unseen features spaces.

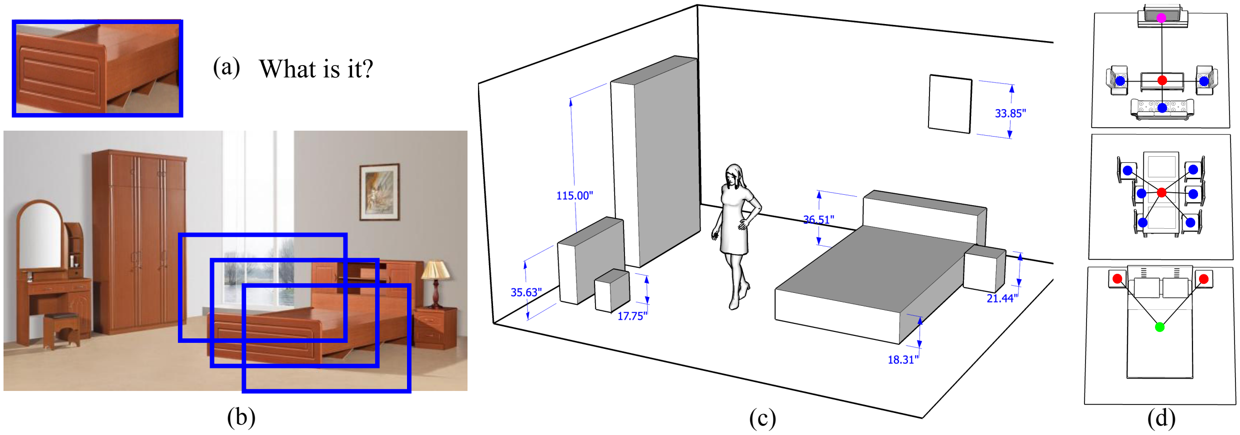


Figure 3.2: (a) A large image window cropped from an input image in (b). The window is hardly recognizable in traditional appearance-based recognition but can be recognized in the whole scene. (c) An imagined human pose and estimated geometric sizes of objects in 3D, from which *functions* are reasoned; (d) *contextual relations* of functional objects as groups.

One example is shown in Figure 3.1. Taken from a similar viewing angle, the two images have drastically different appearance and geometry, but are both considered kitchen by human vision. What are common to the two images are the functionality of objects and the 3D spaces in serving a set of human actions – preparing food.

Functionality refers to the property of an object or scene, especially man-made ones, which has a practical use for which it was designed. Psychologist [Gib77] used another term, *affordance*, which refers to the property of an object that affords the opportunity for humans to perform some specific actions. From such view point, we argue that

- *objects, especially man-made ones, are defined by their functions and actions that they are involved.*
- *scenes, especially man-made ones, are defined by the activities and actions that they can provide space for.*

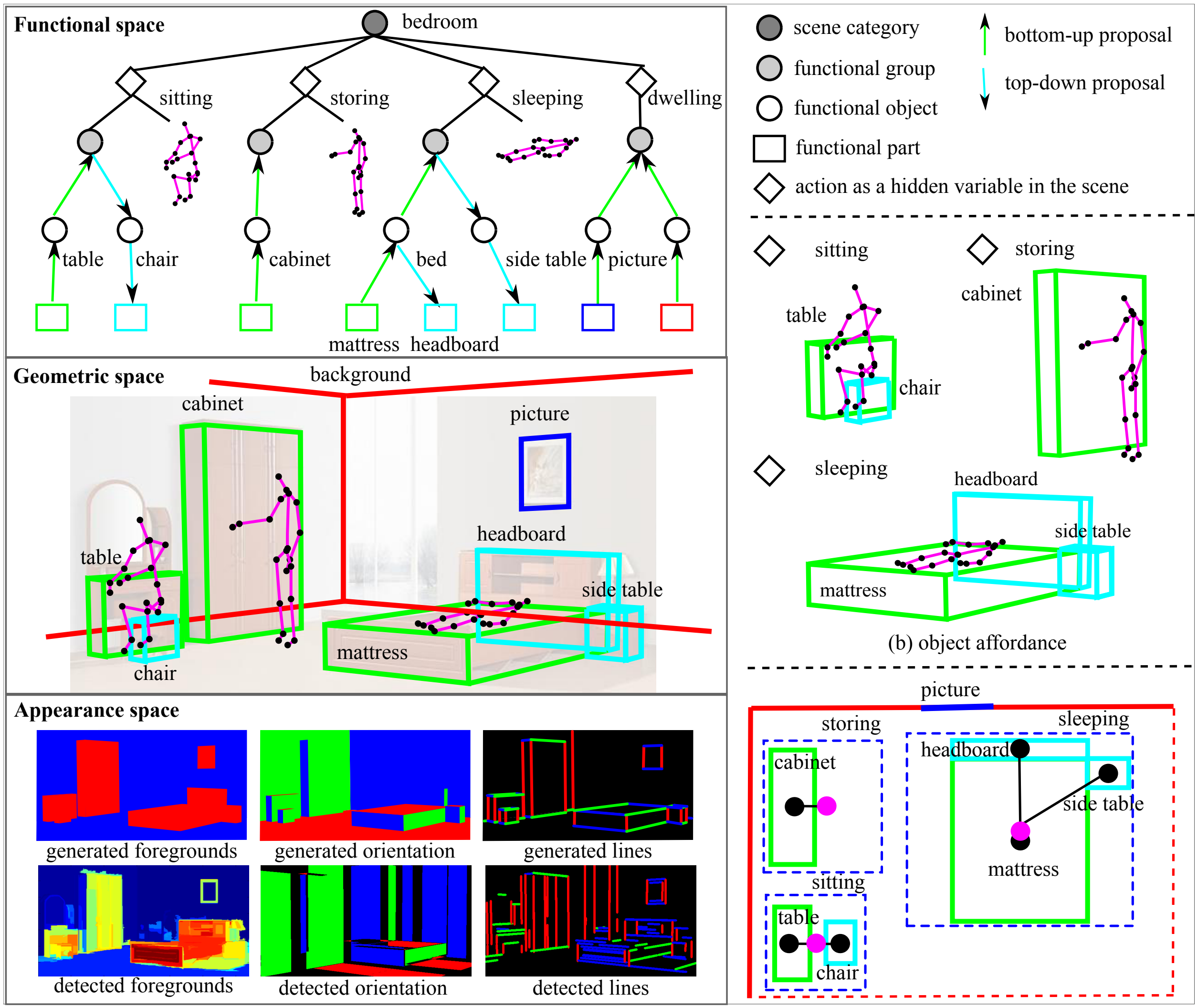
So, functionality is deeper than geometry and appearance and thus is a more invariant concept for scene understanding.

This represents a different philosophy that views vision tasks from the perspective of agents, that is, agents (humans, animals and robots) should perceive objects and scenes by reasoning their plausible functions. We believe this perspective is a more robust way and will take us to deeper human-like scene understanding systems.

Motivated by the above observations, this paper poses scene understanding as an image parsing problem following the work of [TCY05] and is aimed at two objectives in the following.

Our first objective is to present a Stochastic Scene Grammar (SSG) as a hierarchical compositional representation which integrates functionality, geometry and appearance in a FGA hierarchy. For example, Fig. 3.3 shows a parse tree derived from this grammar in the joint FGA spaces for a bedroom image. In contrast to traditional syntactical parsing advocated by [Fu82], the scene (root node) is defined by a set of most probable actions (diamonds) that may occur in the scene. The actions are reasoned based on the geometry of the objects and imagined human skeletons, as Fig. 3.2.(c) illustrates. Such human object interaction models can be learned offline through RGBD videos, *e.g.*[WZZ13]. The geometric objects are grouped from line segments extracted from image appearances. In Fig. 3.2(c), the geometric dimensions of furniture in the room are designed to fit the sizes of humans. For example, any flat surface for sitting is usually 18 inches tall, *i.e.*knee height, and a place to sleep is usually between 6-8 feet. Moreover, the contextual relations between the furniture pieces are helpful in distinguishing their functions and therefore assigning their names, *e.g.*the nightstand is near the bed and the lamp is on top of the nightstand. Some typical functional groups are illustrated in Fig. 3.2(d).

Our second objective is to present an effective algorithm for inferring the FGA hierarchy, *i.e.*parse trees, from a single input image. Due to the flexibility of 3D objects in the space and their contextual relations, it is ineffective to use the prevailing sliding window methods for object detection, and it is also infeasible to search objects of all dimensions in an image pyramid, we adopt a Markov chain Monte Carlo method to optimize the Bayesian a posteriori probability. In the spirit of data-driven MCMC proposed by [TZ02], our parsing algorithm consists of a set of Markov chain dynamics which, in combination, can traverse the entire joint FGA space. For computational efficiency, these MC dynamics are driven by proposal probabilities computed in bottom-up steps.



(a) the Function-Geometry-Appearance hierarchy

(b) object affordance

(c) contextual relations from an overhead view

Figure 3.3: (a) The function, geometry and appearance (FGA) hierarchy in our proposed scene parsing grammar. The scene category (bedroom) at the root note is defined by the background and three most likely actions (sitting, storing and sleeping) in the scene. These actions impose the object affordance and contextual relations to the geometric entities. The final parsing result is evaluated on top of the synthesis of appearance likelihood maps. (b) The 3D human-object interactions. (c) The contextual relations between objects.

3.1.2 Overview of Our Approach

By analogy to natural language parsing, we pose the scene understanding problem as parsing an image in a hierarchical *parse tree* (or parse graph if we count on the spatial context relations) using the Stochastic Scene Grammar (SSG). Fig. 3.3 shows an example of the parse tree in a Function-Geometry-Appearance (FGA) hierarchy. In comparison to the literature reviewed above, this paper has two major contributions to the scene parsing problems.

(I) A Function-Geometry-Appearance hierarchy.

We embed the FGA hierarchy in the syntactic grammar discussed above, and Fig. 3.3.(a) illustrates the FGA hierarchy in three layers.

Functionality. In the top layer, an indoor scene is defined by a small set of plausible human actions, and each action involves a few objects as a group. The table and chair (and the mirror) for a person to sit (and to make up face/hair), a bed with side table (and lamp) for people to sleep (and read). Here, each action is a composition of the 3D geometric relations between the pose and objects, as Fig. 3.3.(b) shows.

Geometry. The 3D sizes (dimensions) are used to evaluate how likely an object is able to afford a human action, known as the *affordance* in [Gib77]. Fortunately, most furniture has regular structures, *i.e.*rectangular shapes, therefore the detection of these objects is tractable by inferring their geometric affordance. For objects like sofas and beds, we use a more fine-grained geometric model with compositional parts, *i.e.*a group of cuboids. For example, the bed with a headboard is a better explanation of the image in terms of segmentation accuracy as shown at the bottom of Fig. 3.3. In the geometric space, each 3D shape is directly linked to a concept in the functional space. Shown in Fig. 3.3.(c), the contextual relations are utilized when multiple objects are assigned to the same functional group, *e.g.*a bed and a nightstand for sleeping. The distribution of the 3D geometry is learned from a large set of 3D models as shown in Fig. 3.6.

Appearance: The appearance of the furniture has large variations due to material properties, lighting conditions, and viewpoints. In order to ground our model on the input image, we detect and estimate line segments, surface orientations, and coarse foreground detection as the local evidences to support the geometry reasoning above as Fig. 3.3 illustrates.

(II) MCMC inference algorithm with reversible jumps.

We design a MCMC algorithm to simulate a Markov Chain to traverse the space defined by the FGA hierarchy in a data driven MCMC paradigm proposed by [TZ02].

The MCMC includes three types of dynamics for reversible jumps: i) add: sample a subtree and attach it to a non-terminal node randomly chosen from the current parse tree; ii) delete: delete a subtree whose root is a node randomly chosen from the current parse tree; iii) functional jump: switch a functional label of a node randomly on the current parse tree.

The inference algorithm also includes three types of geometric diffusion moves: i) α-diffusion: data-driven bottom-up detection that directly draws cuboid proposals from a non-parametric distribution built up by the line segments detected from the image;

ii) β-diffusion: grammar-driven bottom-up prediction that proposes cuboid for a parent node in the parse tree from the children nodes by inversely computing a geometric transformation;

iii) γ-diffusion: grammar-driven top-down prediction that proposes cuboid by top-down sampling for a child node in the parse tree from its parent node based on the geometric model.

3.2 Integrating Function, Geometry and Appearance

The previous section overview stochastic context sensitive grammar and its general probabilistic formulation. In this section, we elaborate on how the SSG integrates the three layers of concepts in the functional space, the geometric space and the appearance space.

In this section, we will explain two production rules (the functional set rule, the affordance rule) for generating graph nodes on the grammar. And an image formation process that evaluates the generated 3D scene by rendering the synthetic image.

3.2.1 The Functional Space

The grammar model has advantages to handle the compositionally of the visual entities as well the dimensional changes of the scene. For example, it is common that a bedroom either has one bed or has two beds. The traditional grammar deals with the dimensional change by recursive production rules, such as $A \to \alpha \cdot A$. The production rules defined in the functional space are not recursive.

We introduce a set rule in functional space as

$$v \to \{l, \{G(u_i) : i = 1 \cdots \#(l)\} : l \in L\} \tag{3.1}$$

where l is a label of a child node from a label set L, $\#(l)$ is a number variable controlling the number of objects for each label l. The set rule is a nested OR-AND node. As shown in Fig. 3.4, the number variable decides the dimensionality of the parse tree. Therefore the production of the functional set rule can generate various parse trees with different dimentionalities.

Thus, the probability distribution of each production rule $P(r : v \to Ch_v)$ in Eq. 2.3 is unfolded as

$$P(v \to Ch_v) = \prod_{l \in L} \left[P(\#(l)) \prod_{i \in \{1 \cdots \#(l)\}} P(G(u_i)|l) \right] \tag{3.2}$$

The geometric attributes $G(u_i)$ of each object u_i is defined in the geometric space.

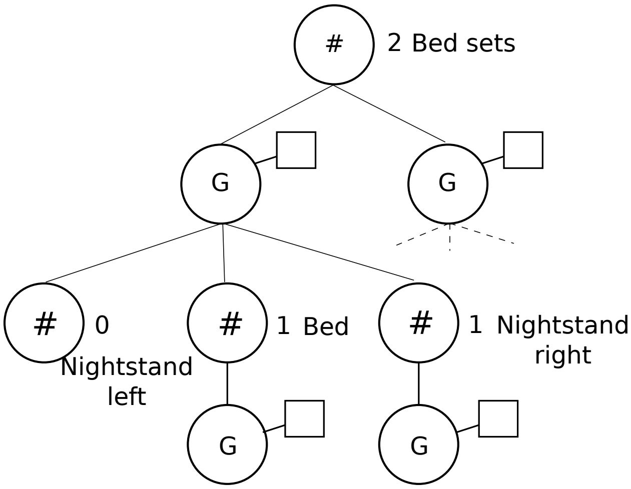


Figure 3.4: An example parse tree generated from the grammar with the set rule. The dimensionality of a parse tree is decided by number variables for each label.

3.2.2 The Geometric Space

We model each geometric entities in the grammar as a 3D cuboid.

Each 3D cuboid is encoded by three geometric attributes including 3 DoF size $Size(v)$, 3 DoF relative position $Pos(v)$ and 1 DoF relative orientation $Ori(v)$,

$$G(v) = \{Size(v), Pos(v), Ori(v)\} \tag{3.3}$$

Object affordance $p(Size(v)|l(v))$ models the distribution of geometric attributes of each functional object, for example, how large the bed mattress is, how far the bed is from the wall. If we consider human actions as hidden variables in the space, then the affordance probability measures how likely the geometric shape of an object is able to afford an action. As shown in Fig. 3.3, a cube around 1.5ft tall is comfortable to sit on despite its appearance, and a ”table” of 6ft tall loses its original function – to place objects on while sitting in front of.

We model the 3D sizes, relative position, and relative orientation of functional objects by a mixture of Gaussians respectively, such as

$$p(Size(v)|l(v)) = \sum_{i=1}^{K} a_i N(\mu_i, \Sigma_i) \tag{3.4}$$

where the a_i is the mixture coefficient of each Gaussian $N(\mu_i, \Sigma_i)$. The model characterizes the sub-category of the geometry, which allows for simultaneous alternatives of canonical sizes, such as king size bed, full size bed *etc*. We estimated the model by EM clustering, and we manually picked a few typical samples as the initial mean for the Gaussian, *e.g.*a coffee table, a side table and a desk from the table category.

The contextual relations are defined with respect to the relative position $Pos(v)$ and relative orientation $Ori(v)$. The relative position is the position of a child with respect to the parent coordinate system. The relative orientation is the orientation of the child with respect to the reference orientation of the parent.

The absolute coordinates of an object can be calculated recursively along the grammar productions. We showed an example of the geometric transformation between a child coordinate system X and a parent coordinate system X' in Fig. 3.5. The transformation can be decomposed as two independent transformation H_1 and $H2$. The H_1 represent the transformation from child coordinate system to its center of mass coordinate system, and the H_2 represents the transformation from its center of mass coordinate system to its parent coordinate system. The geometric transformation equation is calculated by

$$
\begin{aligned}
X' &= H_2 H_1 X \\
&= \begin{bmatrix} cos(Ori) & sin(Ori) & 0 & Pos_x \\ -sin(Ori) & sin(Ori) & 0 & Pos_y \\ 0 & 0 & 1 & Pos_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & Size_x/2 \\ 0 & 1 & 0 & Size_y/2 \\ 0 & 0 & 1 & Size_z/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} X
\end{aligned}
\tag{3.5}
$$

In order to learn the geometric model, we collected a dataset of functional indoor furniture, as shown in Fig. 3.6. The functional objects in the dataset are modeled with real-world measurements, and therefore we can generalize our model to real images by learning from this dataset. We found that the real-world 3D sizes of the objects has less variance than the projected 2D sizes. As we can see, these functional categories are quite distinguishable solely based on their geometric shapes as shown in Fig. 3.7. For example, the coffee tables and side tables are very short and usually lower than the sofas, and the beds generally wider than others.

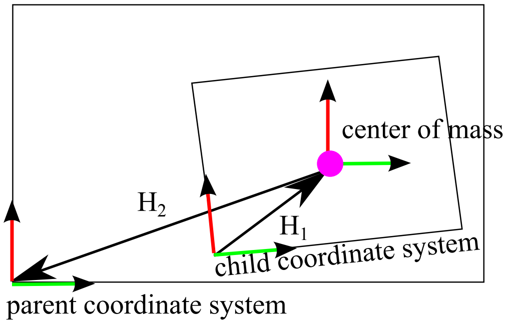


Figure 3.5: The geometric transformation between a child coordinate system and a parent coordinate system.

In this version of algorithm, we directly model the cooperative ”+” relations as parent-child geometric relationship as discussed above without explicitly addressing the cooperative ”+” relations among child nodes as [ZZ11]. This parent-child relation facilitates the inference algorithm traveling along the depth of the hierarchy, *e.g.*the top-down prediction and bottom-up prediction in Sect. 3.3.

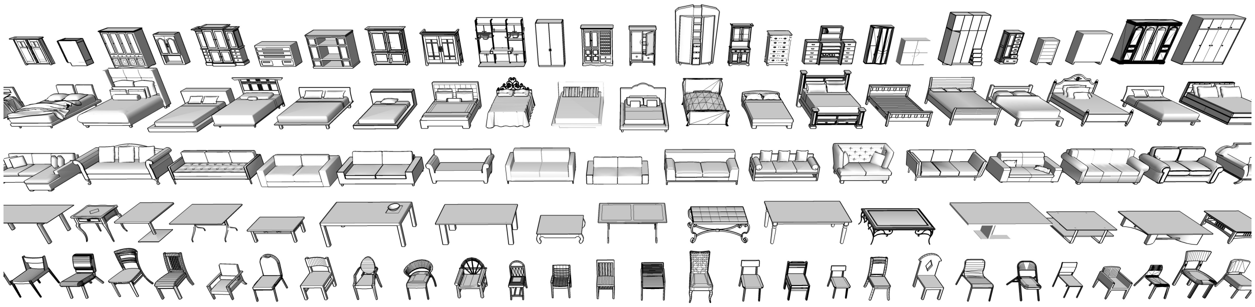


Figure 3.6: Examples of 3D indoor furniture products collected from the Trimble 3D Warehouse.

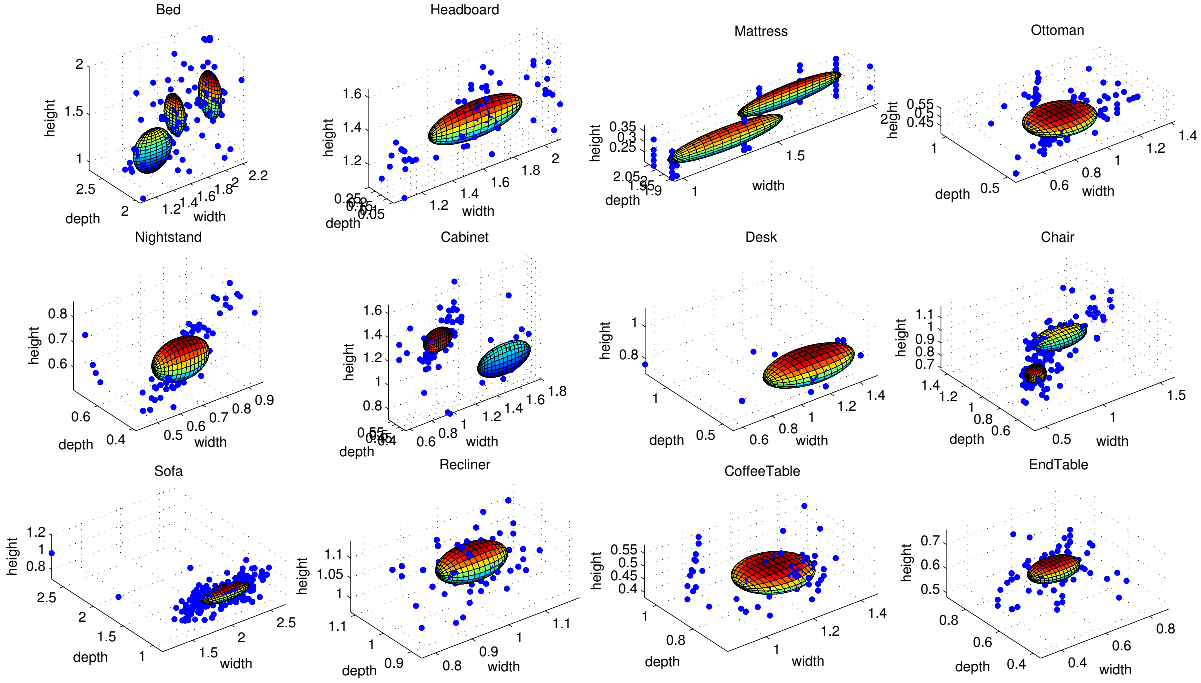


Figure 3.7: The empirical and fitted distributions of the 3D sizes of some functional objects in meters plotted in 3D spaces.

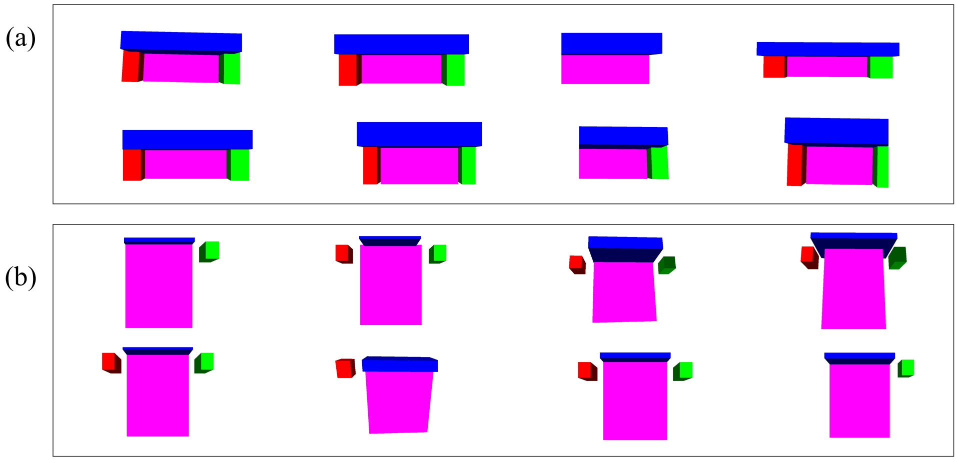


Figure 3.8: Samples drawn from the distributions of 3D geometric models (a) the functional object "sofa" and (b) the functional group "sleeping".

However, we still model the competitive "-" relations specify penalties or constraints among child nodes. The sufficient statistics is defined on the penetrating rate between their occupied 3D spaces $G(.)$ to penalize penetrating objects.

$$h^{-}(G(Ch_v)) = \sum_{a,b\in Ch_v} (G(a)\cap G(b))/(G(a)\cup G(b)), . \tag{3.6}$$

3.2.3 The Appearance Space

The functional and geometric hierarchies are generative models on a cartoon like image with line segments for object boundaries and labeled regions for object surfaces. There is still a gap between the synthesized cartoon scene and the observed image. To fully explain (reconstruct) the input image, we need to know the lighting conditions, textures and material properties for object surfaces. This is a challenge problem which is beyond the scope of this paper.

To circumvent the tasks of modeling and inferring textures and lighting conditions, we use of set discriminative methods to detect some intermediate results in the following.

- a map of line segments detected by an algorithm proposed in [VJM10];
- a foreground/background label map computed by an approach used in [HHF09]; and
- a surface orientation map calculated by an approach in [LHK09].

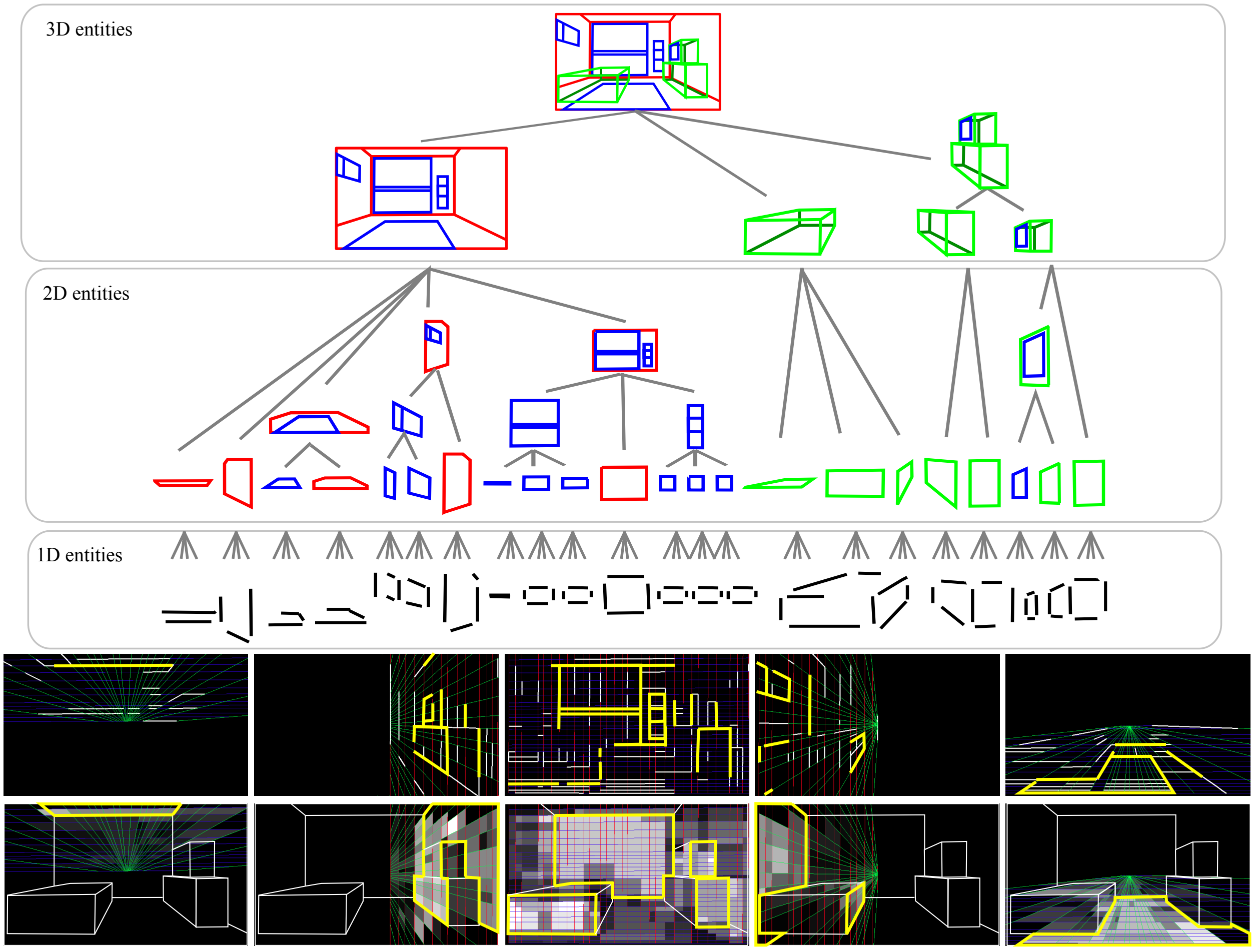
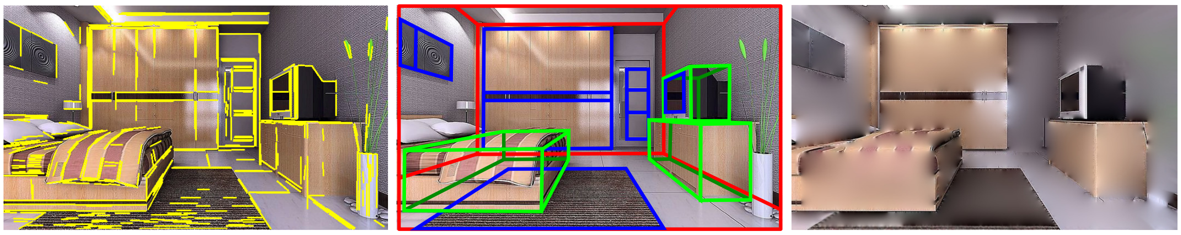


Figure 3.9: The decomposition of geometric parse tree. The ten images on the bottom show the likelihood of the parse graph calculated and quantized by the five major orientations, whose normal directions point to down, left, front, right, and up respectively. The first five images show line segments (yellow) detected on their corresponding orientations, and the second five images show region likelihood calculated on their correspondent orientations. The lighter a cell, the higher the probability is. The yellow contours outline the inferred regions.



(a) input image with line segments (b) geometric parsing result (c) image reconstructed via sturectures in (b)

Figure 3.10: Input image and output results of the geometric parsing.

Thus instead of grounding our model on raw pixels, we define the likelihood model on these 2D label maps using a Sum of Squared Difference (SSD) function $d()$.

$$\begin{aligned} p(I_{obs}|pt) &= p(I_{label}|I_{syn} = f(V^T)) \\ &= 1/Z * \exp\left(-\sum_{i\in 1\cdots 3} \lambda_i d(I^i_{label}, I^i_{syn})\right) \end{aligned} \tag{3.7}$$

where $f(V^T)$ is a rendering function of all the terminal nodes V^T. The rendering function generates the synthesized image I_{syn}, and the likelihood is defined how likely the parse graph generate label maps I_{label}.

For example, the appearance space is illustrated at the bottom of Fig. 3.3(a). The figure shows the three detected line segment map, foreground segmentation map, and orientation map from left to right. Above the three maps are the corresponding maps rendered from the parse tree pt. Once a parse tree pt is decided, the algorithm projects the 3D geometric entities V^T on the parse tree to the 2D image plane with respect to the relative depth order and camera parameters. The projection is implemented with OpenGL.

3.3 Inference Algorithm

We design a top-down/bottom-up algorithm to infer an optimal parse tree pt. The compositional structure of the continuous geometric parameters and discrete functional labels introduces a large solution space, which is infeasible to enumerate all the possible explanations. Neither the sliding windows (top-down) nor the binding (bottom-up) approaches can handle such an enormous number of configurations independently.

3.3.1 Reversible Jumps

In this paper, we design Markov chains with reversible jumps (RJMCMC) algorithm to construct the parse tree and re-configure it dynamically using a set of moves. Formally, our scene parsing algorithm simulates a Markov chain $\mathcal{MC} =< \Omega, v, \mathcal{K} >$ with kernel $\mathcal{K}$ in space Ω and with probability v for the starting state. We specify stochastic dynamics by defining the transition kernels of reversible jumps. For each Markov chain move is defined by a kernel with a transition matrix $\mathcal{K}(pt^*|pt : I)$, which represents the probability that the Markov chain make a transition from state pt to pt^* when a move is applied.

The kernels are constructed to obey the detailed balance condition:

$$p(pt|I)\mathcal{K}(pt^*|pt : I) = p(pt^*|I)\mathcal{K}(pt|pt^* : I). \tag{3.8}$$

Kernels which change the graph structure are grouped into reversible pairs. For example, the kernel for node creation $\mathcal{K}_+$ is paired with the kernel for node deletion $\mathcal{K}_-$ to form a combined move of node switch. To implement the kernel, at each time step the algorithm randomly selects the choice of move and then uses kernel $\mathcal{K}(pt^*|pt : I)$ to select the transition from state pt to state pt^*. Note that the probability $\mathcal{K}(pt^*|pt : I)$ depends on the input image I. This distinguishes our algorithms as a Data-Driven MCMC from conventional MCMC computing ([TZ02, TCY05]).

The kernel is designed using proposal probabilities and correspondent acceptance probability.

$$\mathcal{K}(pt^*|pt : I) = Q(pt^*|pt : I)\alpha(pt^*|pt : I) \tag{3.9}$$

The acceptance probability follows:

$$\alpha(pt \to pt^*) = min\{1, \frac{Q(pt|pt^*, I)}{Q(pt^*|pt, I)} \cdot \frac{P(pt^*|I)}{P(pt|I)} J_{f_{pt \to pt^*}}\} \tag{3.10}$$

$J_{f_{pt \to pt^*}}$ is the Jacobian of the dimension matching function. $f_{pt \to pt^*}$ is the dimension matching function. It is used to map the variables at dimensionalities of pt and pt^* into a space of common dimensionality. It is usually done by introducing additional $pt^* - pt$ parameters, or projecting out the corresponding $pt^* - pt$ parameters. Notice that each variable in Δpt is independently sampled from pt, hence the Jacobian is 1 in this case ([YYW12]).

The Metropolis-Hasting form ensures that the Markov chain search satisfies the detailed balance principle. A simulated annealing technology is also used to find the maximum of complex posteriori distribution with multiple peaks while other approaches may trap the algorithm at local optimal peaks. The parse tree is initialized with random number of object and random geometric properties. During each iteration, if a proposal increases the posterior probability with respect to the proposal ratio, the move is taken. Otherwise, the move is taken only with a certain probability, which decreases over time. Hence early on the algorithm will tend to take moves even if they don't improve the probability. Later on, the algorithm will only make moves which improve the posterior probability. The temperature function used is: $T(n) = 1000/n$ where n is the iteration number.

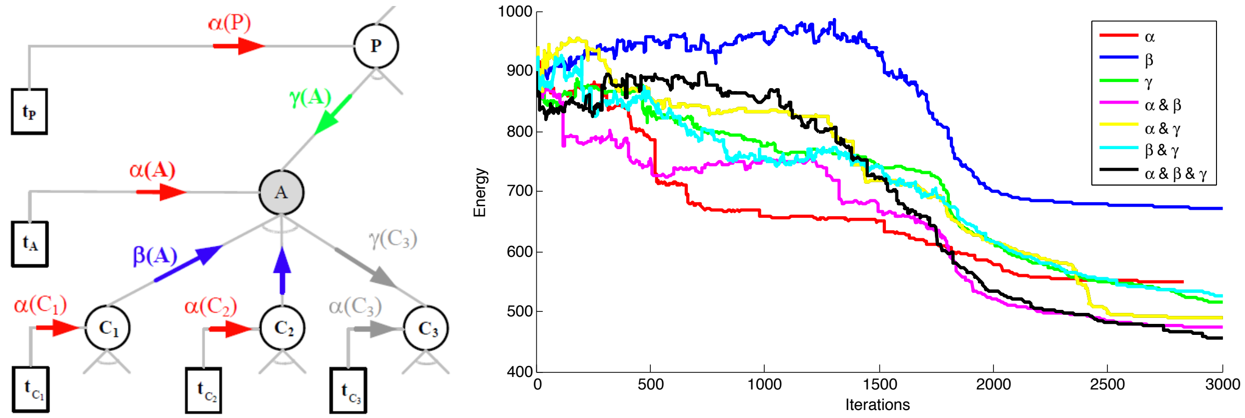


Figure 3.11: The bottom-up top-down proposals for geometric diffusion moves. Our inference algorithm generates three kinds of geometric diffusion proposals: α: bottom-up detection, β: bottom-up prediction, and γ: top-down prediction. The plot on the right panel shows the average energy convergence of hundreds of Markov Chains using different proposal strategies: By only using the α diffusion from bottom-up detection (red curve), the Markov chain converges very fast at the beginning, but cannot keep reducing the energy due to limitation of bottom-up detections. Using the β diffusion from bottom-up prediction (blue curve) is the worst strategy, because if the terminal node can not be optimized, the prediction from bottom-up can be very bad. The black curve which combines three diffusions together is the best strategy, it has sufficient exploration at the beginning, and gradually converges to the lowest energy. Besides that, the combination of $\alpha\&\beta$ (magenta curve) and the combination of $\alpha\&\gamma$ (yellow curve) achieve good results which are very close to the black one.

Algorithm 1: Inference algorithm

```
Data: an input 2D image
Result: an output parse tree
Calculating data-driven 3D proposals;
while the rejection time larger than $K$ do
    Choose one of the following moves randomly;
        • add an entity
        • delete an entity
        • diffuse geometric attributes of an entity
    if add/remove a non-terminal node then
        Recursively add/remove its children;
    end
    if diffuse a non-terminal node then
        Choose one of the following geometric diffusion moves randomly;
            • $\alpha$ diffusion from bottom-up detection
            • $\beta$ diffusion from bottom-up prediction
            • $\gamma$ diffusion from top-down prediction
    end
    Calculate the posterior probability and validate the solution by projecting the 3D parse
    tree to the 2D image plane;
    Accept/reject the new parse tree with the acceptance probability;
end
Return the parse tree with the highest posterior
```

3.3.2 Generating Data-Driven 3D Proposals

The algorithm starts from detecting straight line segments by [VJM10]. Based on the Manhattan assumption, we group the line segments into N groups, each of which is correspondent to a vanishing point. We then select three dominate orthogonal vanishing point to build our coordinate system. We assume the camera parameters are reliably calibrated in this step, the calibration algorithm is discussed in Sect. 3.3.2.3.

We incrementally group noisy line segment into larger geometric structures. The 2D rectangles are formed by filtering over the combinations of two pairs of parallel lines or T junctions. As shown in Fig. 3.9, we first define five normal directions: facing down, facing left, facing front, facing right and facing up according to the vanishing points. The normal direction facing back is not visible from the camera position. All the 2D line segments are aligned on the mesh for each normal orientation. And surface orientation maps and foreground maps are also projected to each cell. And our algorithm goes over each rectangle on the mesh and calculates a local likelihood normalized by the size of the rectangle according to Eq. 3.7. In this way, we detect an exhaustive set of 2D rectangle candidates by applying a threshold for a high recall rate. Similarly, the cuboids are formed by filtering over the combinations of any two hinged rectangles, a threshold is applied to the distance between rectangle corners to evaluate how well the structure is formed. Please refer to [ZZ11] for more details.

3.3.2.1 The Composition of 3D Geometric Entities

As shown in Fig. 3.9, the geometric space $\mathcal{G}$ contains the geometric entities of 3D cuboids, 2D rectangles and 1D line segments. Each entity is composed by several lower dimensional shapes. The detection of 3D entities starts from detection of line segments in the 2D image space as shown in Fig. 3.10(a). The composition of the geometric entities is coded by a series of AND rules where the relations between children nodes are set to a constraint within a threshold. The threshold is set to 5 pixels in the image, which means we tolerate 5 pixels offset between those rigidly combined components. The OR rule also plays a role by representing alternative ways of composition under different the view points. The production rules of geometric composition is illustrated in Fig. 3.9.

We project all the terminal primitives to five normal directions as discussed above.

3.3.2.2 The Calculation of Marginal Likelihood

The probability of the proposal is calculated by local marginal likelihood based on the bottom-up image labelling results. In order to properly quantize the geometric space and speed up the computation, we first group detected line segments into three main groups corresponding to three vanishing points. Then we further group the line segments into a series of rays pointing from the vanishing points to each line segments. We enforce the angle between two nearby rays to be larger than 2°, therefore line segments along the same orientation will be grouped together. We will also interpolate rays between two nearby rays if the angle between them are larger than 5°. Any two groups of rays will form an oriented mesh as shown at the bottom of Fig. 3.9. This quantization process guarantee that each detected line will be represented by several pieces of edges on the mesh, and each pixel will fall into a cell as well. In this way, the line/region likelihood of bottom detection is stored in the quantized meshes for each surface orientation. The brighter the intensity the higher the likelihood for each cell.

At the bottom of Fig. 3.9, there are ten images. The yellow lines on the first row of images represent the activated line segments. The line segment is activated when the geometric parsing result in Fig. 3.10.(b) match with the bottom-up detection result in Fig. 3.10.(a). The edge probability measures how many line segments are activated, which implicit encourages more line segment to be explained by final parsing result. the region with yellow boundary on the lower penal represent the activated surface region. A surface region is activated only the surface orientation is matched with geometric parsing results in Fig. 3.10.(b) by considering the depth ordering. The depth ordering guarantee the occluded region will not affect the likelihood of parsing result. Therefore, the quantization of image likelihood not only accelerates the inference process by a lookup table of pre-computation, but also avoids the double counting of the shared edges and the occluded regions.

From the geometric primitives and their line segments, we can reconstruct the 2D image using a primal sketch model proposed in [GZW07] which was also used in [HZ09] for scene synthesis. 3D reconstruction results are shown in Fig. 3.14.

3.3.2.3 Single View 3D Scene Reconstruction

After detect each 3D line drawing cuboid, we need to recover the 3D geometric shape in the real world scale for each proposal. It enables us to perform inference on the 3D world.

Camera calibration: We cluster line segments to find three vanishing points whose corresponding dimensions are orthogonal to each other [HHF09]. The vanishing points are then used to determine the intrinsic and extrinsic calibration parameters [CRZ00, HZ04b]. We assume that the aspect ratio is 1 and there is no skew. Any pair of finite vanishing points can be used to estimate the focal length. If all three vanishing points are visible and finite in the same image, then the optical center can be estimated as the orthocenter of the triangle formed by the three vanishing points. Otherwise, we set the optical center to the center of an image. Once the focal length and optical center has been determined, the camera rotational matrix can be estimated accordingly [HZ04b].

3D reconstruction. We now present how to back-project a 2D structure to the 3D space and how to derive the corresponding coordinates. Considering a 2D point p in an image, there is a collection of 3D points that can be projected to the same 2D point p. This collection of 3D points lays on a ray from the camera center $C = (Cx, Cy, Cz)^T$ to the pixel $p = (x, y, 1)^T$. The ray $P(\lambda)$ is defined by $(X, Y, Z)^T = C + \lambda R^{-1}K^{-1}p$, where λ is the positive scaling factor that indicates the position of the 3D point on the ray. Therefore, the 3D position of the pixel lies at the intersection of the ray and a plane (the object surface). We assume a camera is 4.5ft high. By knowing the distance and the normal of the floor plane, we can recover the 3D position for each pixel with the math discussed above. Any other plane contacting the floor can be inferred by its contact point with the floor. Then we can gradually recover the whole scene by repeating the process from the bottom up. If there is any object too close to the camera to see the bottom, we will put it 3 feet away from the camera.

3.3.3 The Top-down and Bottom-up MCMC Inference

We design a four-step MCMC algorithm that enables a Markov chain travel up and down through the FGA hierarchy. In each iteration, the algorithm proposes a new parse tree pt^* based on the current one pt according to the proposal probability.

We design jump and diffusion methods to ensure the ergodicity of the Markov Chain. There are two kinds of functional jump proposals: add and delete. The functional jump proposals change the dimensionality of the parse tree. Two kinds of geometric diffusion proposals: α diffusion, β diffusion, and γ diffusion. The α diffusion: data-driven bottom-up detection that directly draws cuboid proposals from a non-parametric distribution built up by the line segments detected from the image; β diffusion: grammar-driven bottom-up prediction that proposes cuboid for a parent node in the parse tree from the children nodes by inversely computing a geometric transformation; γ diffusion: grammar-driven top-down prediction that proposes cuboid by top-down sampling for a child node in the parse tree from its parent node based on the geometric model.

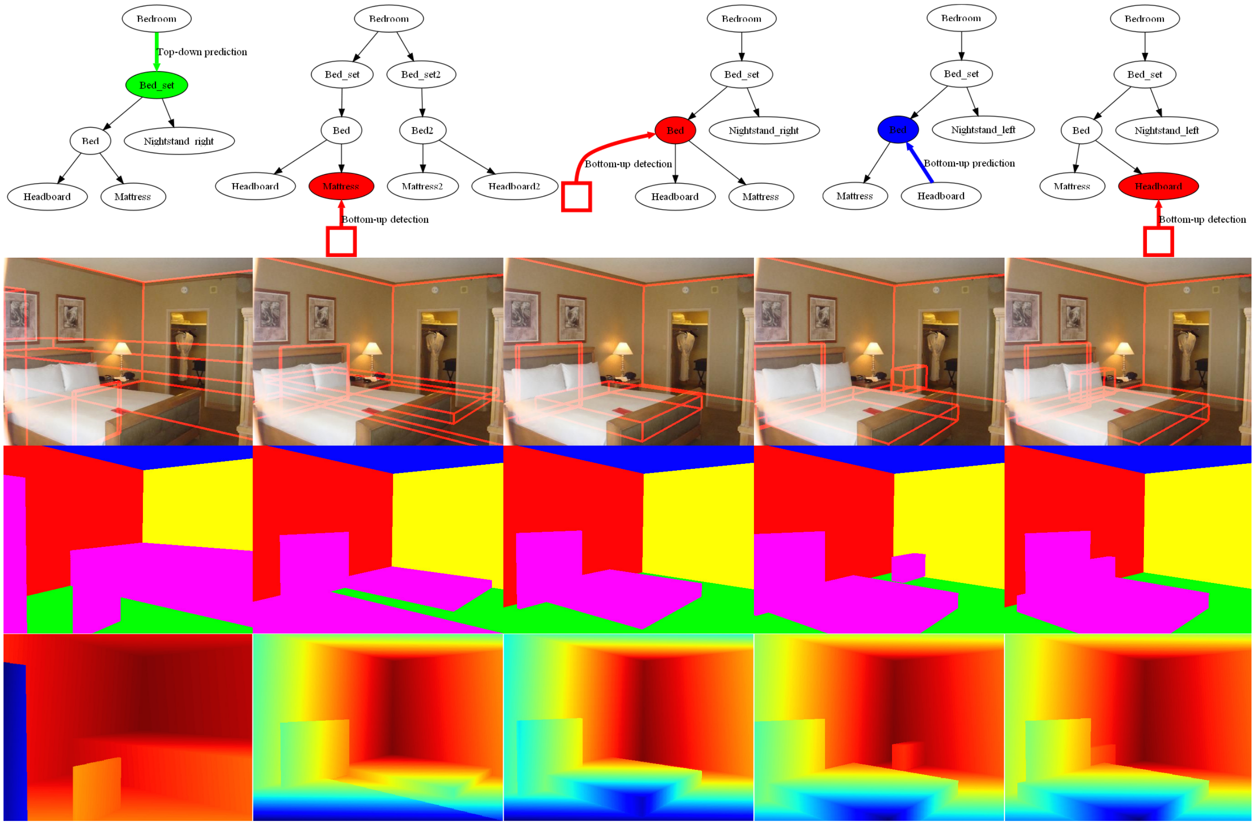


Figure 3.12: Qualitative results of bottom-up top-down Inference. These pictures are overlaid images, label maps, depth map and their corresponding parse trees for 250, 500, 750, 1000, 1250 accepted moves. In particular, the red, blue, and green arrows on the parse trees represent proposals from bottom-up detection, bottom-up prediction, and top-down prediction respectively.

3.3.4 The Functional Jump Proposal

This step re-assigns functional number variables. The switching of functional labels can be happened in any layers of the functional parse tree as shown in Fig. 3.3, and number variables

3.3.4.1 The add proposal

The add proposal samples a subtree pt_v from a non-terminal node $v \in V^N$ randomly chosen from the current parse tree;

$$Q_+(pt \to pt^*) = p(v \in pt)p(pt(v)) \tag{3.11}$$

The proposal first chooses a node $v \in pt$ in grammar randomly. The $p(pt(v))$ is a recursive derivation of production rules from node v. $\prod_{\alpha_0 = v} P(\alpha_i \to \beta_i)$

3.3.4.2 The delete proposal

The detect proposal removes a subtree whose root $v \in pt$ is a node randomly chosen from the current parse tree.

$$Q_-(pt \to pt^*) = p(v \in pt) \tag{3.12}$$

Similarly the delete proposal is calculated by choosing a node v from pt, which is discrete uniform distribution.

Both add proposals and delete proposals essentially change the dimensionality of the parse tree. In order to simplify the Jacobian in Eq. 3.10 of RJMCMC, we designed these jumps Δpt as independently samples from pt so that the Jacobian is 1 in this case ([YYW12]).

3.3.5 The Geometrical Diffusion Proposal

We also defined three kinds of geometric diffusions.

3.3.5.1 α bottom-up detection proposal

As mentioned in the initialization step, we group the line segments to reconstruct 3D cuboid proposals. Each cuboid proposal is assigned with a weight indicating the local likelihood of this proposal. We further process the cuboid proposals by building a non-parametric distribution of the cuboid proposals. The non-parametric distribution is approximated by a weighted KDE (kernel density estimation). Since different objects have different distributions of sizes, we filter all cuboid proposals by the sizes of different objects and combine the score with the original weights, to generate different cuboid distributions for specific objects.

$$Q_{\alpha}(pt \rightarrow pt^{*}) = p(v \in pt)p_{KDE}(G(v)|I_{obs}) \tag{3.13}$$

The $p_{KDE}(G(v)|I)$ is a nonparametric probability distribution estimated by Kernel Density Estimation (KDE) of detected object proposals

$$\begin{aligned} p_{KDE}(x|I_{obs}) &= \sum_{i=1}^{n} w_i K_h(x - x_i) \\ &= \frac{1}{\sum_{i=1}^{n} P(x_i|I_{obs})} \sum_{i=1}^{n} P(x_i|I_{obs}) K_h(x - x_i) \end{aligned} \tag{3.14}$$

where $x_i, i \in 1 \cdots n$ are geometric entities detected from Sect. 3.3.2, and the KDE estimates the non-parametric distribution by considering the local marginal likelihood of each geometric entities $P(x_i|I_{obs})$. h is the window parameter of the kernel $K_h(\cdot)$.

3.3.5.2 β:: bottom-up prediction proposal

The bottom-up prediction refine a higher-level structure $par(v)$ of an existing child v, such as proposing the geometry of a bed set given the geometry of a bed.

$$Q_{\beta}(pt \rightarrow pt^{*}) = p(v \in pt)p(Pos(v))p(Ori(v))p(Size(par(v))) \tag{3.15}$$

This proposal calculate a node's parent by re-sampling the relative position of the child $Pos(v)$ and relative orientation of the child $Ori(v)$ with respect to its parents's coordinate system. And the result coordinates of the parent is calculated by the inverse transformation of Eq. 3.5: $X' = (H_2 H_1)^{-1} X$. The size of the parent node Size(par(v)) is then sampled independently.

3.3.5.3 γ: Top-down Prediction Proposal

The top-down prediction, from another hand, refine a lower-level structure. This is very useful for the heavily occluded object in a functional group. For example, once a bed is correctly detected, this proposal will try to re-allocate nightstands beside the bed by drawing samples from the geometric distribution. Fig. 3.8 shows some typical samples from top-down prediction.

$$Q_\gamma(pt \to pt^*) = p(v \in N^T)p(G(v)) \tag{3.16}$$

Similar to the Eq. 3.15, the algorithm samples the geometric attributes of the node $G(v)$ and estimate the geometric transformation accordingly, thus propose the new geometry of an object.

Here, we can see that geometric diffusions $Q_\alpha, Q_\beta, Q_\gamma$ proposes pt^* from three major channels. The three bottom-up top-down channels are studied by [WZ11]. The geometric parsing is the main challenge in this work, the space of geometric parameters are huge. So most of the MCMC steps are deal with geometric moves. As shown in Fig. 3.11, the $Q_\alpha, Q_\beta, Q_\gamma$ are three kinds of approximation of the marginal distribution $p(v|pt)$ for a node v. The plot on the right panel of Fig. 3.11 shows the average energy convergence of hundreds of Markov Chains in the test dataset using different proposal strategies: By only using the α diffusion from bottom-up detection (red curve), the Markov chain converges very fast at the beginning, but cannot keep reducing the energy due to limitation of bottom-up detections. Using the β diffusion from bottom-up prediction (blue curve) is the worst strategy, because if the terminal node can not be optimized, the prediction from bottom-up can be very bad. The black curve which combine three diffusions together is the best strategy, it has sufficient exploration at the beginning, and gradually converges to the lowest energy. Besides that, the combination of $\alpha\&\beta$ (magenta curve) or $\alpha\&\gamma$ (yellow curve) achieve good results which very close to the black one.

3.4 Experiments

We evaluate our algorithm on two public datasets: the UIUC indoor dataset by [HHF09] and the UCB dataset by [DGB11]. The UCB dataset contains 340 images and covers four cubic objects (bed, cabinet, table and sofa) and three planar objects (picture, window and door). The ground-

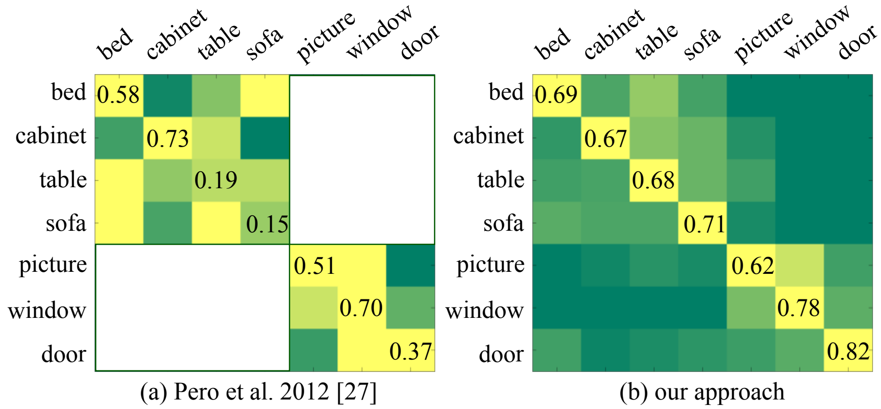


Figure 3.13: The confusion matrix of functional object classification on the UCB dataset.

truths are provided with hand labeled segments for geometric primitives. The UIUC indoor dataset contains 314 cluttered indoor images and the ground-truth is two label maps of the background layout with/without foreground objects.

The functional part of our model is trained with the “bedroom” category (2119 images) and the “living room” category (2385 images) of SUN dataset by [XHE10]. In particular, the branching probability of the number variable for each class is calculated by frequency of each production. The geometric part of our model is trained with CAD data in Fig. 3.6 collected from Trimble 3D Warehouse as discussed in Sect. 3.2.2. We estimated the mixture of Gaussian model by EM clustering, and we manually picked a few typical samples as the initial mean for the Gaussian. And the appearance part of our model is trained on the UIUC dataset as [HHF09]. The weighting parameters of these three components are tuning by cross validation on the training set of UIUC dataset.

Quantitative evaluation:

We first compared the confusion matrix of functional object classification rates among the successfully detected objects on the UCB dataset as shown in Fig. 3.13. The state-of-the-art work by [DBF12] performed slightly better on the cabinet category, but our method get better performance on the table and sofa categories. This is mainly attributed to our fine-grained part model and functional groups model. It is worth noting that our method reduced the confusion between the bed and the sofa. Because we also introduced the hidden variables of scene categories, which help to

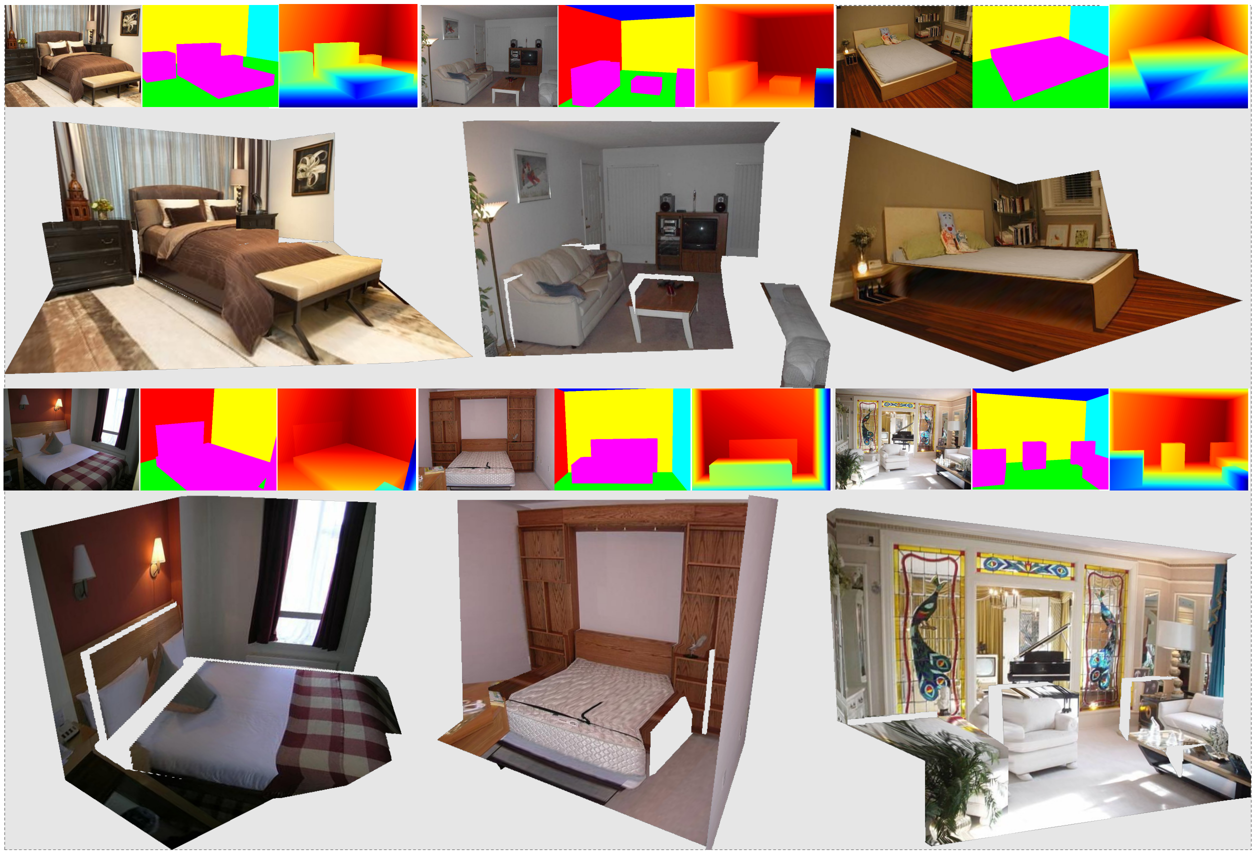


Figure 3.14: 3D reconstruction results based on 3d image parsing. For each image, we show an original image, a segmentation map, a recovered depth image, and a reconstruction result respectively.

distinguish between the bedroom and living room according to the objects inside.

In Table. 3.1, we compared the precision and recall of functional object detection with [DBF12]. The result shows our top-down process did not help the detection of planner objects. But it largely improves the accuracy of cubic object detection from 30.8% to 34.8% with the recall from 24.3% to 29.7%.

In Table. 3.2, we also test our algorithm on the UCB dataset and the UIUC dataset together with five state-of-the-art algorithms: [HHF09], [WGK10], [LGH10], [DGB11] and [DBF12]. The results show the pixel-level segmentation accuracy of proposed algorithms not only significantly widens the scope of indoor scene parsing algorithm from the segmentation and 3D recovery to the functional object recognition, but also yields improved overall performance.

Table 3.1: The precision (and recall) of functional object detection on the UCB dataset.

UCB dataset	planar objects	cubic objects
[DBF12]	27.7% (19.7%)	31.0% (20.1%)
Ours w/o top-down	28.1%(18.5%)	30.8% (24.3%)
Ours w/ top-down	28.1%(18.7%)	34.8% (29.7%)

Table 3.2: The pixel classification accuracy of background layout segmentation on the UCB dataset and the UIUC dataset.

	UCB	UIUC
[HHF09]	-	78.8%
[WGK10]	-	79.9%
[LGH10]	-	83.8%
[SU12]	-	83.54%
[DGB11]	76.0%	73.2%
[DBF12]	81.6%	83.7%
Our approach	82.8%	85.5%

Qualitative evaluation:

Some experimental results on the UIUC and the SUN datasets are illustrated in Fig. 3.15. The green cuboids are cubic objects proposed by the bottom-up AG step, and the cyan cuboids are the cubic objects proposed by the top-down FG step. The blue rectangles are the detected planar objects, and the red boxes are the background layouts. The functional labels are given to the right of each image. Our method has detected most of the indoor objects, and recovered their functional labels very well. The top-down predictions are very useful to detect highly occluded nightstands as well as the headboards of the beds. As shown in the last row, our method sometimes failed to detect certain objects. The bottom left image fails to identify the drawer in the left but a door. In the middle bottom image, the algorithm failed to accurately locate the mattress for this bed with a curtain. The last image is a kind of typical failure example due to the unusual camera position. We assumed the camera position is 4.5 feet high, while this camera position in this image is higher than our assumptions. As a result, the algorithm detected a much larger bed instead.

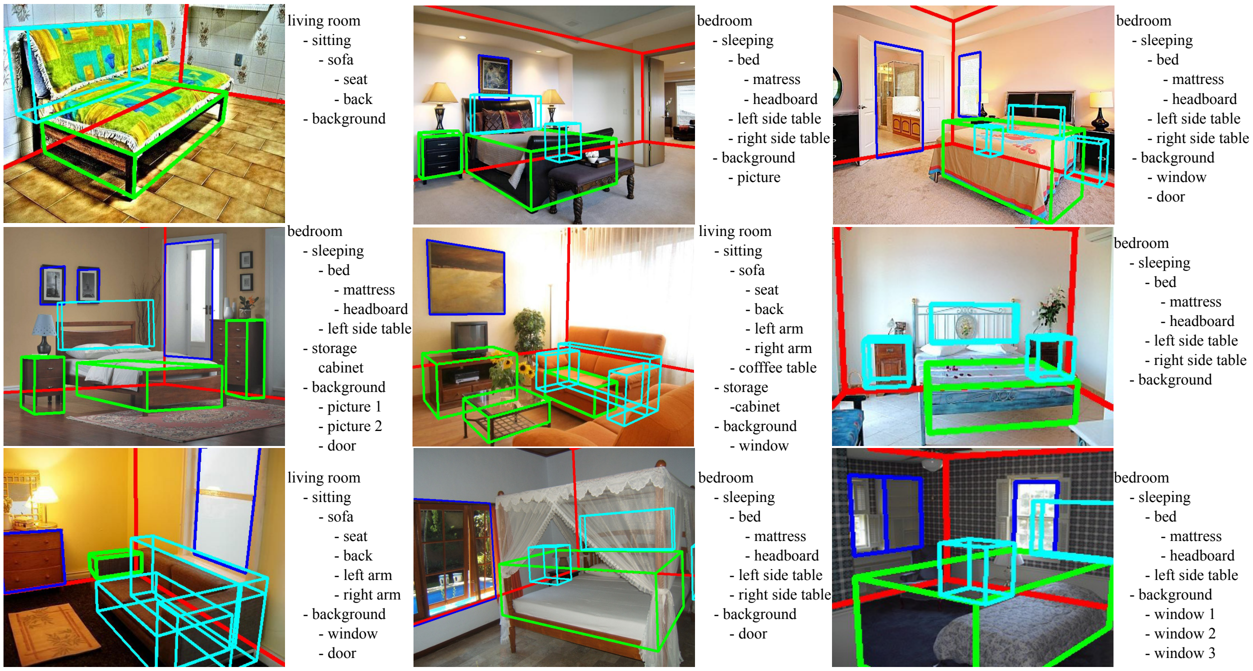


Figure 3.15: Parsing results include cubic objects (green cuboids are detected by bottom-up step, and cyan cuboids are detected by top-down prediction), planar objects (blue rectangles), background layout (red box). The parse tree is shown to the right of each image.

As shown in Fig. 3.11, the algorithm usually converges after three thousand accepted moves. The computational cost of parsing an image in the dataset is around 5-10 minutes. The computational cost varies in terms of the geometric complexity of the image. Usually, the algorithm takes more time to converge if there are more line segments detected.

3.5 Summary

This paper presents a stochastic scene grammar in a function-geometry-appearance (FGA) hierarchy. Our approach parses an indoor image by inferring the object function and the 3D geometry from 2D appearance. The functionality defines an indoor object by evaluating its "affordance". The affordance measures how likely an object can support the corresponding human actions. We found it is effective to recognize certain object functions according to its 3D geometry without observing the actions.

Functionality helps to build a bridge between man-made objects and the human actions, which can motivate other interesting studies in the future: functional objects/areas in a scene attract human's needs and/or intentions; reasoning scene physics and stability in a way similar to [**?**, ZZY15] but from 2D single image instead of RGBD data. As a result, a parsed scene with functional labels defines a human action space, and it also helps to predict people's behavior by making use of the function cues. Furthermore, given an observed action sequence in video, one can recognize the functional objects associated with the rational actions detected from motion.

CHAPTER 4

3D Scene Understanding by Reasoning Physical Stability and Safety

Figure 4.1: A safety-aware robot can be used to detect potentially physically unstable objects in a variety of situations: (a) falling objects at a constructions site, (b) the human assistant for baby proofing, and (c) the disaster rescue (from the recent DARPA Robotics Challenge), where the Multi-Arm robot needs to understand the physical relationships among the debris.

4.1 Introduction

4.1.1 Motivation and Objectives

Traditional approaches, *e.g.*, [SF83, TCY05], for scene understanding have mostly focused on segmentation and object recognition from 2D/3D images. Such representations lack important physical knowledge, such as the stability of the objects, potential physical safety, and supporting relations, which are critical for scene understanding, situation awareness and especially robot vision. The scenarios illustrate the importance of physical knowledge.

i) *Stability and safety understanding*. Our approach utilizes a simple observation that, by human design, objects in static scenes should be stable in the gravity field and be safe with respect to various physical disturbances such as human activities. This assumption poses useful constraints for the plausible interpretations (parses) in scene understanding.

ii) *Human assistant robots*. Objects have the risk to fall onto / hit people at workplaces, such as the construction site in Fig.4.1 (a). To prevent objects from falling freely from one level to another, safety surveillance ensures that people will not get hurt by their environment, especially for children, elders and people with disabilities. As the example shows in Fig.4.1 (b), we see a child is reaching for a teapot, and we can predict possible consequences of his action - the teapot may fall, and the child may get hurt by the falling teapot.

iii) *Disaster rescue robots*. Fig.4.1 (c) shows a HDR-IAI Multi-Arm robot rescuing people during a mock disaster of the DARPA robot challenge [DAR14]. Before planning how to rescue people, the robot needs to understand the physical information, such as which wood block is unsafe or unstable, and the support relations between them.

4.1.2 Overview

In this paper, we present an approach for scene segmentation and potential falling object detection by reasoning physical stability and safety respectively. We define physical stability and safety in a similar form but they are under two different assumptions:

1) **Stability assumption**: objects in the static scene should be stable with respect to the gravity

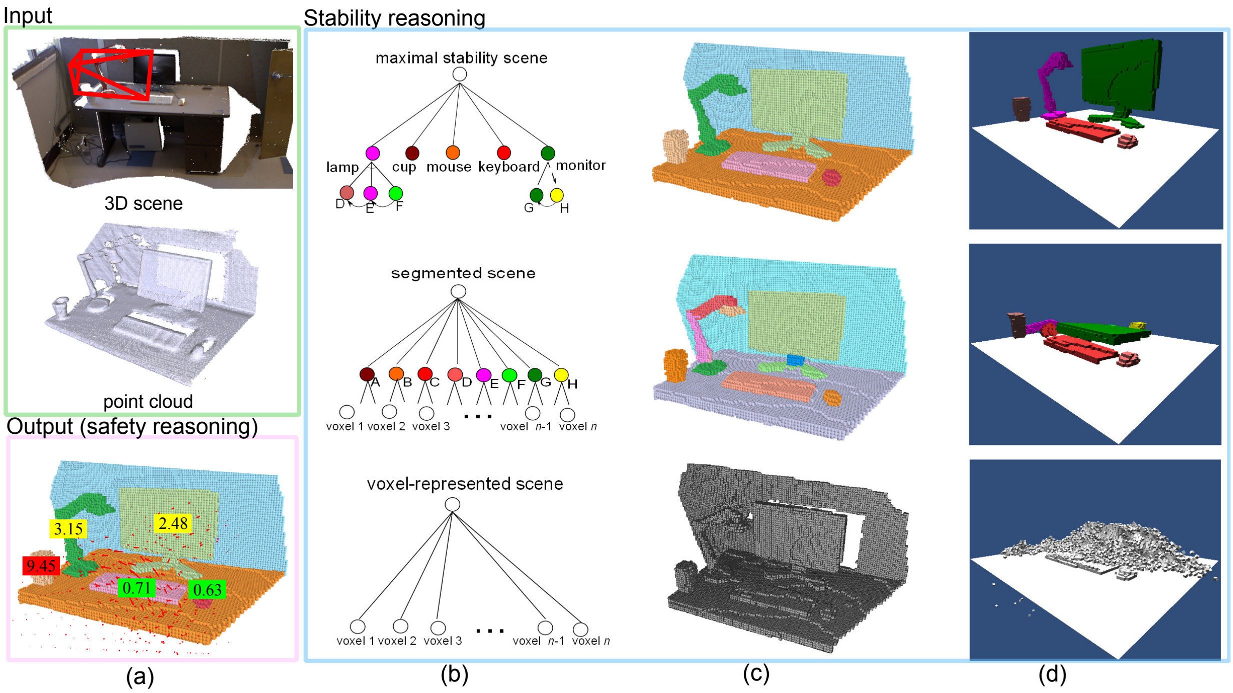


Figure 4.2: Overview of our method. (a) Input: 3D scene reconstructed by SLAM technique and Output: parsed 3D scene as stable objects with supporting relations. The number are risk scores for each object under the disturbance field (in red arrows), (b) scene parsing graphs corresponding to 3 bottom-up processes: voxel based representation (bottom), geometric preprocess including segmentation and volumetric completion (middle), and stability optimization (top). (c) result at each step. (d) physical simulation result of each step.

field. In other words, if any object is not stable on its own, it must be either grouped with neighbors or fixed to its supporting base. Therefore, in the first task, we pursue a physically stable scene understanding, namely “a parse tree”, by inferring object stability in the physical world.

2) **Safety assumption**: Even an object is stable with gravity, but it may not be safe with human activities or other disturbances, e.g., wind or earthquake. Such safety analysis ability is important for robotics applications, such as security surveillance and disaster rescue.

These assumptions are applicable to general scene categories thus pose powerful constraints for physically plausible scene interpretation and understanding.

Our method consists of three main components: geometric reasoning, physical stability reasoning and safety reasoning.

1) **Geometric reasoning**: Given the input point cloud, this step recovers solid 3D volumetric primitives for future physical reasoning. Firstly we segment and fit the input 2.5D depth map or point cloud to small simple (*e.g.*, planar) surfaces; secondly, we merge convexly connected segments into shape primitives; and thirdly, we construct 3D volumetric shape primitives by filling the missing (occluded) voxels, so that each shape primitive has physical properties: volume, mass and supporting areas to allow the computation of the potential energies in the scene.

2) **Physical stability reasoning**: Given a solid 3D volumetric primitives, this step groups primitives into physically stable objects by optimizing the stability and the scene prior. We build a contact graph for the neighborhood relations of the primitives. For example, as shown in Fig.4.2(c) in the second row, the lamp on the desk originally was divided into 3 primitives and would fall under gravity (see result simulated using a physical simulation engine in Fig. 4.2(d)), but becomes stable when they are group into one object – the lamp. So is the computer screen grouped with its base.

3) **Safety reasoning** – Given a static scene consisting of stable objects, this step first infers hidden and situated causes (disturbance field, red arrows in Fig. 4.2(a)) of the scene, and then introduces intuitive physical mechanics to predict the risk scores (e.g., falls) as the consequences of the causes. As shown in Fig. 4.2(a) Output), since the cup is unsafe (falls off the table) under the act of the disturbance field, it gets a high risk score and a red label.

Our method adopts a novel intuitive physics model based on an energy landscape representation using disconnectivity graph (DG). Based on the energy landscape, it defines the physical stability function explicitly by studying the minimum energy (physical work) needed to change the pose and position of an object from one equilibrium to another, and thus release potential energy. For optimizing the scene stabilities, we propose to construct a contact graph and adopt the cluster sampling method, Swendsen-Wang Cut, introduced in image segmentation [BZ05]. The algorithm groups/partitions the contact graph into groups, each being a stable object.

In order to detect unsafe objects in a static scene, our method first infers the “cause” - disturbance field, such as human activities or natural effects. To model the field of human disturbance, we collect the motion capture data of human actions, and apply it to the 3D scene (walkable areas) to estimate the statistical distribution of human disturbance. In order to generate a meaningful human action field, we first predict primary motions on the 2D ground plane which recodes the visiting frequency and walking direction for each walkable position, and add detailed secondary body part motions in 3D space. In addition, we explore two natural disturbances: wind and earthquakes. We then reason the ”effects” (*e.g.*, falling) of each possible disturbance by our intuitive physics model. In this case, we calculate the minimum kinetic energy to move an entity from one stable point to a local maximum, *i.e.*knocking it off equilibrium, and then we further evaluate the risk by calculating the energy released in reaching a deeper minimum. That is, the greater the energy it releases, the higher the risk is.

In experiments, we demonstrate that the algorithms achieve a substantially better performance for i) object segmentation, ii) 3D volumetric recovery of the scene, and iii) scene understanding in comparison to state-of-the-art methods in both public datasets [NF12]. We evaluate the accuracy of potentially unsafe object detection by ranking the falling risk w.r.t. human judgements.

4.1.3 Related Work

Our work is related to 6 research streams in the vision and robotics literature.

1. Geometric segmentation and grouping. Our approach for geometric pre-processing is related to a set of segmentation methods, *e.g.*, [FH04, JKJ11, AFS06, PVB08]. Most of the existing methods are focused on classifying point clouds for object category recognition, not for 3D volumetric completion. For work in 3D geometric reasoning, [AFS06] extracts 3D geometric primitives (planes or cylinders) from a 3D mesh. In comparison, our method is more faithful to the original geometric shape of object in the point cloud data. There has also been interesting work in constructing 3D scene layouts from 2D images for indoor scenes, such as [ZZ11, LHK09, LGH10, HHF10]. [FCS09] also performed volumetric reasoning with the Manhattan-world assumption on the problem of multi-view stereo. In comparison, our volumetric reasoning is based on complex point cloud data and provides more accurate 3D physical properties, *e.g.*, masses, gravity potentials, contact area,*etc*.

2. Physical reasoning. The vision communities have studied the physical properties based on a single image for the ”block world” in the past three decades [BMR82, GEH10a, GSE11, ZZ11, LHK09, LGH10]). *e.g.*Biederman *et al.* [BMR82] studied human sensitivity of objects that violate certain physical relations. Our goal of inferring physical relations is most closely related to [GEH10a] who infer volumetric shapes, occlusion, and support relations in outdoor scenes inspired by physical reasoning from a 2D image, and Silberman *et al.* [NF12, JKS13, GH13] who infers the support relations between objects from a single depth image using supervised learning with many prior features. In contrast, our work is the first that defines explicitly the mathematical model for object stability. Without a supervised learning process, our method is able to infer the 3D objects with maximum stability.

3. Intuitive physics model. The intuitive physics model is an important perspective for human-level complex scene understanding. However, to our best knowledge, there is little work that mathematically defines intuitive physics models for real scene understanding. [JGS13] adopts an intuitive physics model in [McC83], however this model lacks deep consideration on complex physical relations. In our recent work [ZZY13, ZZY14], we propose a novel intuitive physics model based on gravity potential energy transfer. In this paper, we extend this intuitive physics model by combining specific physical disturbance fields. While Physics engines in graphics can accurately simulate the motion of objects under the influence of gravity, it is computationally too expensive for the purpose of measuring object stability.

4. Safe Motion Planning. As motion planning is a classic problem in robotics, [PF05, PL11] tackled the problem of safe motion planning in the presence of moving obstacles. They consider the moving obstacles as a real-time constraint inherent to the dynamic environment. We first argue that a robot needs to be aware of potential dangers even in a static environment due to possible incoming disturbances.

5. Human in the loop. This stream of research emphasizes a human-centric representation, differing from the classic feature-classifier paradigm of object recognition. Some recent work utilized the notion of ”affordance”. [GGG11] recognized chairs by hallucinating a ”sitting” actor interacting with the scene. [GSE11] predicted the ”workspace” of a human given an estimated 3D scene geometry. [FDG12] and [DFL12] demonstrated that observing people performing different actions can significantly improve estimates of scene geometry and scene semantics. [JKS13] and [JS13] proposed scene labeling algorithms by considering humans as the hidden context.

6. Cognitive studies. Recent psychology studies suggested that approximate Newtonian principles underlie human judgements about dynamics and stability [FBB10, HBT11] Hamrick *et al.* [HBT11] showed that knowledge of Newtonian principles and probabilistic representations are generally applied for human physical reasoning. These intuitive models are studied for understanding human behaviors, not for vision robotics.

Recently many semantic labeling methods for 3D point clouds play an important role in robotics: [KAJ11, AKJ12, WLS14], *etc*; in graphics: *e.g.*, [NXS12, SXZ12, SMZ14, SCH14], *etc*; in 3D shape recognition: [KMF13], *etc*. In this paper, however we only focus on the stability and safety reasoning and show its influence on scene understanding.

4.1.4 Contributions

This paper makes the following contributions.

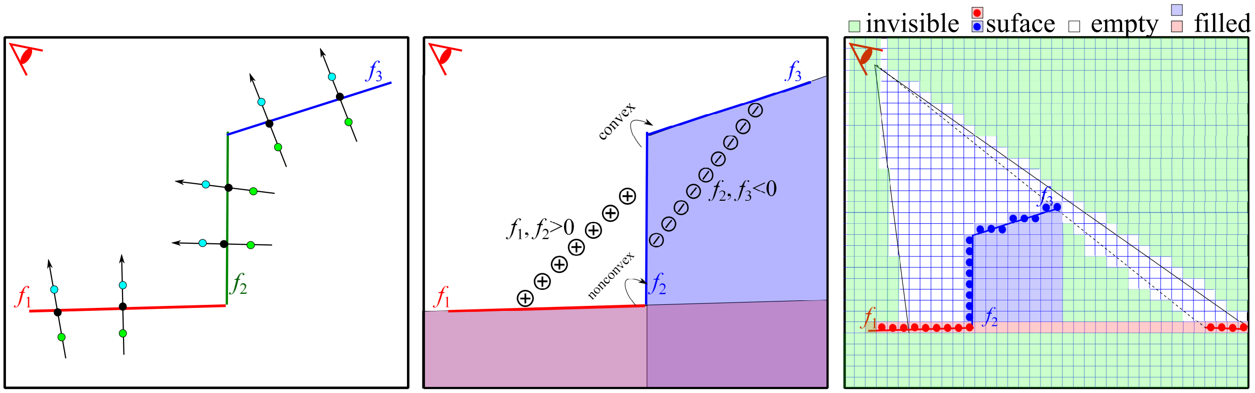
1. It defines the physical stability function explicitly by studying minimum forces and thus physical work needed to change the pose and position of an primitive (or object) from one equilibrium to another, and thus to release potential energy.

2. It introduces a novel disconnectivity graph (DG) from physics [Wal04] to represent the energy landscapes of objects.

3. It solves the complex optimization problem by applying the cluster sampling method Swendsen-Wang cut used in image segmentation [BZ05] to physical reasoning.

4. It collects a new dataset for large scenes using depth sensors for scene understanding and the data and annotations will be released to the public.

The rest of this paper is organized as: Section 2 presents our geometric preprocessing method that first forms solid object primitives from raw point clouds; then the method for reasoning the maximal stability for a static scene is described in Section 3; and reasoning the safety for each object in the scene is presented in Section 4 followed by experimental results and discussions in



(a) (b) (c)

Figure 4.3: (a) Splitting. Two 1-degree IAMs f_1, f_2 and f_3 (in red, green and blue lines respectively) are fitted to the 3-Layer point cloud. Points in green and blue are the extra layer points generated from original points in black. (b) Merging. the segments fitted by f_2 and f_3 are merged together, because they are convexly connected. The convexity can be detected by drawing a line (in circular points) between any two connected segments and checking if their function values are negative. (c) Volumetric completion. Four types of voxels are estimated in volumetric space: invisible voxels (light green), empty voxels (white), surface voxels (red and blue dots), and the voxels filled in the invisible space (colored square in light red or blue).

Sections 5 and 6 respectively.

4.2 Geometric Reasoning: Computing Solid Volumes from Point Clouds

In order to infer the physical properties (*e.g.*, mass, gravity potential energy, supporting area) of objects from point clouds, we first compute a 3D volumetric representation for each object part. We proceed in two steps: 1) point cloud segmentation, and 2) volumetric completion.

4.2.1 Segmentation with Implicit Algebraic Models

We adopt a segmentation method using implicit algebraic models (IAMs) [BLC00] which fits IAMs to point clouds with simple geometry.

$$f_i(\mathbf{p}) \approx 0, \tag{4.1}$$

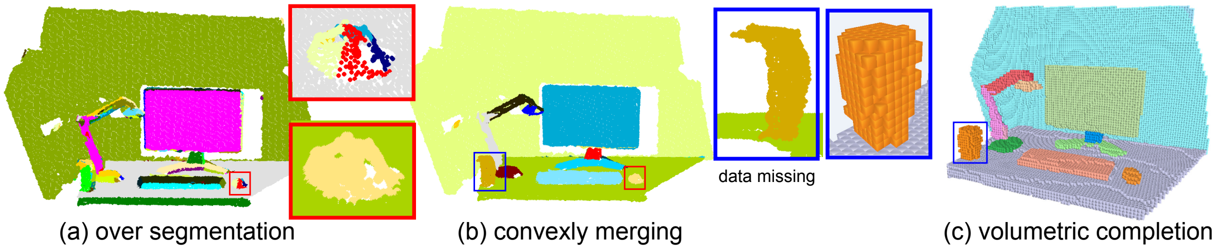


Figure 4.4: Geometric reasoning process: (a) Over-segmentation result obtained by splitting with IAMs. (b) Result after merging the convexly connected faces. (see the difference on "mouse" object). (c) Result after volumetric completion. (see the difference on "cup" object and hole on the back wall).

where $\mathbf{p} = \{x, y, z\}$ is a 3D point and f_i is defined by an n-degree polynomial:

$$f_i(\mathbf{p}) = \sum_{0 \le i,j,k; i+j+k \le n} a_{ijk} x^i y^j z^k, \tag{4.2}$$

where a_{ijk} are the unknown coefficients of the polynomial. The main advantage of IAM is that it is convenient for accessing the "inside" ($f_i < 0$) or "outside" ($f_i > 0$) of a surface fitted by an IAM.

Our method is in 2 steps as Fig.4.3 (a) and (b) illustrated: 1) splitting step: over-segmenting the point cloud into simple regions approximated by IAMs, and then 2) merging step: merging them together with respect to their convexly connected relations.

4.2.1.1 Splitting Step

The objective in this step can be considered to be finding the maximal 3D regions, each of them well fitted by an IAM.

The IAM fitting for each region is formulated in least squares optimization using the 3-Layer method proposed by [BLC00] As shown in Figure 4.3(a), it first generates two extra point layers along the surface normals. Then, the IAM can be fitted to the point set constrained by 3 layers with linear least squared fitting.

We adopt a region growing scheme [PVB08] in our segmentation. Thus our method can be described as: starting from several given seeds, the regions grow until there is no point that can be merged into the region fitted by an IAM. We adopt the IAM of 1 or 2 degree, *i.e.*, planes or second order algebraic surfaces and use the IAM fitting algorithm proposed by Zheng *et al.* , [ZTI10] to select the models in a degree-increasing manner.

4.2.1.2 Merging Step

The above segmentation method over-segments the objects into pieces. This is still a poor representation for objects, since only the segments viewed as faces of objects are obtained. According to a common observation that an object should be composed of several convex hulls (primitives) whose faces are convexly connected, we propose a merging step that merges the convexly connected segments together to approach the representation of object primitives.

To detect the convex connection, as shown in Fig. 4.3 (b), we first sample the points on a line which connects two adjacent regions (the circle lines in Fig. 4.3 (b)) as: $\{\mathbf{p}_l | \mathbf{p}_l \in L\}$, where L denotes a line segment whose ends are on the two connected regions respectively. To detect the convexly connected relationship, we take a condition as the judgment:

$$\frac{\#\{\mathbf{p}|\mathbf{p}_l \in L \wedge f_i(\mathbf{p}_l) < 0 \wedge f_j(\mathbf{p}_l) < 0\}}{\#\{\mathbf{p}|\mathbf{p}_l \in L\}} > \delta_2, \tag{4.3}$$

where the ratio threshold δ_2 is set as 0.6. As illustrated in Fig 4.3 (b), since the circular points drawn between f_2 and f_3 are negative, the segments should be merged. Fig. 4.4 (a) and (b) shows the difference before and after merging the convexly connected regions.

4.2.2 Volumetric Space Completion

The primitives output from the above method are still insufficient to reason the physical properties, *e.g.*, in Fig. 4.4 (b), the wall and table have hollow surfaces with holes and the cup has missing volume. To overcome this, we first generate a voxel-based representation for the point cloud such that each voxel can be viewed as a small mass unit with its own volume, gravity and contact region (contact faces of the cube). Secondly, we fill out the hidden voxels for each incomplete volumetric primitive obtained by the segmentation result above.

4.2.2.1 Voxel Generation and Gravity Direction

Our voxel based representation is generated by constructing the octree of the point cloud as proposed by Sagawa *et al.* [SNI05], after which the point cloud is regularized into the coordinate system under the Manhattan world assumption [FCS09], supposing many visible surfaces orient along one of three orthogonal directions. To detect gravity direction, 1) we first calculate the distributions of the principal orientations of the 3D scene by clustering the surface normals into K ($K > 3$) clusters; 2) Then we extract three biggest clusters and take their corresponding normals as three main orientations; 3) After the orthogonalization of these three orientations, we choose the one with smallest angle to the Y-axis of camera plane as the gravity direction.

4.2.2.2 Invisible Space Estimation

As light travels in straight lines, the space behind the point clouds and beyond the view angles is not visible from the camera's perspective. However this invisible space is very helpful for completing the missing voxels from occlusions. Inspired by Furukawa's method in [FCS09], the Manhattan space is carved by the point cloud into three parts, as shown in Figure 4.3(c): Object surface $\mathbb{S}$ (colored-dots voxels), Invisible space $\mathbb{U}$ (light green voxels) and Visible space $\mathbb{E}$ (white voxels).

4.2.2.3 Voxels Filling

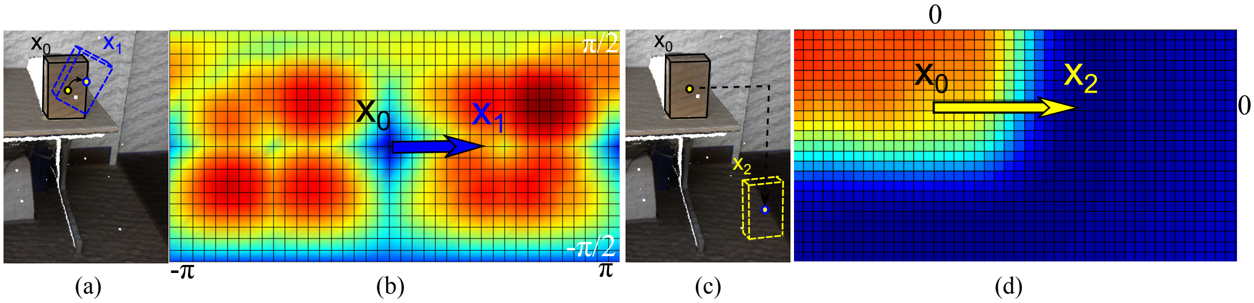
After obtaining labels by the above point cloud segmentation, first each voxel on surface $\mathbb{S}$ inherits the labels from the points that it enclosed. Then the completion of the missing parts for the volumetric primitives can be considered as guessing the label for each voxel which are invisible but should be belong to the object. As Figure 4.3 (b) illustrates, the algorithm can be described as:

Loop: for each invisible voxel $v_i \in \mathbb{U}$, $i = 1, 2, \ldots$

1. Starting from v_i to search the voxels, along 6 directions, until reach a voxel $v_j, j = 1 \ldots, 6$ that $v_j \in \mathbb{S}$. or v_j belongs to boundary of the whole space.
2. Checking the labels of v_js, if there are more than two same labels exist, then assign this label

to current voxel.

Fig. 4.4 (c) shows an example of volumetrically completing the primitives from (b). With the voxel representation, the primitives' mass, center of gravity (CoG) can be efficiently calculated.



(a) (b) (c) (d)

Figure 4.5: An example of the potential energy map determined by pose and position changes: (a) the box on desk changes pose from state x_0 to x_1. Mass center trajectory is shown as black arrow. (b) the energy map of changing box poses in arbitrary directions. State x_0 is at local minimum on the map. (c) the box on desk changes position from state x_0 to x_2; (d) the energy map of changing box position. Due to friction is considered, State x_0 is at local minimum on the map.

4.3 Modeling Physical Stability and Safety

4.3.1 Energy Landscapes

Since any object (or primitive) has potential energy determined by its mass and height to the ground, we can generate its potential energy landscape according to the environment where it stays.

The object is said to be *in equilibrium* when its current state is a local minimum (stable) or non-local minimum (unstable) of this potential function (See Fig 4.5 for illustration). This equilibrium can be broken after the object has absorbed external energy, and then the object moves to a new equilibrium and releases energy. Note that if too much uncontrolled energy is released, the object is perceived to be "unsafe", which we will discuss later. Without loss of generality, we divide the change into two cases.

Case I: pose change. In Fig. 4.5 (a), the box on a desk is in a stable equilibrium and its pose is changed with external work to raise its center of mass. We define the energy change needed for the state change $\mathbf{x}_0 \to \mathbf{x}_1$ by

$$E_r(\mathbf{x}_0 \to \mathbf{x}_1) = (R\mathbf{c} - \mathbf{t}_1) \cdot m\mathbf{g}, \tag{4.4}$$

where $\cdot$ denotes inner product, R is rotation matrix; $\mathbf{c}$ is the center of mass, $\mathbf{g} = (0, 0, 1)^T$ is the gravity direction, $\mathbf{t}_1$ is the lowest contact point on the support region (its corners). Suppose the support region is flat, only the rotations of roll and pitch change the object CoM. Thus we can visualize the energy landscape in a spherical coordinate system (ϕ, θ): $S^2 \to \mathbb{R}$ with two pose angles $\{\phi \in [-\pi\ \pi], \theta \in [-\pi/2, \pi/2]\}$. In Fig. 4.5 (b), the blue color means lower energy and red means high energy. Such energy can be computed for any rigid objects by bounding the object with a convex hull. We refer to the early work of Kriegman [Kri95] for further details.

Case II: position change. We consider the position change when object is viewed as a mass point and can move to different position in its environment. For example, as shown in Fig. 4.5 (c), the box on desk at stable equilibrium state $\mathbf{x}_0$, one can push it to the edge of the desk. Then it falls to the ground and releases energy to reach a deeper minimum state $\mathbf{x}_2$. The total energy change need to consider the gravity potentials and the frictions which is overcome by a work absorbed.

$$E_t(\mathbf{x}_0 \to \mathbf{x}_2) = -(\mathbf{c} - \mathbf{t}) \cdot m\mathbf{g} + W_f, \tag{4.5}$$

where $\mathbf{t} \in \mathbb{R}^3$ is the translation parameter (shortest path to the final position $\mathbf{x}_2$), and W_f is the absorbed energy for overcoming the frictions:

$W_f = f_c \cdot mg\sqrt{(t_1 - c_1)^2 + (t_2 - c_2)^2}$ given the friction coefficient f_c. Note for common indoor scenes, we choose f_c as 0.3 as common material such as wood. Therefore the energy landscape can be viewed as a map from 3D space $\mathbb{R}^3 \to \mathbb{R}$.

4.3.2 Disconnectivity Graph Representation

The energy map is continuously defined over the object position and pose. For our purpose, we are only interested in how deep its energy basin is at the current state (according to the current interpretation of the scene). As the interpretation changes during optimization process, the energy landscape for each object will be updated. Therefore, we represent the energy landscape by a so-called disconnectivity graph (DG) which has been used in studying spin-glass models in physics [Wal04]. As Fig. 4.6 illustrates that, in the DG, the vertical lines represent the depth of the energy basins and the horizontal lines connect adjacent basins. The DG can be constructed by an algorithm scanning energy levels from low to high and checking the connectivity of components at each level [Wal04].

From the disconnectivity graph, we can conveniently calculate two quantities: *Energy absorption* and *Energy release* during the state changes.

Definition 1 *The energy absorption $\Delta(\mathbf{x}_0 \to \widetilde{\mathbf{x}})$ is the energy absorbed from the perturbations, which moves the object from the current state $\mathbf{x}_0$ to an unstable equilibrium $\widetilde{\mathbf{x}}$ (say a local maximum or an energy barrier).*

For the box on the desk in Fig.4.5, its energy absorption is the work needed to push it in one direction to an unstable state $\mathbf{x}_1$. For the state $\mathbf{x}_2$, its energy barrier is the work needed (to overcome friction) to push it to the edge. In both cases, the energy depends on the direction and path of movement.

Definition 2 *Energy release $\Delta(\widetilde{\mathbf{x}} \to \mathbf{x}_0')$ is the potential energy released when an object moves from its unstable equilibrium $\widetilde{\mathbf{x}}$ to a minimum $\mathbf{x}_0'$ which is lower but connected by the energy barrier.*

For example, when the box falls off from the edge of the table to the ground, energy is released. The higher the table, the larger the released energy.

4.3.3 Definition of Stability

With DG, we define object stability in 3D space.

Definition 3 *The instability $S(a, \mathbf{x}_0, W)$ of an object a at state $\mathbf{x}_0$ in the presence of a disturbance work W is the maximum energy that it can release when it moves out of the energy barrier by the*

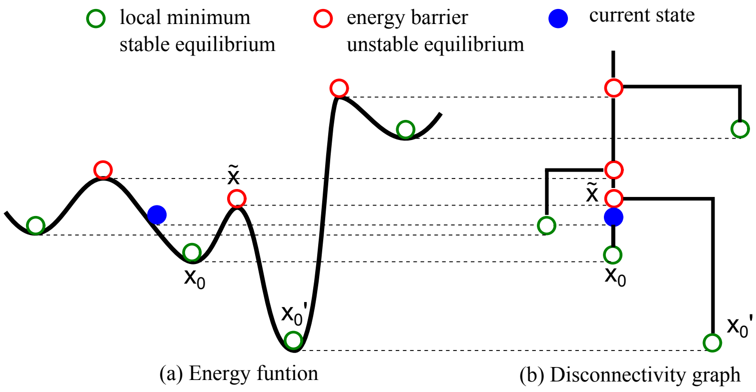


Figure 4.6: (a) Energy landscapes and its corresponding disconnectivity graph (b).

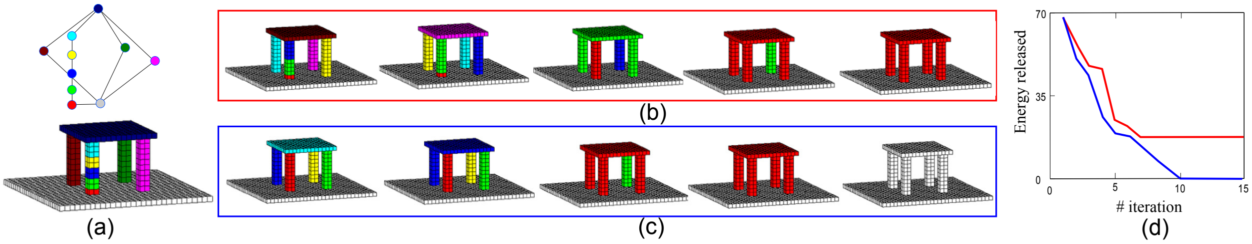


Figure 4.7: Example of illustrating the Swendsen-Wang cut sampling process. (a) Initial state with corresponding contact graph. (b) shows the grouping proposals accepted by SWC at different iterations. (c) convergence under increasingly (from left to right) larger disturbance W and consequently the table is fixed to the ground. (d) shows two curves of Energy released v.s. number of iteration in SWC sampling corresponding to (b) and (c).

external work W.

$$\begin{aligned} &\mathrm{S}(a, \mathbf{x}_0, W) \\ &= \max_{\mathbf{x}_0'} \triangle(\widetilde{\mathbf{x}} \rightarrow \mathbf{x}_0')\delta([\min_{\widetilde{\mathbf{x}}} \triangle(\mathbf{x}_0 \rightarrow \widetilde{\mathbf{x}})] \leq W), \end{aligned} \tag{4.6}$$

where $\delta()$ is an indicator function and $\delta(z) = 1$ if condition z is satisfied, otherwise $\delta(z) = 0$. $\triangle(\mathbf{x}_0 \rightarrow \widetilde{\mathbf{x}})$ is the energy absorbed, if it is overcome by W, then $\delta() = 1$, and thus the energy $\triangle(\widetilde{\mathbf{x}} \rightarrow \mathbf{x}_0')$ is released. We find the easiest direction $\widetilde{\mathbf{x}}$ to minimize the energy barrier and the worst direction $\mathbf{x}_0'$ to maximize the energy release. Intuitively, if $S(a, \mathbf{x}_0, W) > 0$, then the object is said to be unstable at level W disturbance.

4.3.4 Definition of Safety

We measure the safety by supposing a specific disturbance field as potentially existing in the scene, such human activities, winds or earthquakes. This specific disturbance field should have nonuniform and directional energy distribution.

Definition 4 *The risk $R(a, \mathbf{x}_0)$ of an entity a at position $\mathbf{x}_0$ in the presence of a disturbance field $p(W, \mathbf{x})$ is the expected risk with respect to the disturbance distribution.*

$$R(a, \mathbf{x}_0) = \int p(W, \mathbf{x}_0) S(a, \mathbf{x}_0, W) dW, \tag{4.7}$$

For example, it is more unsafe if there exist a disturbance that makes the box in Fig. 4.5 fall off from the desk than just fall down on the desk.

With the definition of the instability and risk, we first present the algorithm for static scene understanding by reasoning the stability in Sec.4, and then we introduce the inference of the disturbance field in Sect.4.5.1 and the calculation of potential energy and initial kinetic energy given a disturbance in Sect.4.5.2

4.4 Stability Reasoning

Given a list of 3D volumetric primitives obtained by our geometric reasoning step, we first construct the contact graph, and then the task of physical reasoning can be posed as a well-known graph labelling or partition problem, through which the unstable primitives can be grouped together and assigned the same label to achieve global stability of the whole scene at a certain disturbance level W.

4.4.1 Contact Graph and Group Labeling

The contact graph is an adjacency graph $G =< V, E >$, where $V = \{v_1, v_2, ..., v_k\}$ is a set of nodes representing the 3D primitives, and E is a set of edges denoting the contact relation between the primitives. An example is shown in Fig.4.7 (a) top where each node corresponds to a primitive in Fig. 4.7 (a) bottom. If a set of nodes $\{v_j\}$ share a same label, that means these primitives are fixed to a single rigid object, denoted by O_i, and their instability is re-calculated according to O_i.

The optimal labeling L^* can be determined by minimizing a global energy function, for a disturbance level W

$$E(L|G;W) = \sum_{O_i \in L} (\mathrm{S}(O_i, \mathbf{x}(O_i), W) - (O_i)), \tag{4.8}$$

where $\mathbf{x}(O_i)$ is the current state of grouped object O_i. The new term F represents a penalty function expressing the scene prior and can be decomposed into three terms.

$$(O_i) = \lambda_1 f_1(O_i) + \lambda_2 f_2(O_i) + \lambda_3 f_3(O_i), \tag{4.9}$$

where f_1 is the total number of voxels in object O_i; f_2 is the geometric complexity of O_i, which can be simply computed as the summation of the difference of normals for any two connected voxels on its surface; and f_3 is the freedom of object movement on its support area. f_3 can be calculated as the ratio between the support plane and the contact area $\frac{\#S}{\#CA}$ of each pair of primitives $\{v_j, v_k \in O_i\}$, where one of them is supported by the other. After they are regularized to the scale of objects, the parameters λ_1, λ_2 and λ_3 are set as 0.1, 0.1, and 0.7 in our experiment. Note, the third penalty is designed from the observation that, *e.g.*, a cup should have freedom of movement supported by a desk, and therefore the penalty arises if the cup is assigned the same label as the desk, as shown in Fig. 4.2. Therefore under the stable conditions, objects should have maximal freedom of movement.

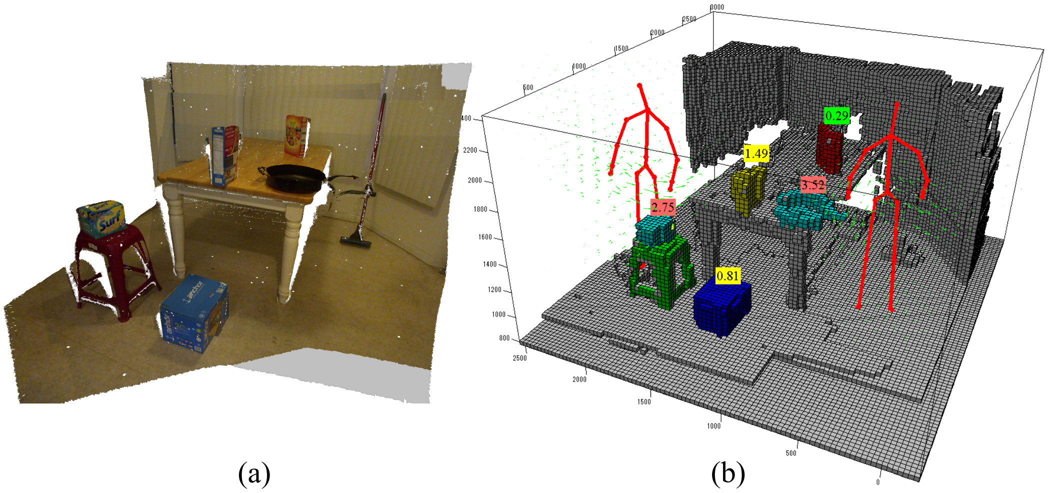
4.4.2 Inference of Maximum Stability

As the labels of primitives are coupled with each other, we adopt the graph partition algorithm Swendsen-Wang Cut (SWC) [BZ05] for efficient MCMC inference. We obtain the globally optimal L^* by maximizing a posterior probability

$$L^* = \arg\max p(L|G;W) = \arg\max 1/Z \exp\{-E(L|G;W)/T\} \tag{4.10}$$

and T is a temperature factor.

The SWC algorithm performs three iterative steps until convergence as described as follows.



(a) (b)

Figure 4.8: (a) The input point cloud; (b) Hallucinated human action field and detected potential falling objects with red tags.

(i) *Edge turn-on probability.* Each edge $e \in E$ is associated with a Bernoulli random variable $\mu_e \in \{\text{on}, \text{off}\}$ indicating whether the edge is turned on or off, and a weight reflecting the possibility of doing so. In this work, for each edge $e =< v_i, v_j >$, we define its turn-on probability as:

$$q_e = p(\mu_e = on | v_i, v_j) = \exp\left(-(F(v_i, v_j)/T\right), \tag{4.11}$$

where T is a temperature factor and $F(\cdot,\cdot)$ denotes the feature between two connected primitives. Here we adopt a feature using the ratio between contact area (plane) and object planes as: $F = \frac{\#CA}{max(\#A_i, \#A_j)}$, where CA is the contact area, A_i and A_j are the areas of v_i and v_j on the same plane of CA.

(ii) *Graph Clustering.* Given the current label map, it removes all edges between nodes with different labels. Then all the remaining edges are turned on independently with probability q_e. Thus, we have a set of connected components (CCPs) Π's, in which all nodes have the same category label.

(iii) *Graph Flipping.* It randomly selects a CCP Π from the set formed in step (ii) with a uniform probability, and then flips the labels of all nodes in Π to a label $l' \in \{1, 2, ..., L\}$. The flip

```
Algorithm 2: SWC Inference
Input: A contact graph $G =< V, E >$ with discriminative edge probabilities $q_e$, $e \in E$
Output: A physically plausible explanation $L^*$
while Not converge do
    for $e \in E$ do
        turn $e =$"on" with probability $q_e$ in Eq. (4.11)
    end
    Collect all the connected components $CPPs = \{\Pi_i : i = 1, \ldots, n\}$
    Randomly select a connected component $\Pi \in CPPs$ and randomly assign a new label
    to it
    Accept the move with probability $\alpha(L \to L')$ in Eq.(4.12)
    Lower the temperature T
end
```

is accepted with probability [BZ05]:

$$\begin{aligned} &\alpha(L \to L') = \\ &\min{(1, \frac{\prod_{e \in (V_0, V_{L'} \setminus V_0)}(1 - q_e)}{\prod_{e \in (V_0, V_L - V_0)}(1 - q_e)} \cdot \frac{p(L'|G;W)}{p(L|G;W)})}, \end{aligned} \tag{4.12}$$

where $p = \frac{1}{z}\exp{(-E)}$. Fig. 4.7 illustrates the process of labeling a number of primitives of a table into a single object. SWC starts with an initial graph in (a), and some of the sampling proposals are accepted by the probability (4.12) shown in (b) and (c), resulting in the energy v.s. iterations in (d). It is worth noticing that i) in case of Fig. 4.7 (b), the little chair is not grouped to floor, since the penalty term A_3 penalizes the legs grouping with the floor; and ii) with increased disturbance W, the chair is fixed to the floor.

We summarize the main three steps above in Algorithm 2. Here we adopt an annealing scheme in the MCMC sampling process, when the temperature is low, the algorithm will converge to a global optimal solution, i.e. partition, with very high probability. Fortunately, the solution space in our algorithm is quite small after geometric processing. For example, there are only 12 geometric entities as graph nodes in the table scene in Fig. 4.4 (c), and the algorithm converges in several seconds.

4.5 Safety Reasoning

While the objects are stable in the gravity field of a static scene after reasoning the stability, they might be unsafe under a potential specific physical disturbance, such as human activities. For example, all the objects shown in Fig.4.8 (a) can be parsed correctly to be stable in the scene, but if the physical disturbance generated from human common activities is applied, the objects show different safety levels.

Our method infers the disturbance field caused by an earthquake or wind, as well as the human action disturbance field. Given the scene geometry and walkable area, we detect the potential falling objects by calculating its expected falling risk given a disturbance field in Fig.4.8 (b).

4.5.1 Safety Under Different Disturbances

4.5.1.1 Natural Disturbance Field

Aside from the gravity applying a constant downward force to all the voxels, other natural disturbances such as earthquakes and winds are also present in a natural scene.

1) **Earthquake** transmits energy by forces of interactions between contacting surfaces, typically by the frictions in our scenes. Here, we estimate the disturbance field by generating random horizontal forces to the voxels along the contacting surfaces. We use a certain constant to simulate the strength of the earthquake and the work W it generates.

2) **Wind** applies fluid forces to exposed voxels in the space. A precise simulation needs to simulate the fluid flow in the space. Here, we simplify it as a uniformly distributed field over the space.

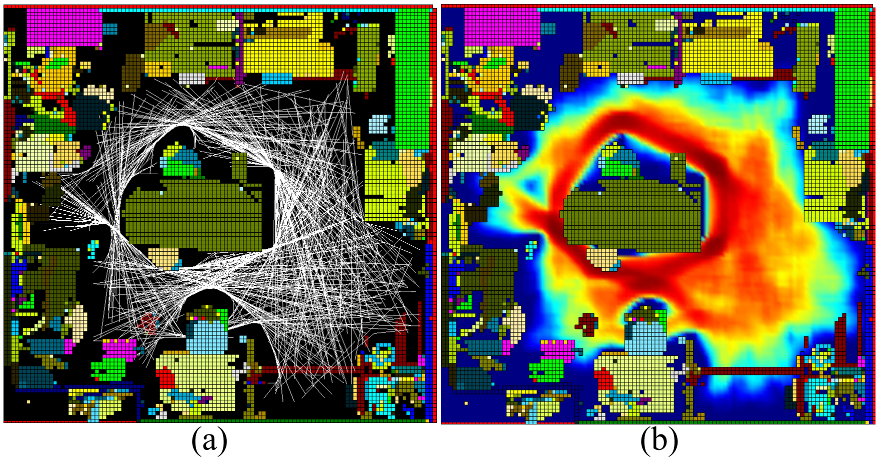


Figure 4.9: Primary motion field: (a) The hallucinated human trajectories (white lines); (b) The distribution of the primary motion space. The red represents high probability to be visited.

4.5.1.2 Human Action Disturbance Field

In order to generate a meaningful disturbance field of human actions, we decompose the human actions into the primary motions *i.e.*the center of mass movements in Fig.4.9 and the secondary motions *i.e.*the body parts' movements in Fig.4.10 We first predict a human primary motion field on the 2D ground plan, and add detailed secondary motions in 3D space on top. The disturbance field is characterized by the moving frequency and moving velocity for each quantized voxel.

The primary motion field captures the movement of human body as a particle. We estimate the distribution of primary human motion space by synthesizing human motion trajectories following two simple observations:

1) A rational agent mostly walks along a shortest path with minimal effort;

2) An agent has a basic need to travel between any two walkable positions in the scene.

Therefore, we randomly pick 500 pairs of positions in the walkable space, we calculate the shortest path connecting these two positions as shown in Fig.4.9 (a), and we calculate the walking frequency as well as walking directions based on the synthesized trajectories. Fig.4.9 (b) demonstrates a distribution of walkable space; the red color means the position has high probability to be visited, and the length of the small arrows shows the probability of moving directions.

In Fig.4.9 (b), we can see that convex corners, e.g. table corners, are more likely to be visited, and objects in these busy area may have higher risk than the ones in concave corners. A hallway connecting two walkable area is also frequently visited, and objects in the hallway are less safe too. Note the distribution of moving directions is also very distinctive. It helps to locate human body movement in the right direction for generating the human disturbance field.

The secondary motion field is the movement that is not part of the main action, for example, arms swinging while walking. The secondary motion is important to capture the random disturbance; for example, people may push objects off the edge of the table by hand or kick objects on the ground by foot. We also the Kinect camera to collect human motion capture data Fig.4.10 (a), and then calculate the distribution of moving velocities as shown in Fig.4.10 (b).

The primary motion field further convolves with secondary motion field, thus generating a dense disturbance field that captures the distribution of motion velocity for each voxel in the space. The disturbance field is then represented by a probability distribution over the entire space for the velocities along different directions and frequencies that they occur. For example, a box in the middle of a large table will not be reachable by a walking person and thus the distribution of velocity above the table center, or any unreachable points, is zero. Five typical cases in the integrated field is demonstrated in Fig.4.11

4.5.2 Calculating the Physical Energy

Given the disturbance field, in this section we present a feasible way for calculating input work (energy) that might lead to an object falling. However, building sophisticated physical engineering models is not feasible, as it becomes intractable if we consider complex object shapes and material properties, *e.g.*, to detect a box falling off from a table, a huge amount of actions need to be simulated until meeting the case of the human body acting on the box. The relation between intuitive physical models and human psychology was discussed by a recent cognitive study [HBT11]

In this paper, for simplicity, we make following assumptions: 1) All the objects in the scene are rigid; 2) All the objects are made from same material, such as wood (friction coefficient: 0.3, uniform density: $700kg/m^3$); and 3) A scene is a dissipative mechanical system such that total

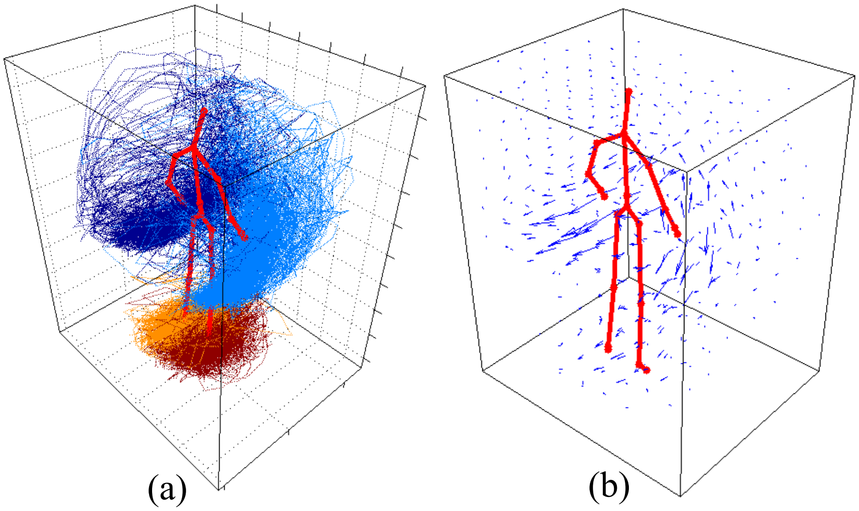


Figure 4.10: Secondary motion field: (a) Secondary motion trajectories from motion capture data; (b) Distribution of the secondary motion field. Long vectors represent large velocity of body movement.

mechanical energy along any trajectory is always decreasing due to friction, while kinetic and potential energy may be traded off at different states due to elastic collision.

Given the human motion distribution with velocity of each body part, we intuitively calculate the kinetic energy of human motion, as the input work. Here, we simplify the parts of body as mass points and at each location in 3D space its kinetic energy can be calculated given the mass of parts. For example, supposing the mass of right hand with upper arm is about 700g, we can simply calculate out the kinetic energy distribution by multiplying half of the velocity squares.

4.6 Experimental Result

We quantitatively evaluate our method in four criteria: i) single depth image segmentation, ii) volumetric completion evaluation, iii) physical inference accuracy evaluation, and iv) safety ratings for objects in scene.

All these evaluations are based on three datasets:

- the NYU depth dataset V2 [NF12] including 1449 RGBD images with manually labeled

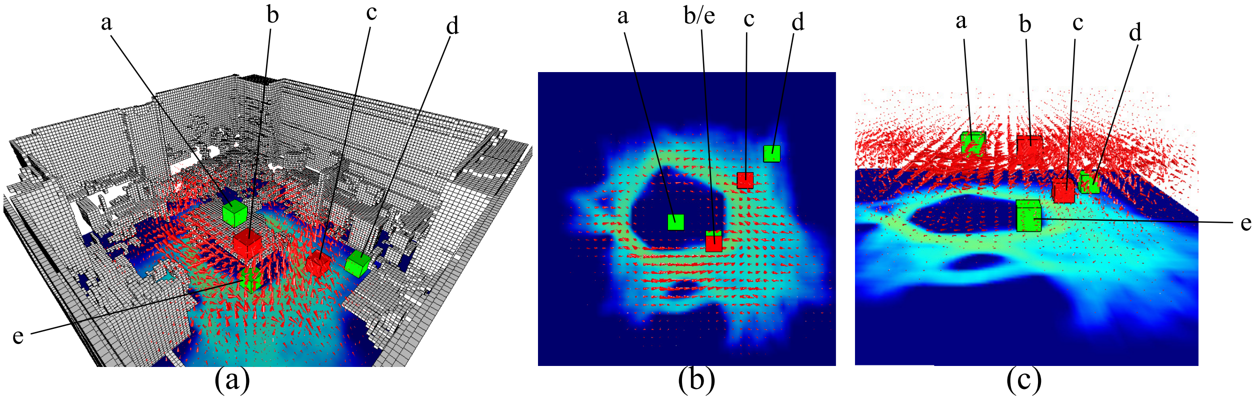


Figure 4.11: The integrated human action field by convolving primary motions with secondary motions. The objects **a-e** are five typical cases in the disturbance field: the object **b** on edge of table and the object **c** along the passway exhibit more disturbances (accidental collisions) than other objects such as **a** in the center of the table, **e** below the table and **d** in a concave corner of space.

	Office	Living room	desk	total
Scenes	5	4	4	13

Table 4.1: Summary of the Dataset. Some samples are shown in Fig. 4.12

ground truth.

- synthesized depth map and volumetric images simulated from CAD scene data.
- 13 reconstructed 3D scene data captured by Kinect Fusion [NIH11] gathered from office and residential rooms with ground truth labeled by a dense mesh coloring.

4.6.1 Evaluating Single Depth Image Segmentation

Two evaluation criteria: “Cut Discrepancy” and “Hamming Distance” mentioned in [CGF09] are adopted. The former measures errors of segment boundaries to ground truth, and the latter measures the consistency of segment interiors to ground truth. As shown in Fig. 4.14, our segmentation by physical reasoning has a lower error rate than the other two: region growing segmenta-

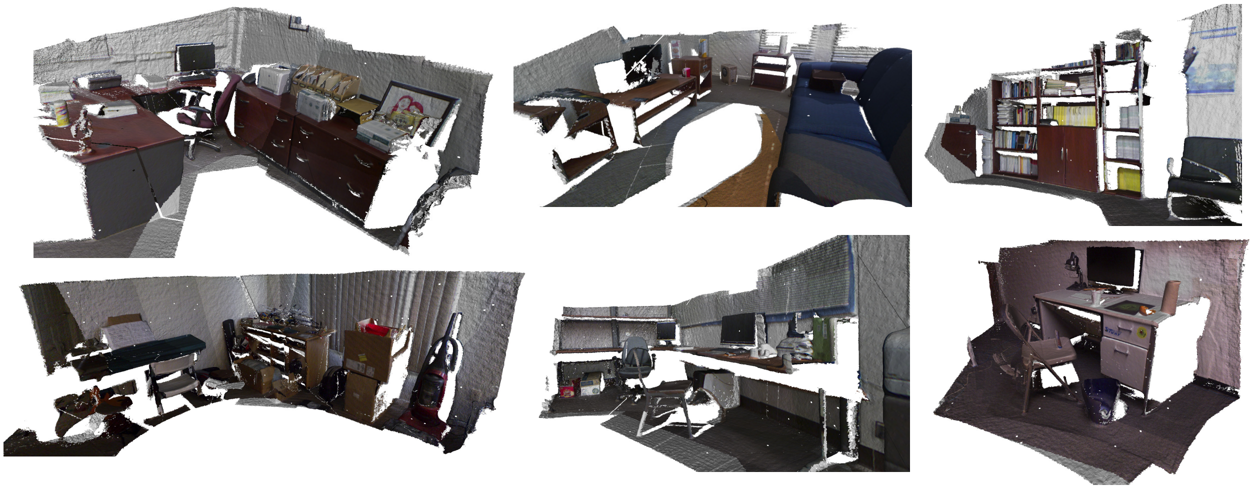


Figure 4.12: Samples of our dataset

tion [PVB08], and our geometric reasoning.

Fig. 4.13 shows some examples of comparing another point cloud segmentation result [PVB08] and our result. However it is worth noticing that, beyond the segmentation task, our method can provide richer information such as volumetric information, physical relations, stabilities, *etc*.

4.6.2 Evaluating Volumetric Completion

For evaluating the accuracy of volumetric completion, we densely sample point clouds from a set of CAD data including 3 indoor scenes. We simulate the volumetric data (as ground truth) and depth images from a certain view (as test images). We calculate the precision and recall which evaluates voxel overlapping between ground truth and the volumetric completion of testing data. Tab. 4.6.2 shows the result that our method has much better accuracy than traditional Octree methods such as [SNI05]. Fig. 4.16 intuitively illustrates the completed objects (bottom row) by our method have more overlaps with ground truth planes (top row in red) than the original sample point clouds (top row in blue).

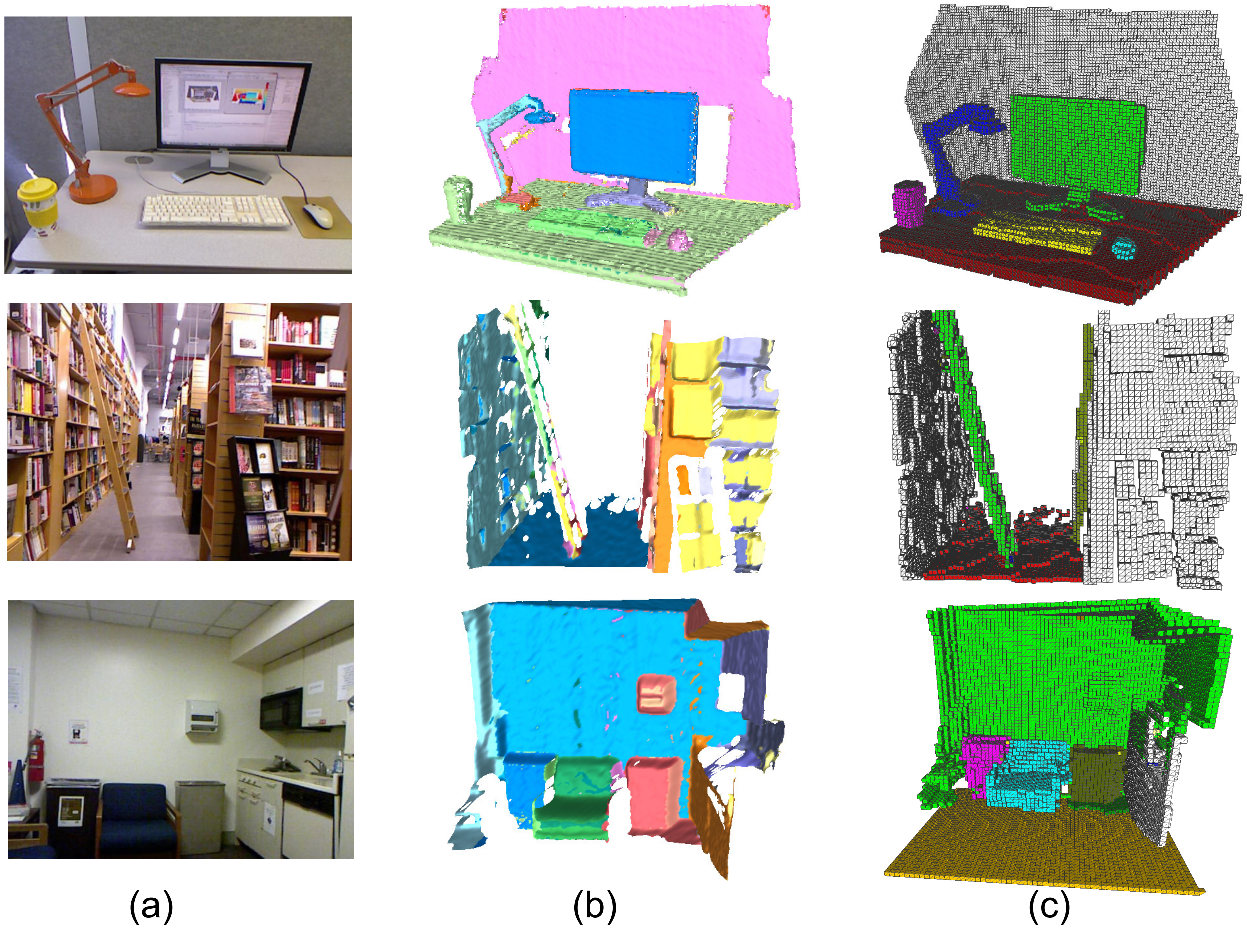


Figure 4.13: Segmentation results for depth images. (a) RBG images for reference. (b) segmentation result by region growing [PVB08]. (c) stable volumetric objects by physical reasoning.

4.6.3 Evaluating Physical Inference Accuracy

Because the physical relations are defined in terms of our contact graph, we map the ground-truth labels to the nodes of contact graphs obtained by geometric reasoning. Than we evaluate our physical reasoning against two baselines: discriminative methods using 3D feature priors similar to the method in [NF12], and greedy inference methods such as the marching pursuit algorithm for physical inference. The result shown in Tab. 4.6.3 is evaluated by the average over 13 scene data captured by Kinect Fusion.

Figure 4.17 (a)-(d) and (e)-(j) show two examples from the results. Here we discuss some irregular cases illustrated by close-ups of the figures.

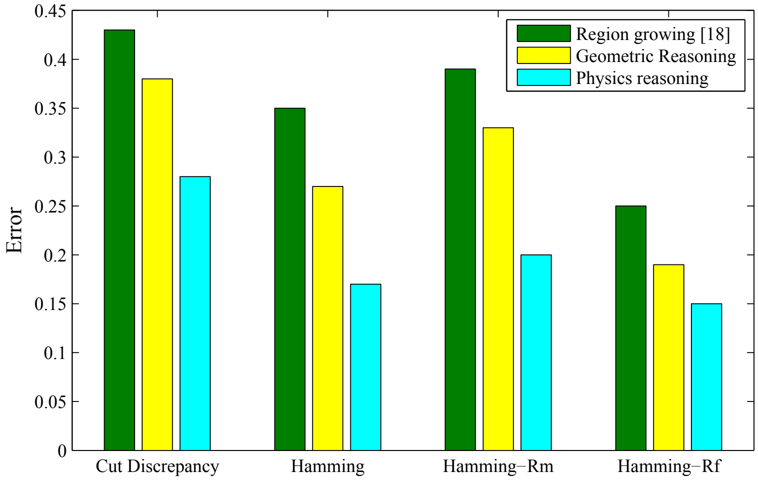


Figure 4.14: Segmentation accuracy comparison of three methods: Region growing method [PVB08], result of our geometric reasoning and physical reasoning by one "Cut Discrepancy" and three "Hamming Distance".

	Octree	Invisible space	Vol. com.
Precision	98.5%	47.7%	**94.1%**
Recall	7.8%	95.1%	**87.4%**

Table 4.2: Precision and recall of Volumetric completion. Comparison of three method: 1) voxel-based representation generated by Octree algorithm [SNI05], 2) voxels in surface and invisible space (sec. 2.2), and 3) our volumetric completion.

Case I: Figure 4.17 (c) the ball is fixed onto the handle of sofa. The reason can be considered as: stability of the "ball" is very low measured by Eq. (4.6). The unstable state is calculated out as that it trends to release much potential energy (draw from the sofa) by absorbing little possible energy (*e.g.*, the disturbance by human activity).

Case II: Figure 4.17 (d) the "air pump" unstably stands on floor but is an independent object, because although its stability is very low, the penalty designed in Eq.(7) penalized it to be fixed onto the floor. The lamp is not affixed for the same reason, as shown in Figure 4.17 (h).

Case III: Figure 4.17 (g) the "empty Kinect box" with its base is fixed together with the shelf, because of the mis-segmentation of the base, *i.e.*, the lower part of base is mis-merged to the top

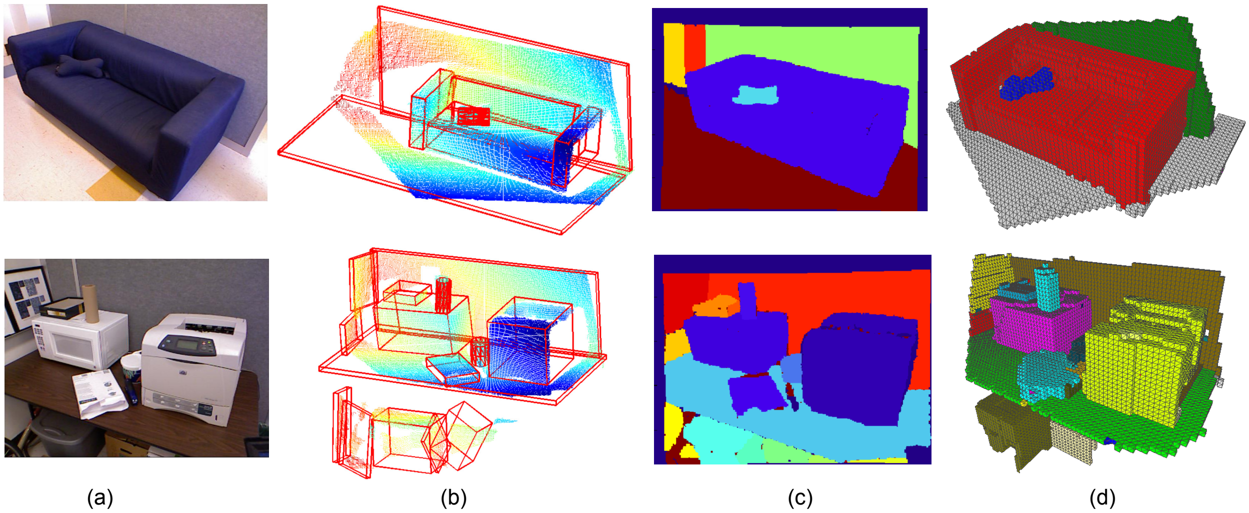


Figure 4.15: Intuitive result comparison: (a) original RGB iamges for reference, (b) 3D point cloud and boxes calculated with the method proposed by [JGS13], (c) the corresponding segmentation result of (b), and (d) our result.

relations	Discriminative	Greedy	SWC
fixed joint	20.5%	66%	**81.8%**
support	42.2%	60.3%	**78.1%**

Table 4.3: Results of inferring the fixed joints and support relations between primitives. Accuracy is measured by nodes of the contact graph whose label is correctly inferred divided by the total number of labeled nodes.

of the shelf.

Case IV: Figure 4.17 (i) voxels under the “chair” are completed with respect to stability. The reasons are: 1) our algorithm reasons the hidden part occluded in invisible space. 2) the inference of the hidden part is not accurate geometrically, but it helps to form a stable object physically. In contrast, the original point cloud shown in Figure 4.17 (j) misses more data.

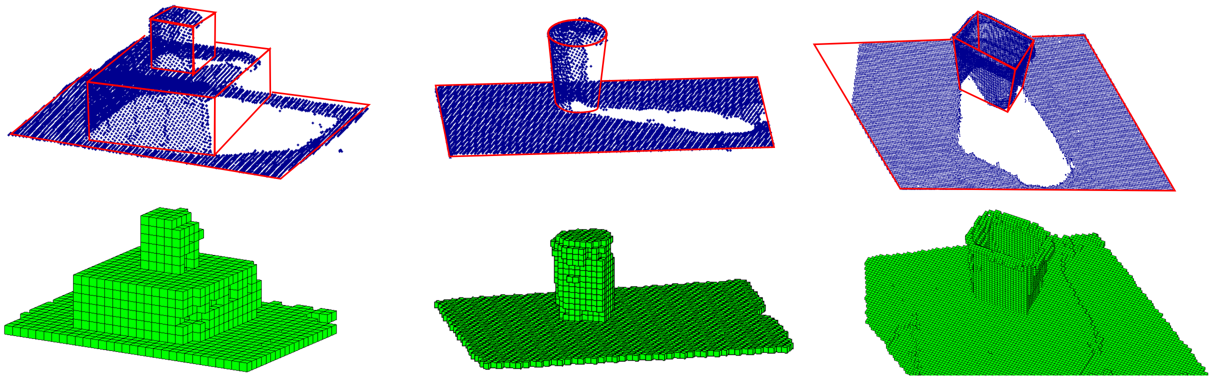


Figure 4.16: Examples of volumetric completion. Top row: densely sampled point clouds (in blue) in a view direction with missing parts referring to the original shape guide lines (in red). Bottom row: volumetric completions of the objects in top row.

4.6.4 Running Time

All the experiments were implemented in Matlab 2012a with a modern PC having an Intel core i7 CPU, 3.4 GHz, and 16 GB memory. For dealing with one single image, such as shown in Fig. 4.13, the running time is around 2 minutes. For large scene data, such as the cases shown in Fig. 4.17 and 4.12 , the running time is around 7-12 minutes.

4.6.5 Evaluating Safety Ratings

First we provide a selected qualitative result shown in Fig. 4.18. We compare the potential falling objects under three different disturbance fields: 1) The human action field in Fig. 4.18 (b,e); 2) The wind field (a uniform directional field) in Fig. 4.18 (c,f) and 3) earthquake (random forces on contacting object surface) in Fig. 4.18 (d,g). As we can see the cups with red tags are detected as potential falling objects, which is very close to human judgments: (i) objects around the table corner are not safe w.r.t human walking action; (ii) objects along the edge of wind direction are not safe w.r.t wind disturbance; and (iii) object along all the edges are not safe w.r.t earthquake disturbance.

Next, we report qualitative results in different 3D scenes, as shown in Fig. 4.19 top row: vending machine room and bottom row: copy machine room. We can see that, according to human

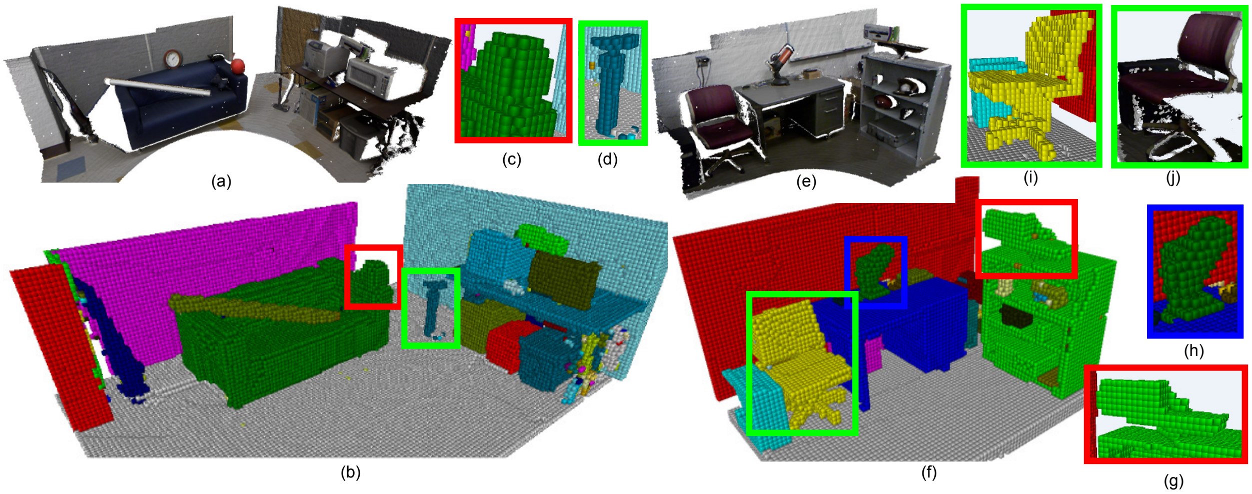


Figure 4.17: Example results. (a) and (e): data input. (b) and (f): volumetric representation of stable objects. (c): the ball is fixed onto the handle of sofa. (d): the "pump" is unstable (see text). (i): a irregular case of (g). (j): hidden voxels under chair compared to (h).

motions, the cans on vending machine room has a risk of being kicked off, while the can near the window is considered stable, since people can rarely reach there. In the copy room, the objects put on the edges of table are at more risk than others.

4.6.6 Discussion

For evaluating safety ratings, we rank object unsafeness in a scene in comparison with human subjects. Fig. 4.20 (a) shows a 3D scene (constructed in CAD design), from which we pick 8 objects and ask 14 participants to rank the unsafeness of these objects considering gravity, common life activity and the risk of falling. We compare the human ranking with our unsafeness function $R(a, \mathbf{x})$ in Fig. 4.20 (b). We found that 1) humans got big variations while considering the safeness, due to deeper consideration of information such as material; 2) however, the model got similar ranking scores with the average of human rankings. As shown in Fig. 4.20 (b), the average of human vs. model scores for each object lies near to the diagonal line.

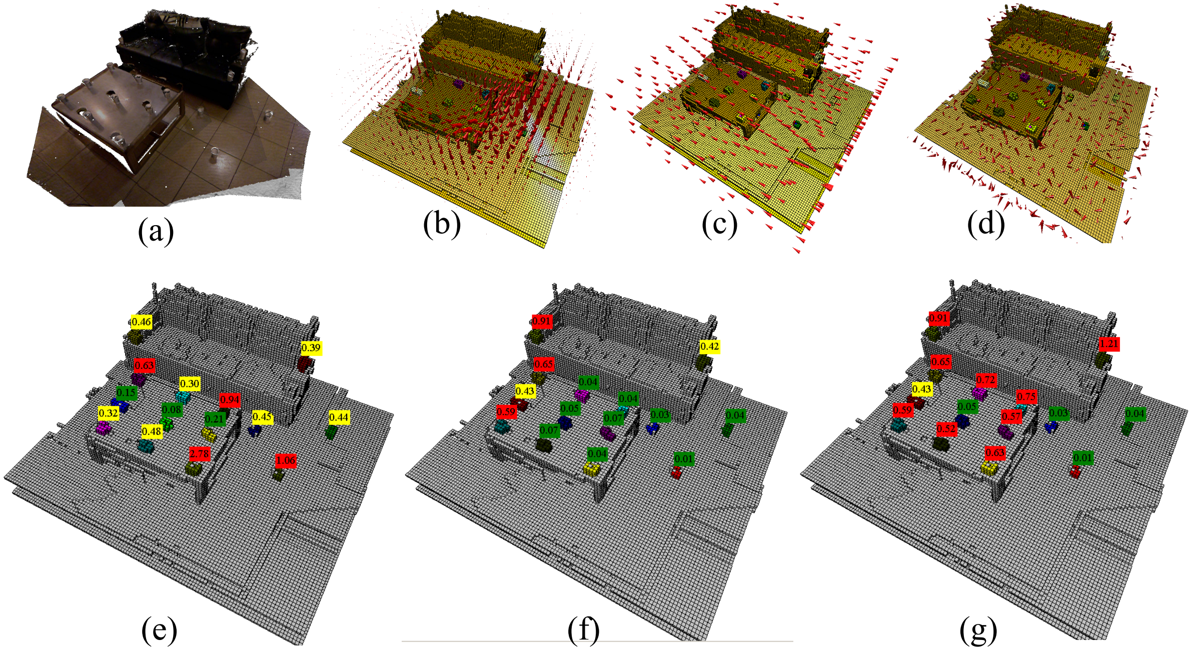


Figure 4.18: The potential falling objects (with red tags) under the human action field (b,e), the wind field (c,f) and the earthquake field (d,g) respectively. The results match with human perception: (i) objects around table corner are not safe w.r.t human walking action; (ii) object along the edge of wind direction are not safe w.r.t wind disturbance; and (iii) object along all the edges are not safe w.r.t earthquake disturbance.

4.7 Summary

We present a novel approach for scene understanding by reasoning their instability and risk using intuitive mechanics with the novel representations of the disconnectivity graph and disturbance fields. Our work is based on a seemingly simple but powerful observation that objects, by human design, are created to be stable and have maximum utility (such as freedom of movement). We demonstrated its feasibility in experiments and show that this provides a new method for object grouping when it is hard to pre-define all possible object shapes and appearance in an object category.

This paper also presents a novel approach for detecting potential unsafe objects. We demonstrated that, by applying various disturbance fields, our model achieves a human level recognition rate of potential falling objects on a dataset of challenging and realistic indoor scenes. Differing from the traditional object classification paradigm, our approach goes beyond the estimation of 3D scene geometry. The approach is implemented by making use of “causal physics”. It first infers hidden and situated “causes” (disturbance) of the scene, and introduces intuitive mechanics to predict possible “effects” (falls) as consequences of the causes. Our approach revisits classic physics-based representation, and uses the state-of-the-art algorithms. Further studies along this way, including friction, material properties, causal reasoning, can be very interesting dimensions of vision research.

In future research, we plan to explore several directions: i) Connecting our work to human psychology models like the one in [HBT11], and to compare our results with human experiments; ii) Studying material properties in typical indoor scenes, and thus to reason about the materials jointly with stability, especially if we can see object movements in video; iii) Combing the physical cues with other appearance and geometric informations for scene parsing; and iv) Studying other specific action distributions to reason about whether a room is safe to children and infants.

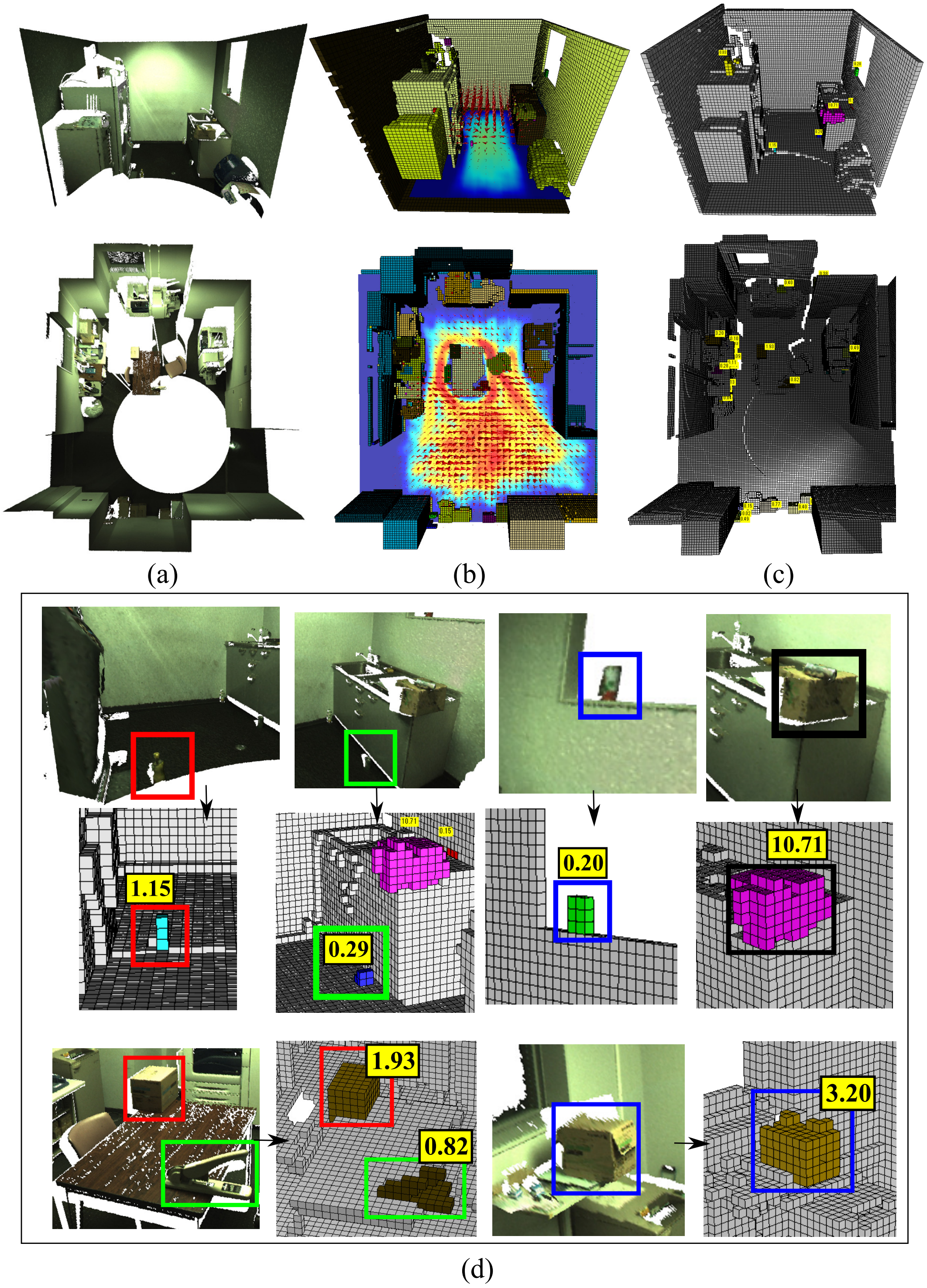


Figure 4.19: (a) Input 3D scene point clouds; (b) Segmented volumetric objects in different colors and inferred disturbance fields of human activity; (c) objects with risk scores. (d) Zoom-in details of detected potential falling objects.

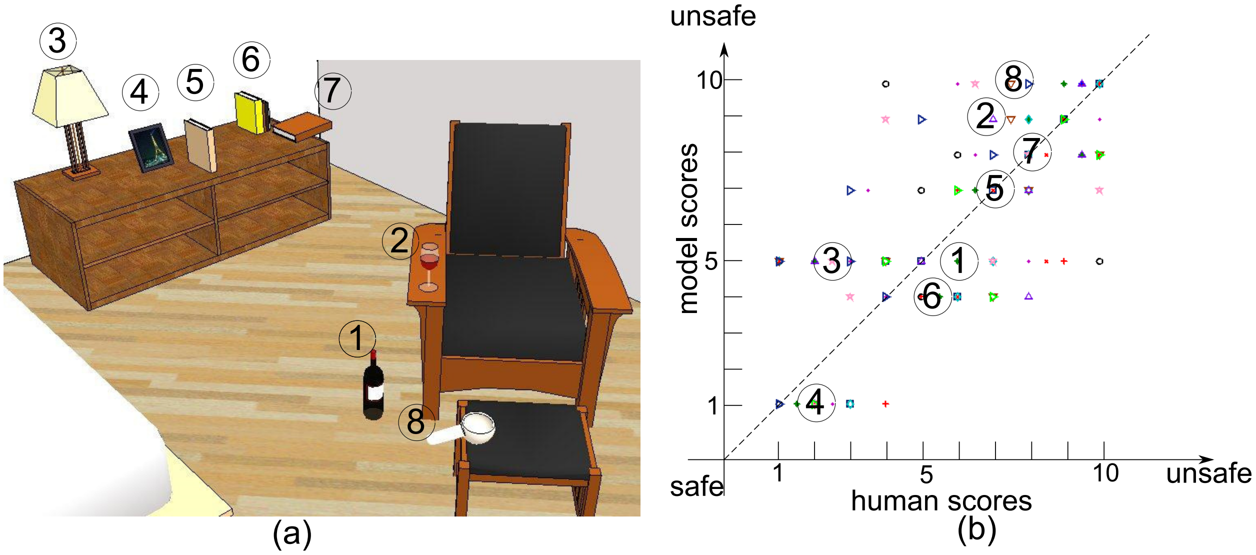


Figure 4.20: Scoring object unsafeness in a scene (a) with 8 objects. We show the correlation graph (b) with human score against our measurement $R(a, \mathbf{x})$ which is normalized from 1 to 10. Color/shape points show human vs. model scores corresponding to different persons. Circle points with numbers inside show the average of human vs. model scores for each object corresponding to (a).

CHAPTER 5

Acquisition of the Commonsense – Three case studies

5.1 Learning Affordance from Observation

Recognizing events, segmenting sequence, and localizing objects in RGBD video are important yet challenging problems in computer vision. Events in RGBD video are with complex temporal structures and often disrupted by noisy human poses and motions. Objects in RGBD video present huge appearance variances due to the occlusions, view variations, and low resolutions, as the instances of *cellphone* from the video sequence shown in Fig. 5.1 (b).

While the previous work mostly conducted those tasks separately [MR06, LN06, SM00, ZTH13, DT05, KF12], we model human-object interactions and jointly deal with those tasks to overcome the challenges. Events and objects in RGBD videos are in a 4D spatially-temporally interactive space. In 1D time, an event is decomposed into several sequential atomic events [PJZ11]. In 3D space, an event is jointly braced by the human pose, objects, and their interaction relations. For example, the events *drink with mug* and *call with cellphone* are hard to be distinguished by human body motions, because they are both performed by the upper body parts and have similar motion forms. When incorporating the interacting objects - *mug* and *cellphone*, the two events can be better distinguished because *mug* and *cellphone* have different appearances, and their interactions with human are different.

On the other hand, human action can help to improve the accuracy of object recognition and localization. In fact, the man-made objects are mainly defined by function rather than appearance. Function carries the information of relation between the human action and object, i.e. what human can do with an object. The ability that an object can afford a human to perform an action is known as the affordance [Gib77, GGG11, ZZ13]. By recognizing actions, we can predict objects' attributes (e.g. category and location) according to the affordance. For example, a cellphone is a cellphone not only because of its specific appearance, but mainly its ability to allow the human to perform the action *make a call*. When someone is making a call in the video, it is hard to recognize and localize the occluded cellphone. But we can reasonably predict it according to the action and affordance. By recognizing the action *make a call*, we can reasonably predict the object's category-*cellphone* and location-*in the hand*, even if the cellphone itself is occluded by the human hand. This is like a 'pantomime'- even if the actor performs actions silently, the audiences can catch the hidden stories according to the actions.

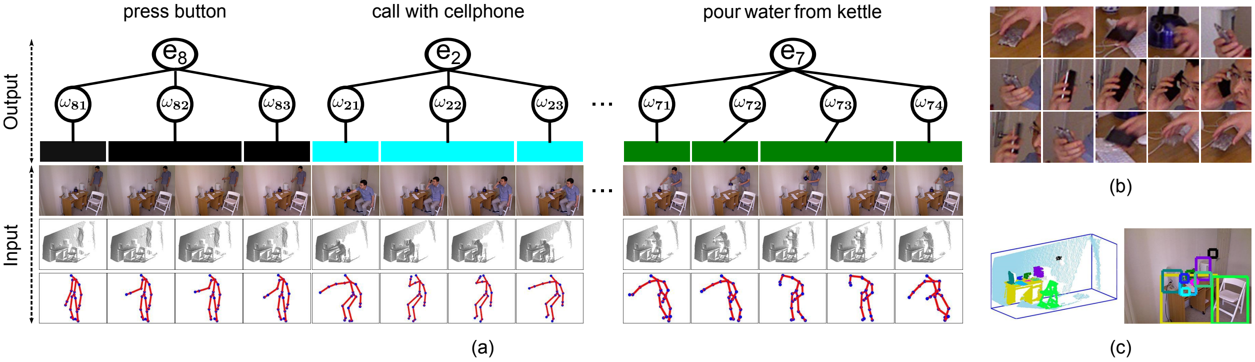


Figure 5.1: The 4DHOI affordance model. (a) The framework of the model. The inputs are the RGBD videos and human skeletons, and the outputs are the hierarchical interpretations to the video sequence, including the event recognition, sequence segmentation, and object localization. (b) Object instances of *cellphone* in the video sequence of the event *call with cellphone*. (c) Object recognition and localization both in the 3D point cloud and RGB image.

We define an event as a sequence of time-varying interactions between human and objects with hierarchical structures in 4D spatial-temporal space, and an object as the entity which affords the human to perform actions in 4D spatial-temporal space.

We propose a unified framework for affordance modeling - 4D human-object interaction (4DHOI) - to jointly: i) recognize events, ii) segment video sequences, and iii) localize objects in RGBD video. The inputs are the RGBD video and 3D human pose sequence estimated by the motion capture technology [SFC11]. The event, object, and human pose form a hierarchical structure in the 4D human-object interaction space, as shown in Fig. 5.2. We formulate this structure with a stochastic hierarchical graph.

We design a dynamic programming beam search algorithm to solve the model. The possible interpretations to each frame are proposed according to the human pose, the objects being searched, and the 3D geometric relations between them. The temporal relations between frames are incorporated to optimize these proposals. In this way, the video is hierarchically interpreted and the corresponding objects are labeled.

To learn the graph structure and model parameters, we propose an ordered expectation maximization (OEM) algorithm. Different from the conventional EM [Bis06], our OEM incorporates temporal orders of video frames and temporal alignments of atomic events into clustering. It therefore produces temporally-continuous clusters.

We built a large-scale multiview 3D event dataset and has released it to public [1]. It is captured by three stationary Kinect cameras from different viewpoints simultaneously. It includes 8 event categories, 11 interacting object classes, 3815 event video sequences, and 383,036 RGBD frames. We test our method on this dataset and the experiment result demonstrates the strength of the model.

4D Human-Object Interaction Model Event has the hierarchical structure in 4D spatial-temporal space. In 1D temporal domain, an event is decomposed into multiple ordered atomic events. An atomic event corresponds to a continuous segment in the video sequence. It is composed of successive frames which contain similar human poses and object interactions. Atomic events are hidden variables, which means we do not manually label the segment of each atomic event in learning. Fig. 5.2 shows the event *fetch water from dispenser* is decomposed into three sequential atomic events-*approach the dispenser*, *fetch water*, and *leave the dispenser*.

In 3D spatial domain, an atomic event is decomposed into human pose, interacting objects, and the relations between them. An atomic event integrates a specific type of human pose and one or multiple objects. The semantic relation between the object category and a specific atomic event is hard constraint. For example, the atomic event *fetch water* consists of the pose *fetch* and the objects *dispenser, mug*, as is shown in Fig. 5.2.

We formulate the 4D hierarchical structure with a stochastic hierarchical graph, which is similar

[1] http://vcla.stat.ucla.edu/download.html

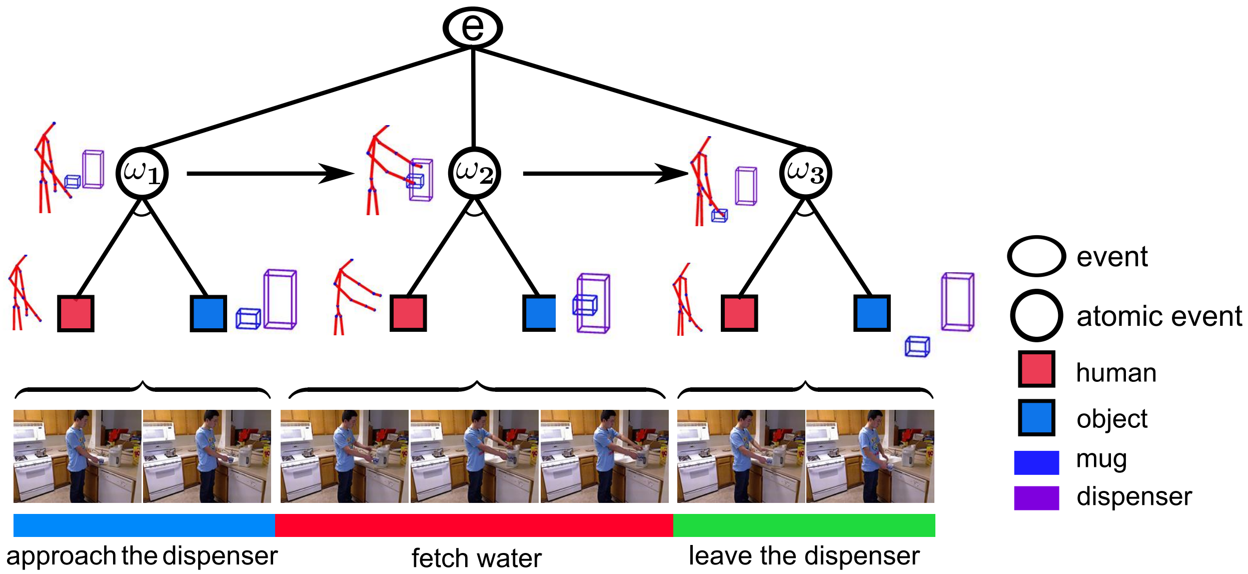


Figure 5.2: Hierarchical graph model of event.

to the And-Or Graph (AoG) [ZM07]. Suppose $V = (f_1, ..., f_\tau)$ is an event video sequence in the time interval $[1, \tau]$. $f_t = (I_t, h_t)$ is the frame at time t, where I_t is the RGBD data and h_t is the human pose in the forms of 3D skeleton joints estimated by the motion capture technology [SFC11]. The sequence V is interpreted with the hierarchical graph $G =< E, L >$:

i) $E \in \Delta = \{e_i | i = 1, ..., |\Delta|\}$ is the event category like *fetch water from dispenser*. Δ is the set of event categories.

ii) $L = (l_1, ..., l_\tau)$ is a sequence of frame labels. $l_t = (a_t, o_t)$ is the interpretation to the frame f_t. $a_t \in \Omega_E = \{\omega_i | i = 1, ..., K_E\}$ is the atomic event label like *fetch water*. Ω_E is the atomic event set of E. Each event category e_i has its distinct atomic event set Ω_{e_i}, i.e. the relations of an event and its atomic events are hard constraints.

$o_t = (o_t^1, ..., o_t^{n_t})$ are the objects interacting with human, where n_t is the number of objects. Each object includes the attributes of class label and 3D location.

According to the AoG [ZM07], the energy that the video V is interpreted with the graph G is defined as

$$\mathrm{En}(G|V) = \sum_{t=1}^{\tau} \Phi(f_t, l_t) + \sum_{t=2}^{\tau} \Psi(l_{1:t-1}, l_t) \tag{5.1}$$

$\Phi(\cdot)$ is the spatial energy term of single frame. It encodes the human-object interactions in 3D space.

$\Psi(\cdot)$ is the temporal energy term of multiple frames. It encodes the temporal relations between frames in 1D temporal domain. $l_{1:t-1}$ are the labels of all the frames from the time 1 to $t-1$. Here, l_t is not only related to the neighbor l_{t-1}, but also related to all the previous frame labels, which is different from the conventional hidden Markov model.We omit the variable E in the right side of Eq. 5.1 for each event has its own distinct atomic event set.

In the following sections, we will detail the energy terms.

Human-object Interactions in 3D Space

$\Phi(f_t, l_t)$ describes the human-object interactions in 3D space, which includes the semantic co-occurrence and geometric compatibility. Semantic co-occurrence means a specific type of human pose and some object classes appear together in an atomic event. Geometric compatibility describes the spatial constraint between human body and objects in 3D space.

$\Phi(f_t, l_t)$ is decomposed as:

$$\Phi(f_t, l_t) = \phi_1(a_t, h_t) + \phi_2(a_t, o_t, I_t) + \phi_3(a_t, h_t, o_t) \tag{5.2}$$

Pose Model

$\phi_1(a_t, h_t)$ is the human pose model. The human pose with 20 3D joints are estimated by the Kinect [SFC11]. To normalize the data, we align all the skeletons to a reference pose so that the torsos and shoulders of all poses have the same locations, scales, and directions.

The feature of each joint is defined as the 3D coordinate concatenating the motion vector which is the difference of joint coordinates in two successive frames. We extract a feature vector containing the features of joints on arms and apply the PCA to the feature vector to reduce the correlation and noise. h_t is the vector of the PC parameters. We assume that h_t follows a Gaussian distribution, then $\phi_1(a_t, h_t) = -\ln N(h_t; \mu_{a_t}, \Sigma_{a_t})$, where μ_{a_t} is the mean and Σ_{a_t} is the covariance.

Object Model

$\phi_2(a_t, o_t, I_t)$ is the object detection model. Suppose z_t^i is the 3D bounding box center of the object o_t^i in the 3D space. The 3D box is projected into the RGB and depth images to form 2D bounding boxes, in which the RGBD HOG features [DT05, KF12] are extracted. The probability of object o_t^i at z_t^i is obtained by normalizing the SVM detector with Platt scaling $p(o_t^i|z_t^i) = 1/\{1 + \exp\{us(z_t^i) + v\}\}$ [KF12, Pla99], where $s(z_t^i)$ is the score of linear SVM object detector with the RGBD HOG features at location z_t^i. $\phi_2(a_t, o_t, I_t)$ is formulated as

$$\phi_2(a_t, o_t, I_t) = -\frac{1}{n_t}\sum\nolimits_{i=1}^{n_t} \ln p(o_t^i|z_t^i) \tag{5.3}$$

where n_t is the number of objects. The coefficient $1/n_t$ is used to offset the influence of different object numbers.

We use a sliding window detection strategy to search the objects. But different from [DT05, KF12] where the sliding window is defined on the 2D image plane, we slide the 3D window box in the 3D space where the point cloud is not empty. Then the 3D window is projected into the 2D image to extract the appearance feature. Since the instances of the same object class usually have similar sizes in 3D space, we define a prior 3D size for each object class.

Our model defines object location and scale in the 3D space, and appearance in the 2D image, which are more robust to the viewpoint and scale changes. It also provides a natural way to define the geometric relations between human and objects.

3D Geometric Compatibility and Object Prediction

$\phi_3(a_t, h_t, o_t)$ measures the geometric relations between human and objects. Geometric relations in the previous work were mostly defined on 2D image plane [GKD09, PSF12, YF12]. But the 2D geometric relation priors are not applicable in different viewpoints, as is shown in Fig. 5.3. The geometric relation priors in 3D space are invariant to the viewpoint changes. We model the geometric relations in 3D space, and project them into 2D image planes to accommodate viewpoint differences.

Geometric relations between human and objects are time-varying, i.e. they are varying in different atomic events. For the example in Fig. 5.2, the human hand is far from the dispenser in the atomic event *approach the dispenser* while touches it in the atomic event *fetch water*. So each atomic event has its distinct geometric relations.

As mentioned above, objects are categorized into part-scale, body-scale, and scene-scale objects in 3D geometric space, according to the interactions with human. So in an atomic event, an object mainly interacts with some body parts, which we call the key parts, as the left arm to the

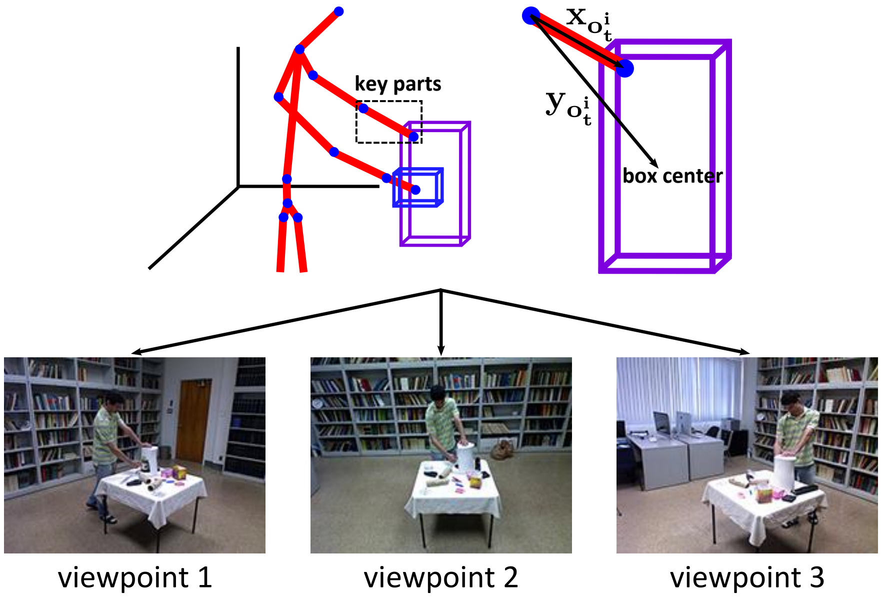


Figure 5.3: Human-object geometric relation in 3D space.

dispenser and right arm to the mug in Fig. 5.3. The object's location in 3D geometric space is closely related to and largely revealed by the key parts' location and direction.

As is shown in Fig. 5.3, suppose $y_{o_t^i}$ is the difference vector from the key parts center to the object bounding box center. $x_{o_t^i}$ is the difference vector between the end points of the key parts. $y_{o_t^i}$ is closely related to $x_{o_t^i}$. We define $\eta_{o_t^i} = y_{o_t^i} - W_{o_t^i}^{a_t} x_{o_t^i}$, where $W_{o_t^i}^{a_t}$ is a similarity transformation matrix. We assume $\eta_{o_t^i}$ follows the Gaussian distribution. The 3D geometric relation is modeled as:

$$\phi_3(a_t, h_t, o_t) = -\frac{1}{n_t} \sum\nolimits_{i=1}^{n_t} \ln N(\eta_{o_t^i}; \mu_{o_t^i, a_t}^R, \Sigma_{o_t^i, a_t}^R) \tag{5.4}$$

where $\mu_{o_t^i, a_t}^R$ is the mean and $\Sigma_{o_t^i, a_t}^R$ is the covariance. The superscript R is a sign which is used to differentiate the 3D relation Gaussian parameters from others. The subscript (o_t^i, a_t) indicates that the human-object geometric relation varies for different atomic events and objects.

The key body parts vector $x_{o_t^i}$ is like a local reference, by which we can estimate $y_{o_t^i}$, and therefore predict the locations of related objects. As is shown in Fig. 5.4, according to the location of key parts, the probability that an object appears at a 3D location can be evaluated with Eq. (5.4).

Temporal Relation

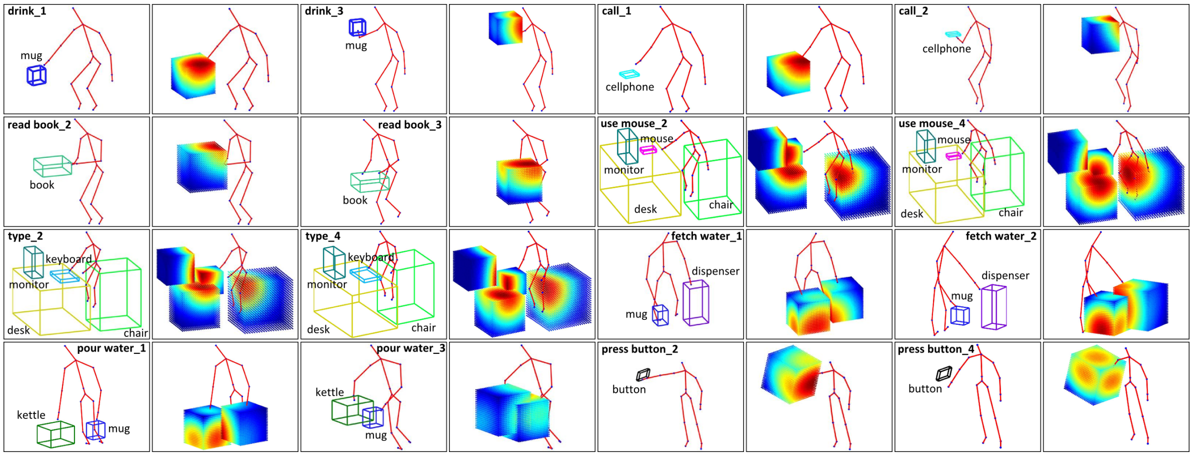


Figure 5.4: Examples of learned atomic events. The examples in the even number columns are the probability maps of object prediction, where warmer colors indicate larger probabilities of the locations where objects appear.

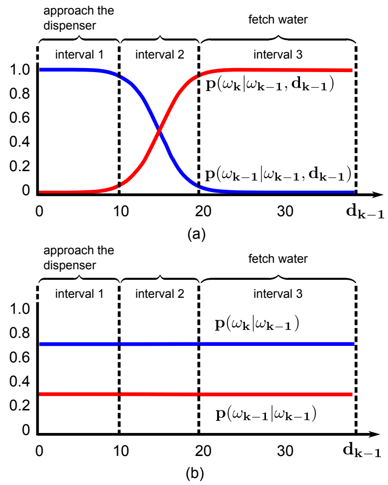


Figure 5.5: The atomic event transition probability. (a) Duration-dependent transition. (b) Duration-independent transition.

The temporal relation $\Psi(l_{1:t-1}, l_t)$ is decomposed as

$$\Psi(l_{1:t-1}, l_t) = \psi_1(a_{1:t-1}, a_t) + \psi_2(o_{t-1}, o_t) \tag{5.5}$$

where $a_{1:t-1}$ are atomic event labels of the frames from time 1 to $t-1$. The first term encodes the atomic event transition, and the second term encodes the temporal coherence of objects.

Atomic Event Transition In an event, the transition probability from the current atomic event to the next atomic event is related to the duration of current atomic event. We propose to model the time-varying transition probability with the logistic sigmoid function.

Suppose ω_{k-1} and ω_k are two neighboring atomic events of event E. Given E and $a_{t-1} = \omega_{k-1}$, the next frame's atomic event a_t can be ω_{k-1} (repeat the same atomic event) or ω_k (start a new atomic event). d_{k-1} is the continuous duration of ω_{k-1} up to time $t-1$. The time-varying transition probability $p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1})$ is modeled as:

$$p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1}) = \sigma(\beta d_{k-1} + \gamma) \tag{5.6}$$

$\sigma(v) = 1/(1 + e^{-v})$ is the logistic sigmoid function. β and γ are the function parameters. We simplify $p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1})$ as $p(\omega_k | \omega_{k-1}, d_{k-1})$. The transition probability to ω_{k-1} is $p(\omega_{k-1} | \omega_{k-1}, d_{k-1}) = 1 - p(\omega_k | \omega_{k-1}, d_{k-1})$. Then $\psi_1(a_{1:t-1}, a_t)$ is modeled as $-\ln p(\omega_k | \omega_{k-1}, d_{k-1})$ or $-\ln p(\omega_{k-1} | \omega_{k-1}, d_{k-1})$, up to the value of a_t.

Figure 5.5 shows two kinds of transition probability. To the duration-dependent transition, at the preliminary stage of *approach the dispenser* when the hand is still far from the dispenser, the probability of transition from *approach the dispenser* to *approach the dispenser* is much larger than the possibility to the next atomic event *fetch water*, as the interval 1 in Figure 5.5 (a). If *approach the dispenser* has been lasting a long time, as in the interval 3, the probability of transition to *approach the dispenser* will be much smaller than the probability to *fetch water*. In interval 2, the transition choice is indeterminate. The interval 1 and 3 describe the common duration distributions while the interval 2 reflects the variance. To the duration-independent transition, the probability is constant, regardless of the duration, as Figure 5.5 (b) shows.

Temporal Coherence of Objects $\psi_2(o_{t-1}, o_t)$ describes the temporal coherence of objects. In an event, the locations of some objects like dispenser are rare to be changed. Some objects like mug can move when human action is applied. To the *moveable* objects, we assume the location follows a Gaussian distribution $p(z_t^i | z_{t-1}^i) = N(z_t^i - z_{t-1}^i; \mu^Z_{o_t^i, a_t}, \Sigma^Z_{o_t^i, a_t})$. To the *non-movable* objects, we set a hard threshold. If the difference of proposed location in the current frame and the last frame is smaller than the threshold, $p(z_t^i | z_{t-1}^i)$ is 1, otherwise 0. The energy is

$$\psi_2(o_{t-1}, o_t) = -\frac{1}{n_t} \sum_{i=1}^{n_t} \ln p(z_t^i | z_{t-1}^i) \tag{5.7}$$

5.2 Learning Tool-Use from Single Demonstration

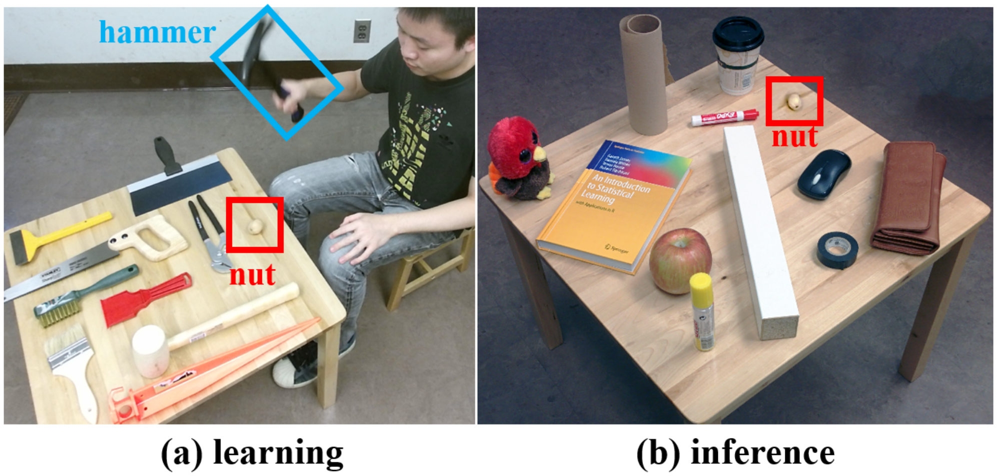


Figure 5.6: Learning Tool-Use from Single Demonstration. (a) In a learning phase, a rational human is observed picking a hammer among other tools to crack a nut. (b) In an inference phase, the algorithm is asked to pick the best object (i.e. the wooden leg) on the table for the same task. This generalization entails physical reasoning.

In this paper, we rethink object recognition from the perspective of an agent: how objects are used as "tools" in actions to accomplish a "task". Here a task is defined as changing the physical states of a target object by actions, such as, cracking a nut or painting a wall. A tool is a physical object used in the human action to achieve the task, such as a hammer or brush, and it can be any daily objects and is not restricted to conventional hardware tools. This leads us to a new framework – task-oriented modeling, learning and recognition, which aims at understanding the underlying functions, physics and causality in using objects as tools in various task categories.

Fig. 5.6 illustrates the two phases of this new framework. In a learning phase, our algorithm observes only one RGB-D video as an example, in which a rational human picks up one object, the hammer, among a number of candidates to accomplish the task. From this example, our algorithm reasons about the essential physical concepts in the task (e.g. forces produced at the far end of the hammer), and thus learns the task-oriented model. In an inference phase, our algorithm is given a new set of daily objects (on the desk in (b)), and makes the best choice available (the wooden leg) to accomplish the task.

From this new perspective, any objects can be viewed as a hammer or a shovel, and this generative representation allows computer vision algorithms to generalize object recognition to novel functions and situations by reasoning the physical mechanisms in various tasks, and go beyond memorizing typical examples for each object category as the prevailing appearance-based recognition methods do in the literature.

Fig. 5.7 shows some typical results in our experiments to illustrate this new task-oriented object recognition framework. Given three tasks: chop wood, shovel dirt, and paint wall, and three groups of objects: conventional tools, household objects, and stones, our algorithm ranks the objects in each group for a task. Fig. 5.7 shows the top two choices together with imagined actions using such objects for the tasks.

Our task-oriented object representation is a generative model consisting of four components in a hierarchical spatial-temporal parse graph:

i) An *affordance basis* to be grasped by hand;

ii) A *functional basis* to act on the target object;

iii) An *imagined action* with pose sequence and velocity;

iv) The *physical concepts* produced, e.g. force, pressure.

In the learning phase, our algorithm parses the input RGB-D video by simultaneously reconstructing the 3D meshes of tools and tracking human actions. We assume that the human makes rational decisions in demonstration: picks the best object, grasps the right place, takes the right action (poses, trajectory and velocity), and lands on the target object with the right spots. These decisions are nearly optimal against a large number of compositional alternative choices. Using a ranking-SVM approach, our algorithm will discover the best underlying physical concepts in the human demonstration, and thus the essence of the task.

In the inference stage, our algorithm segments the input RGB-D image into objects as a set of candidates, and computes the task-oriented representation – the optimal parse graph for each candidate and each task by evaluating different combinations. This parse graph includes the best object and its tool-use: affordance basis (green spot), functional basis (red spot), actions (pose sequence), and the quantity of the physical concepts produced by the action.

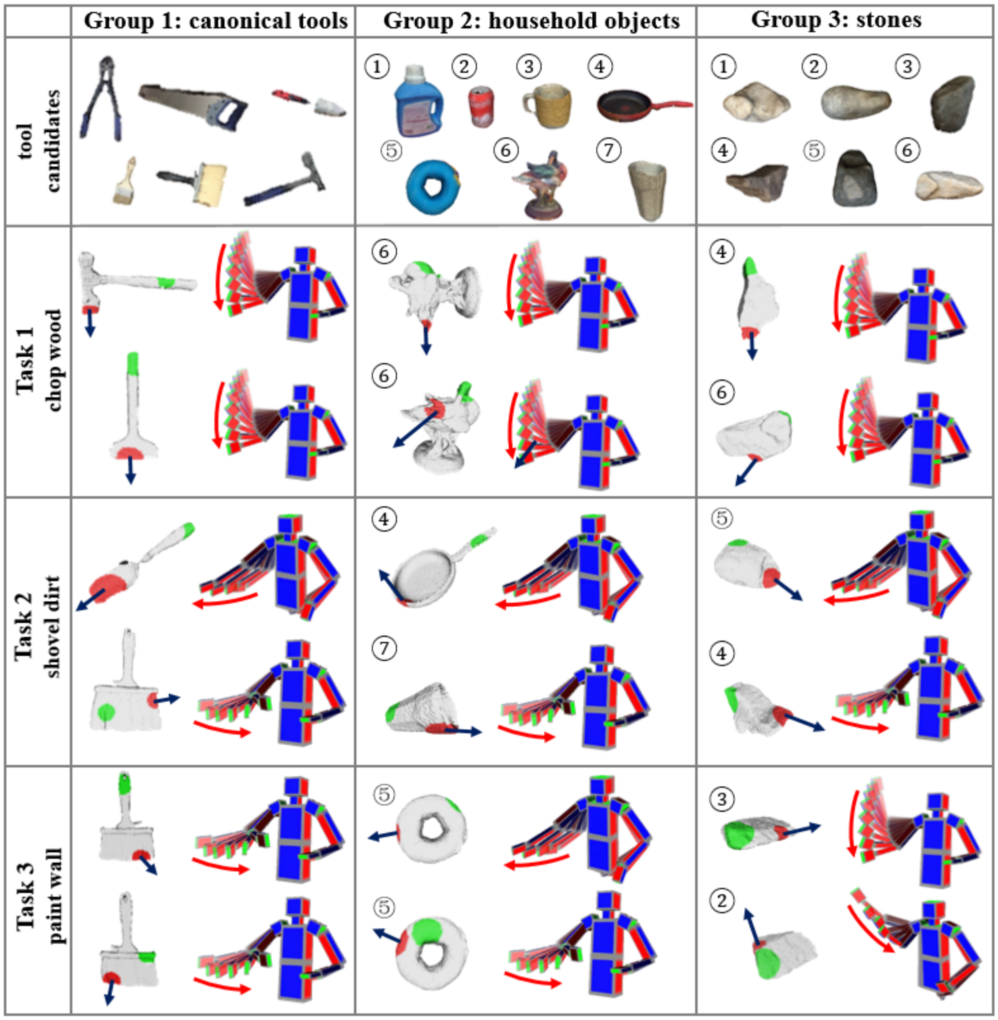


Figure 5.7: Given three tasks: chop wood, shovel dirt, and paint wall. Our algorithm picks and ranks objects for each task among objects in three groups: 1) conventional tools, 2) household objects, and 3) stones, and output the imagined tool-use: affordance basis (the green spot to grasp with hand), functional basis (the red area applied to the target object), and the imagined action pose sequence.

This paper has four major contributions:

1. We propose a novel problem of task-oriented object recognition, which is more general than defining object categories by typical examples, and is of great importance for object manipulation in robotics applications.

2. We propose a task-oriented representation which includes both the visible object and the

imagined use (action and physics). The latter are the ’dark matter’ in computer vision.

3. Given an input object, our method can imagine the plausible tool-use and thus allows vision algorithms to reason innovative use of daily object – a crucial aspect of human and machine intelligence.

4. Our algorithm can learn the physical concepts from a single RGB-D video and reason about the essence of physics for a task.

5.3 Inferring Containing Relations by Physical Simulation

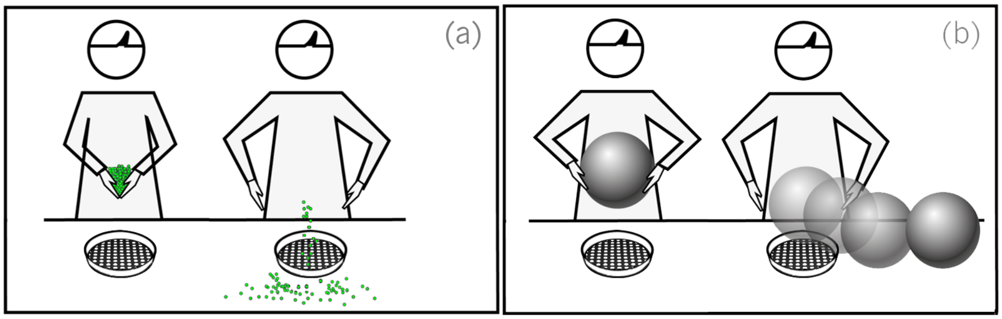


Figure 5.8: Two typical cases when a container fails to contain its contents: (a) the container with holes can not contain tiny objects; (b) the container with a low wall fails to contain a big ball. The left figures of these two panels illustrate a stimuli of our experiments, and the right figures illustrate simulation results with physical engine or in human mind.

Containers are ubiquitous objects in daily life, such as bowls, bottles, baskets, trash cans, refrigerators, etc. Containing relation is a general and fundamental relation in the scene. Containers offer containing relations for carrying, hiding, or ensuring the objects remain in a safe place. The contained objects are called contents. The containing relation characterizes the “affordance” that how likely a container can hold its content.

Different from visual object recognition problems, recognition of containers involves the cognitive process of commonsense reasoning, such as analysis of physical properties, geometric shapes, and material properties, etc. Fig.5.8 shows two examples when a container fails to contain its content: (a) the container with holes can not contain tiny objects or staffs, like beads, sand or water; (b) the container with a low wall fails to contain a big ball.

Containers quantize and organize our perceptual scene space. For example, when people are asked “where the chilled beer is”, the answer will usually be that “it is in the refrigerator” without mentioning the exact 3D coordinates. By containers, the perceptual space of 3D scene is discretized and quantized, and objects are often organized in a hierarchy with respect to their containing relations [ZZ13]. This quantization largely simplifies many tasks, such as planning, detection and tracking.

Inspired by [BHT13] and [ZZY15], human perceive physical scenes by making approximate and probabilistic inference, and the physical engine helps us to reason about common-sense in complex scenes. When we ask about whether a container will hold another object, human may do similar mental simulations. The definition of containers are related to physical properties of containers and contents. In Fig.5.8, the container and its contents are not compatible in these two cases. In this paper, we model and infer the containing relations between two objects by imagining what would happen when one puts an object into a container.

In order to study containers and the factors which affect containing relations, we collected a 3D container dataset and carry out our experiments based on it. In the experiment, we presented some random sampled 3D objects from our dataset to the subjects. The subjects answered questions about container and containing relations according to these pictures. We also built an online physical simulation system with Unity 3D engine on a tablet platform as shown in Fig.5.9. The system is used for evaluating containing relations between objects and comparing with human judgments.

3D container dataset

In the experiment, we built a 3D container dataset including 315 real-world 3D objects. The data was collected using a 3D Structure Sensor attached to a tablet platform as shown in Fig.5.9 (a). The objects in our dataset are full 3D models reconstructed by computer vision algorithms. We then conduct our experiment with these real-world 3D objects. Some results are shown in Fig.4.12. Comparing with previous cognitive studies, our experiments use daily objects in natural physical scenes.

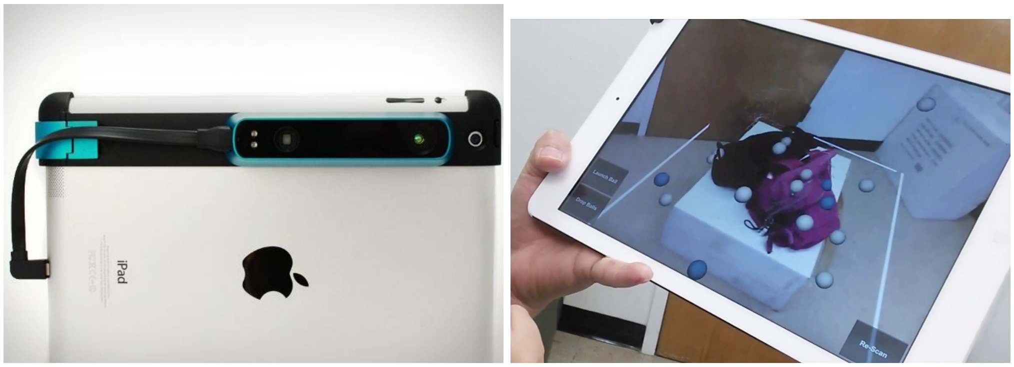


Figure 5.9: A 3D Structure sensor attached to a tablet (left) and the software interface of our simulation engine (right). The software can simulate the object dynamic with respect the gravity. The interface shows a scenario when we simulate balls fall onto an bag.

Participants and stimuli

We conduct human studies with fifty human subjects who are university students around age 25. We are interested in three main questions: i) What is a container? ii) Can an object A contain another object B? iii) How many objects will a container hold? For each of these three questions, we show a 3D scene as a stimulus to human subjects, and ask them to answer a corresponding question. The objects in the 3D scenes are generated randomly from our 3D container dataset.

Physical simulation

We set up a physical simulation system with Unity 3D engine to infer the probability for an object to be a container and containing relations between two objects. We place a 3D object as a potential container on a virtual ground, and initialize another object as its potential content over the container with a few random parameters, i.e. relative height, position, pose, and initial speed. Initializing the 3D scene by randomly sampling these parameters, we calculate the frequency of successful cases of containments through physical simulations. In the physics engine, we model the potential container by a ”Mesh Collider” which calculates the collisions for all the triangle faces (around 17000) on the object. And we simplify the 3D model of potential content to 255 triangle faces, and approximate its physical dynamics by a “Convex Collider” for the consideration of computation feasibility.

Exp. 1: What is a container?

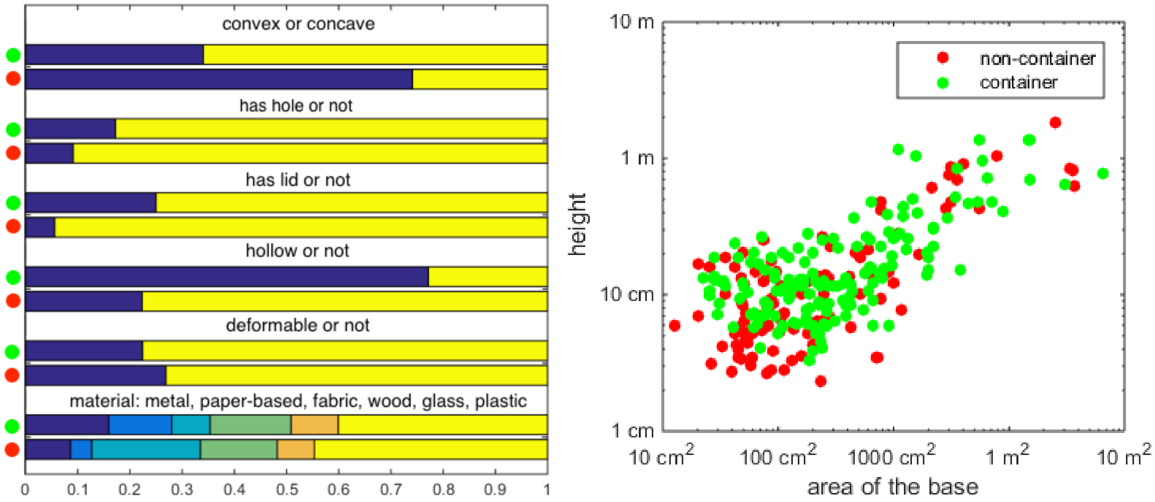


Figure 5.10: The distribution of different container attributes. In the left bar plot, a pair of horizontal bars represents the distribution of containers and non-containers for each discrete attribute; in the right scatter plot, the green and red dots illustrate the distribution of containers and non-containers with respect to the area of the base and height of these 3D objects.

In this experiment, we let subjects see a 3D object and ask following questions: i) is it a container? ii) is it a convex shape? iii) does it have a hole? iv) does it have a lid? v) is it hollow? vi) is it deformable? vii) what kind of material is it?

The figure on the left of Fig.5.10 shows the distribution of six attributes associated to these questions. For each attribute, we plot distributions for both container and non-container. For example, most of the containers are concave shapes, and most of the non-containers are convex. The last material attribute takes categorical values of “metal”, “paper-based”, “fabric”, “wood”, “glass”, and “plastic”.

The distribution of object sizes of the dataset are also showed on the right of Fig.5.10. The size of the object covers from the hand size (a few centimeters) to the body size (a few meters). The size distributions of containers (green dots) and non-containers (red dots) in the dataset are very similar.

Logistic regression analysis for attributes

We analyze the contribution of different attributes to the notion of “container” by logistic regression. We use five binary variables: (convex, has hole, has lid, hollow, deformable), one cate-

Table 5.1: Analysis of logistic regression coefficients.

	Estimate	Std. Err	tStat	pValue
(Intercept)	-3.1168	1.1114	-2.8043	0.005043
convex	-1.8572	0.2692	-6.8999	**5.204e-12**
has hole	0.1248	0.3814	0.3274	0.7434
has lid	1.4893	0.4086	3.6449	0.0002675
hollow	2.2661	0.2736	8.2818	**1.2132e-16**
deformable	-0.7816	0.3067	-2.5485	0.01082
material	0.1712	0.0754	2.2714	0.02312
height	-0.8198	0.5969	-1.3733	0.1697
base area	0.4308	0.2580	1.6702	0.09489

gorical variable (material), and two continuous variables (height and base area) as predictors. The algorithm aims to analyze the influence of different variables for answering the target question “is it a container or not?”.

The results of the regression are shown in Table.5.1.The attributes convex and hollow with low p-values are statistically significant for discriminating the concept of containers.

Container recognition

We address the containers recognition problem as a computer vision problem. We compare two algorithms: 1) classic computer vision algorithm by pattern-recognition, 2) physical simulation-based method as introduced before.

We used a state-of-the-art discriminative classifier based on Hierarchical Kernel Descriptors [**?**]. In order to apply the classic computer vision method, we project the 3D model to RGB images and depth images from canonical views. And we use the RGB images and RGBD images for training and testing the computer vision algorithm. For comparison, we also test the simulation-based method on the same testing set of 3D objects. The probability is calculated by the expected value for containing another objects in the dataset.

In order to evaluate the generalization ability of these algorithms, we test them on three differ-

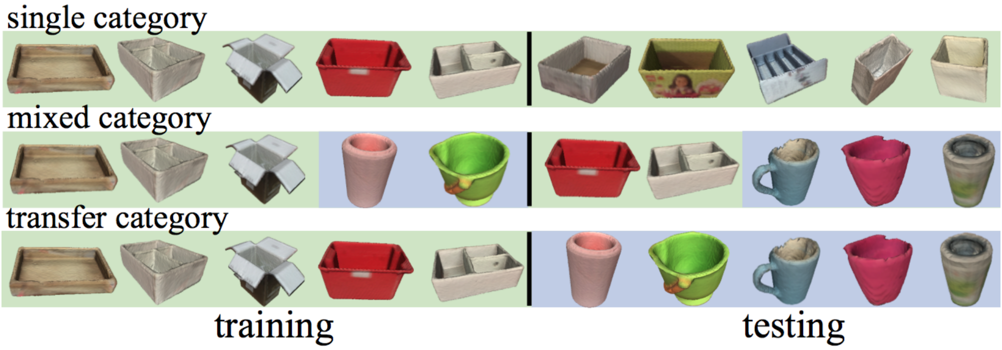


Figure 5.11: The split of training / testing for container recognition. i) The single category: both training and testing samples come from a same single category. ii) The mixed category: both training and testing samples come from a collection of multiple categories. iii) The transfer category: the training samples come from one category, while the testing samples come from another category. The results show in Table.5.2.

ent scenarios:

i) The single category: both training and testing samples come from the same single category, such as boxes. ii) The mixed category: both training and testing samples come from a collection of multiple categories. iii) The transfer category: the training samples come from one category, such as boxes, while the testing samples come from another category, such as cups. The results are summarized in Table.5.2. It is worth noting that the [**?**]'s algorithm works well on single category. Because the simulation-based algorithm does not need any training, and the physical laws are generally applicable, physical simulation-based algorithm has advantages for generalizing across categories.

Exp. 2: Will an object contain another?

In the experiment, we evaluate the “affordance” of a container. Human subjects are shown a 3D scene with two 3D objects randomly sampled from the dataset. One is a potential container, another is a potential content. Some of stimuli are shown in Fig.5.14.

We applied two kinds of approaches to model the containing relations between two objects. i) Regression model. We use features including relative height ratio, base area ratio, and volume ratio, to learn a logistic regression model. ii) Physical simulation model. We compare the results of

Table 5.2: Accuracy of container recognition

	RGB	RGB-Depth	Simulation
single category	0.89	**0.94**	0.93
mixed categories	0.70	0.78	**0.93**
transfer category	0.35	0.59	**0.93**

both models with respect to human judgments in Fig.5.12 (a,b). And we also show the correlations between two human subjects on the right of Fig.5.12. We can see that this task is very challenging, as there are diverse judgments even between human subjects. Although the regression method can capture some correlation between the relative size and the containing relation, the results of simulation model show much strong collinearity with the human subject. The area between two blue lines are the variance interval between 25% percentile and 75% percentile, which means a half of the samples will fall into the region between two blue lines. Each point in the graph is a stimulus in the Fig.5.14. We can not handle the last two challenging cases in current framework. Both containers acquire human intervention to open containers and put in objects, which can not be modeled solely by the rigid-body dynamics.

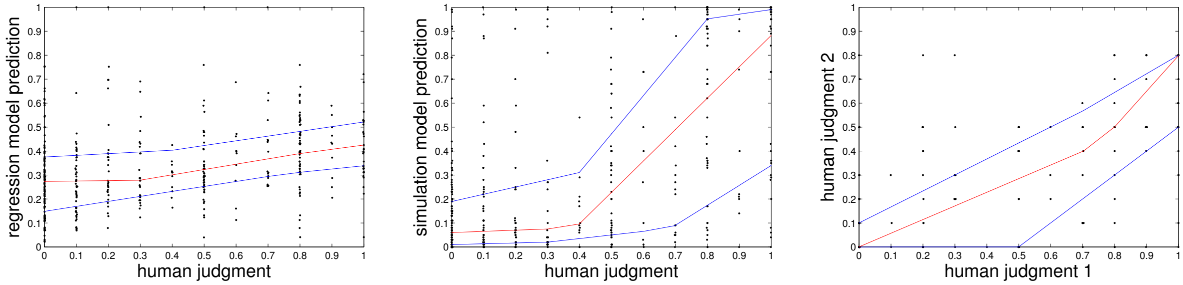


Figure 5.12: Will an object contain another? The left and middle figures show predictions of the regression model and the simulation model with respect to the human judgments. The right figure shows the human judgments of two different subjects. Each data point represents a stimulus with a pair of objects in Fig.5.14. The lower blue line, red line, and upper blue line outline the first quartile (25th percentile), second quartile (median), and third quartile (75th percentile) of the distribution respectively.

Exp. 3: How many objects will a container hold?

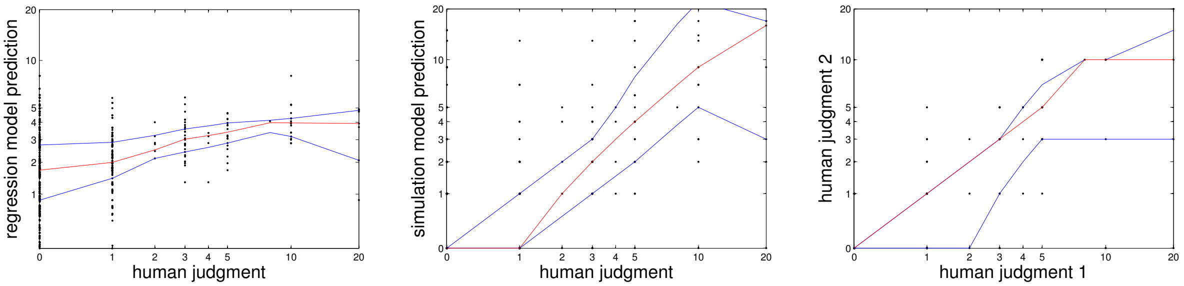


Figure 5.13: How many objects will a container hold? The left and middle figures show predictions of the regression model and the simulation model with respect to the human judgments. The right figure shows the human judgments of two different subjects. Each data point represents a stimulus with a pair of objects in Fig.5.15. The lower blue line, red line, and upper blue line outline the first quartile (25th percentile), second quartile (median), and third quartile (75th percentile) respectively.

In this experiment, the stimuli are the same as Exp.2's. The subjects are shown two random 3D objects and ask "how many objects will a container hold?" The qualitative results and quantitative results are shown in Fig.5.13 and Fig.5.15. Similarly, the simulation results are more consistent with human judgments than the regression model. Although the results exhibit a large variation, similar variations are also existed among judgments from different subjects.

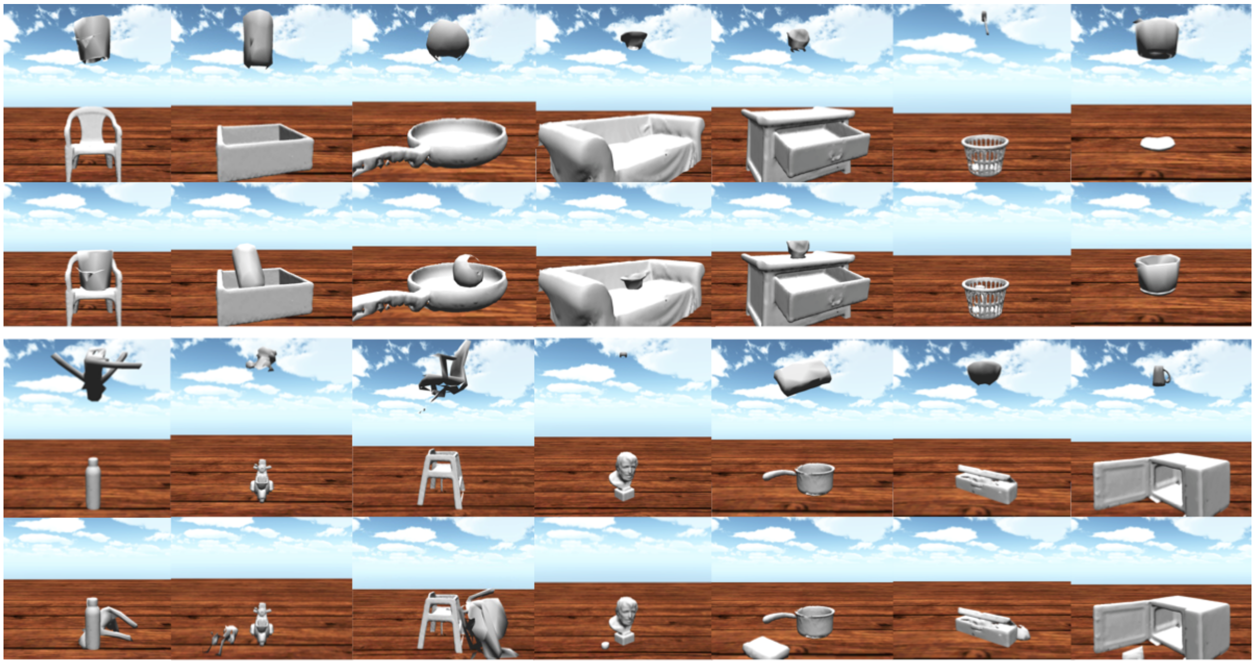


Figure 5.14: More qualitative results for the Exp.2 ”Will an object contain another?”. The first row and third row are screenshots before simulation as stimuli of our experiment. The second row and fourth row show successful containing cases and failed containing cases after physical simulation respectively.

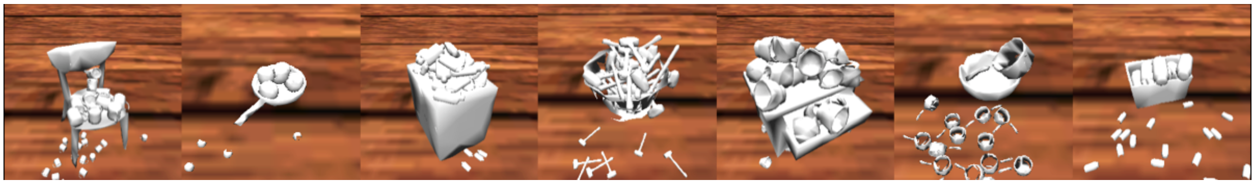


Figure 5.15: Qualitative results after physical simulation for the Exp.3 ”How many objects will a container hold?”.

CHAPTER 6

Conclusion

In this thesis, I construct models of visual commonsense knowledge by bridging advances from statistical learning, computer vision, and cognitive science. My research addresses following challenges:

(i) What is the visual representation?

(ii) How to reason about the commonsense knowledge?

(iii) How to acquire commonsense knowledge?

The main contribution of this paper can be also summarized as follows:

Topic I: Stochastic Grammar: a representation of visual knowledge

Automatic 3D object recognition and reconstruction from a single image are two fundamental problems in Computer Vision, yet they are ill-posed. It is almost impossible to reconstruct a precise 3D scene from a single 2D image. Interestingly, human can recognize 3D structure of a scene and identify object depth effortlessly from a single image.

In this problem, I conjecture that human infer 3D with some basic commonsense knowledge, such as most of the objects are placed on the ground, buildings are mostly straightly aligned with the gravity axis with the consideration of stability, and most of the man-made structures are composed by some shape primitives, like cuboids.

We model the 3D structure of a scene with a Stochastic Scene Grammar, in which 2D/3D structures are stochastically and hierarchically organized like sentences and phrases in a language. From an input 2D image, we will extract line segments, and group them into larger structures, e.g. 3D rectangular bounding boxes and 3D coordinate frame of the scene, by applying production rules iteratively. An MCMC algorithm searches the most plausible explanation with top-down and bottom-up moves. Our preliminary results achieved state-of the-art results in public benchmarks for both indoors and outdoors.

Topic II: Functionality: a basic property that defines man-made scenes / objects

Secondly, I study scenes as functional spaces. A large portion of vision literature studies scenes as a classification problem based on their appearance. I argue that this is ill-posed for two reasons: i) most scenes, especially indoor living spaces, are defined by their functions; and ii) a space may serve multiple functions, and thus cannot be simply classified in one category.

We pose the problem as scene parsing, which integrates all three aspects: function, geometry and appearance. This study is based on two major observations:

i) Functionality is a fundamental property to define an indoor object, e.g. ”a chair is for sitting”. The actions, such as sitting and sleeping, provide the top-down constraints about 3D contexts, and thus disambiguate the appearance uncertainty. The inferred plausible actions define the functions of the space.

ii) Physical sizes of furniture are designed to serve its function, and thus can be fitted to the 3D functional relations learned from data.

On the basis of the Stochastic Scene Grammar discussed before, we augment functional labels and contextual relations on top. The functional relation is defined as a 4D human-objects interaction mentioned before. The functional prior knowledge is learned from a dataset of indoor functional objects from the Trimble 3D Warehouse.

Topic III: Physics: a general principle that underlies our physical world

I also study physics for scene understanding by reasoning about physical stability and safety. We define physical stability and safety under two assumptions:

i) Stability assumption: objects in a static scene should be stable with respect to the gravity field. In other words, if any object is not stable on its own, it must be either grouped with neighbors or fixed to its supporting base. This assumption is applicable to an entire human living space and poses useful constraints on the physically plausible interpretations of visual scenes.

ii) Safety assumption: Even if an object is stable with gravity, it may not be safe with respect to human activities or other disturbances, e.g., wind or earthquake. Such safety analysis ability is important for living beings. In order to give such an ability to a machine, we propose another algorithm for detecting potential falling objects, i.e. physically unsafe objects. Our approach first infers hidden and situated causes of the scene, and then introduces intuitive physical mechanics to predict possible effects. This kind of safety aware algorithm is useful for human assistant robot and disaster rescue robot, as well as driverless cars. Our work is among the first to study scene physics.

Topic IV: Acquisition of commonsense knowledge from video

(i) Learning Affordance from Visual Observation

Action recognition and object detection are challenging in the context of affordance space, which we characterized by a 4D human-object interaction. The 4D human-object interaction has two folds: i) in 1D time, the interactions are presented as atomic event transitions and object coherences; ii) in 3D space, the human and objects interact in the form of semantic co-occurrence and geometric compatibility. We proposed a Stochastic Action Grammar to model events and objects in RGBD videos embedded in a spatially-temporally interactive space. A basic phrase of the grammar is defined by a human-object interaction, i.e. a SVO (subject-verb-object) structure, and multiple phrases can stochastically form larger structures, like sentences, for representing more complex events. Given an input RGBD video, our algorithm jointly: i) recognizes events, ii) segments video sequences, and iii) localizes objects in 3D point cloud and 2D image for each video frame.

(ii) Learning Tool Use from Single Demonstrations:

We present a computational framework for learning tool use from single demonstration. Instead of casting it as a classification or detection problem, we consider tool use as a task-oriented vision problem. We define a tool as a device or an implement used to achieve a goal or carry out a particular function, and represent the tool use in a spatial-temporal-causal space. The learning process takes spatial-temporal parsing results as input, and analyzes functional and physical properties, as well as causal and effect relations of the action. As a result, a robot can learn some intuitive knowledge, such as heavy tools can be used for smashing and sharp tools can be used for cutting, from single human demonstration. The knowledge is then used to identify possible tools in a novel situation.

(iii) Inferring Containing Relations by Physical Simulation

We study a special category of objects “container”. We built a physical simulation system using Unity 3D to infer the “affordance” of containers and containing relations between objects. In the experiment, compared with using regression model of geometric features, the results by physical simulation have stronger correlations with human judgments. We conclude that the physical simulation is a good approximation of human cognition of container and containing relations.

The physical model of the 3D scene quantitatively encodes a large number of static and dynamic variables needed to capture the interactions among objects. These variables include scene configurations, object geometries, masses, material properties, rigidity, fragileness, frictions, collisions, etc. We take advantages of the state-of-the-art 3D scanning technique, which enables us to analyze real-world 3D objects in a physical realistic environment.

REFERENCES

[AFS06] Marco Attene, Bianca Falcidieno, and Michela Spagnuolo. "Hierarchical mesh segmentation based on fitting primitives." *THE VISUAL COMPUTER*, **22**:181–193, 2006.

[AKJ12] Abhishek Anand, Hema Koppula, Thorsten Joachims, and Ashutosh Saxena. "Contextually Guided Semantic Labeling and Search for 3D Point Clouds." *IJRR*, 2012.

[AME14] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. "Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models." In *CVPR*, 2014.

[BHT13] P. Battaglia, J. Hamrick, and J. Tenenbaum. "Simulation as an engine of physical scene understanding." *PNAS*, **110**(45):18327–18332, 2013.

[Bis06] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[BLC00] M. Blane, Z. B. Lei, and D. B. Cooper. "The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data." *IEEE Trans. on Patt. Anal. Mach. Intell (TPAMI)*, **22**(3):298–313, 2000.

[BMR82] Irving Biederman, Robert J. Mezzanotte, and Jan C. Rabinowitz. "Scene perception: Detecting and judging objects undergoing relational violations." *Cognitive Psychology*, **14**(2):143 – 177, 1982.

[BR06] Ezer Bar-aviv and Ehud Rivlin. "Functional 3D Object Classification Using Simulation of Embodied Agent." In *BMVC*, 2006.

[BZ05] A. Barbu and S. C. Zhu. "Generalizing Swendsen-wang to Sampling Arbitrary Posterior Probabilities." *IEEE Trans. on Patt. Anal. Mach. Intell (TPAMI)*, **27**:1239–1253, 2005.

[CCP13] W. Choi, Y. Chao, C. Pantofaru, and S. Savarese. "Understanding Indoor Scenes using 3D Geometric Phrases." In *CVPR*, 2013.

[CDR99] R. Cipolla, T. Drummond, and D. Robertson. "Camera calibration from vanishing points in images of architectural scenes." In *BMVC*, 1999.

[CGF09] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. "A Benchmark for 3D Mesh Segmentation." In *SIGGRAPH*, 2009.

[CLT10] Myung J. Choi, Joseph J. Lim, Antonio Torralba, and Alan S. Willsky. "Exploiting Hierarchical Context on a Large Database of Object Categories." In *CVPR*, 2010.

[CRZ00] A. Criminisi, I. Reid, and A. Zisserman. "Single View Metrology." *International Journal of Computer Vision (IJCV)*, **40**(2):123–148, November 2000.

[CY03] J.M. Coughlan and A.L. Yuille. "Manhattan World: Orientation and Outlier Detection by Bayesian Inference." *Neural Computation*, **15**(5):1063–1088, 2003.

[DAR14] DARPA. "Robots Rescue People." `http://www.i-programmer.info/news/169-robotics/6857-robots-rescue-people.html`, 2014.

[DBF12] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. "Bayesian geometric modeling of indoor scenes." In *CVPR*, pp. 2719–2726, 2012.

[DBK13] L. Del Pero, Joshua Bowdish, Bonnie Kermgard, Emily Hartley, and Kobus Barnard. "Understanding Bayesian rooms using composite 3D object models." In *CVPR*, 2013.

[DFL12] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros. "Scene semantics from long-term observation of people." In *ECCV*, 2012.

[DGB11] L. Del Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard. "Sampling bedrooms." In *CVPR*, 2011.

[DLN07] E. Delage, H. Lee, and A. Ng. "Automatic single-image 3d reconstructions of indoor manhattan world scenes." *Robotics Research*, p. 305321, 2007.

[DR13] C. Desai and D. Ramanan. "Predicting Functional Regions of Objects." In *CVPR Workshop on Scene Analysis Beyond Semantics*, 2013.

[DT05] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection." 2005.

[EBK12] E.Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. "Geometric Image Parsing in Man-Made Environments." *IJCV*, **97**(3):305–321, 2012.

[FBB10] RW Fleming, M Barnett-Cowan, and HH Bülthoff. "Perceived object stability is affected by the internal representation of gravity." *Perception*, **39**:109, 8 2010.

[FCS09] Y. Furukawa, B. Curless, S. M Seitz, and R. Szeliski. "Manhattan-World Stereo." In *CVPR*, 2009.

[FDG12] David F. Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A. Efros, Ivan Laptev, and Josef Sivic. "People Watching: Human Actions as a Cue for Single-View Geometry." In *ECCV*, 2012.

[FDU12] Sanja Fidler, Sven Dickinson, and Raquel Urtasun. "3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model." In *NIPS*, 2012.

[FGH14] David F. Fouhey, Abhinav Gupta, and Martial Hebert. "Unfolding an Indoor Origami World." In *ECCV*, 2014.

[FH04] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation." *Int. J. Comput. Vision*, **59**(2):167–181, 2004.

[Fu82] K. S. Fu. "Syntactic Pattern Recognition and Applications." *Prentice-Hall*, 1982.

[GEH10a] A. Gupta, A. Efros, and M. Hebert. "Blocks World Revisited: Image Understanding using Qualitative Geometry and Mechanics." In *ECCV*, 2010.

[GEH10b] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. "Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics." In *European Conference on Computer Vision(ECCV)*, 2010.

[GGG11] Helmut Grabner, Juergen Gall, and Luc Van Gool. "What Makes a Chair a Chair?" In *CVPR*, 2011.

[GH13] Ruiqi Guo and Derek Hoiem. "Support Surface Prediction in Indoor Scenes." In *ICCV*, 2013.

[Gib77] James J . Gibson. *The Theory of Affordances*. Lawrence Erlbaum, 1977.

[GKD09] A. Gupta, A. Kembhavi, and L.S. Davis. "Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition." *IEEE TPAMI*, **31**(10), 2009.

[GSE11] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. "From 3D scene geometry to human workspace." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1961–1968, Washington, DC, USA, 2011. IEEE Computer Society.

[GT14] N. D. Goodman and J. B. Tenenbaum. *(electronic) Probabilistic Models of Cognition*. http://probmods.org, 2014.

[GZW07] C.E. Guo, S.C. Zhu, and Y.N. Wu. "Primal Sketch: Integrating Texture and Structure." *Computer Vision and Image Understanding*, **106**(1):5–19, 2007.

[HBT11] J. Hamrick, PW. Battaglia, and J.B. Tenenbaum. "Internal physics models guide probabilistic judgments about object dynamics." In *Conf. Cog. Sc.*, 2011.

[HEH] D. Hoiem, A. Efros, and M. Hebert. "Geometric context from a single image.".

[HEH07] D. Hoiem, A. Efors, and M. Hebert. "Recovering Surface Layout from an Image." *IJCV*, **75**(1), 2007.

[HEH09] D. Hoiem, A. Efros, and M. Hebert. "Automatic Photo Pop-up." *TOG*, **31**(1):59–73, 2009.

[HEH10] D. Hoiem, A. Efros, and M. Hebert. "Closing the loop on scene interpretation." In *ECCV*, 2010.

[HHF09] V. Hedau, D. Hoiem, and D. Forsyth. "Recovering the spatial layout of cluttered rooms." In *ICCV*, 2009.

[HHF10] V. Hedau, D. Hoiem, and D. Forsyth. "Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry." In *ECCV*, 2010.

[HHF12] Varsha Hedau, Derek Hoiem, and David Forsyth. "Recovering Free Space of Indoor Scenes from a Single Image." In *CVPR*, 2012.

[HR12] Mohsen Hejrati and Deva Ramanan. "Analyzing 3D Objects in Cluttered Images." In *NIPS*, pp. 602–610, 2012.

[Hu12] Wenze Hu. "Learning 3D object templates by hierarchical quantization of geometry and appearance spaces." In *CVPR*, pp. 2336–2343, 2012.

[HZ04a] Feng Han and Song-Chun Zhu. "Bayesian Reconstruction of 3D Shapes and Scenes From A Single Image." In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision*, 2004.

[HZ04b] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[HZ09] F. Han and S. C. Zhu. "Bottom-Up/Top-Down Image Parsing with Attribute Grammar." *PAMI*, 2009.

[IL13] P. Isola and C. Liu. "Scene collaging: analysis and synthesis of natural images with semantic layers." In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[JGS13] Z. Jia, A. Gallagher, A. Saxena, and T. Chen. "3D-Based Reasoning with Blocks, Support, and Stability." In *CVPR*, 2013.

[JKJ11] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T. Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. "A Category-Level 3-D Object Dataset: Putting the Kinect to Work." In *ICCV Workshop*, 2011.

[JKS13] Yun Jiang, Hema Koppula, and Ashutosh Saxena. "Hallucinated Humans as the Hidden Context for Labeling 3D Scenes." In *CVPR*, 2013.

[JS13] Y Jiang and A. Saxena. "Infinite Latent Conditional Random Fields for Modeling Environments through Humans." In *In Robotics: Science and Systems (RSS)*, 2013.

[KAJ11] H.S. Koppula, A. Anand, T. Joachims, and A. Saxena. "Semantic Labeling of 3D Point Clouds for Indoor Scenes." In *NIPS*, 2011.

[KCG14] Vladimir G. Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. "Shape2Pose: Human-Centric Shape Analysis." In *SIGGRAPH*, 2014.

[KF12] Xiaofeng Ren Kevin Lai, Liefeng Bo and Dieter Fox. "Detection-based Object Labeling in 3D Scenes." In *ICRA*, 2012.

[KMF13] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. "Object Discovery in 3D Scenes via Shape Analysis." In *International Conference on Robotics and Automation (ICRA)*, 2013.

[Kri95] David J. Kriegman. "Let Them Fall Where They May: Capture Regions of Curved Objects and Polyhedra." *International Journal of Robotics Research*, **16**:448–472, 1995.

[KS13] Hema Koppula and Ashutosh Saxena. "Anticipating Human Activities using Object Affordances for Reactive Robotic Response." In *RSS*, 2013.

[KS14] Hema Koppula and Ashutosh Saxena. "Physically-Grounded Spatio-Temporal Object Affordances." In *ECCV*, 2014.

[LCK14] Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G. Kim, Qi-Xing Huang, Niloy J. Mitra, and Thomas Funkhouser. "Creating Consistent Scene Graphs Using a Probabilistic Grammar." In *SIGGRAPH*, 2014.

[LFU13] Dahua Lin, Sanja Fidler, and Raquel Urtasun. "Holistic Scene Understanding for 3D Object Detection with RGBD cameras." In *ICCV*, 2013.

[LGH10] D. Lee, A. Gupta, M. Hebert, and T. Kanade. "Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces Advances in Neural Information Processing Systems." *Cambridge: MIT Press*, pp. 609–616, 2010.

[LHK09] D. Lee, M. Hebert, and T. Kanade. "Geometric Reasoning for Single Image Structure Recovery." In *CVPR*, 2009.

[LKT14] Joseph Lim, Aditya Khosla, and Antonio Torralba. "FPM: Fine pose Parts-based Model with 3D CAD models." In *ECCV*, 2014.

[LMP01] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. "Conditional random fields: probabilistic models for segmenting and labeling sequence data." *ICML*, pp. 282–289, 2001.

[LN06] Fengjun Lv and Ramakant Nevatia. "Recognition and segmentation of 3-d human action using HMM and multi-class adaboost." In *ECCV*, 2006.

[LPT13] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. "Parsing IKEA Objects: Fine Pose Estimation." In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[LYT11] C. Liu, J. Yuen, and A. Torralba. "Nonparametric scene parsing via label transfer." *IEEE Trans. on Patt. Anal. Mach. Intell (TPAMI)*, 2011.

[McC83] M. McCloskey. "Intuitive physics." *Scientific American*, **248**(4):114–122, 1983.

[MKP13] V. Mansinghka, T. Kulkarni, Y. Perov, and J. Tenenbaum. "Approximate Bayesian image interpretation using generative probabilistic graphics programs." In *NIPS*, 2013.

[MR06] Meinard Müller and Tido Röder. "Motion templates for automatic classification and retrieval of motion capture data." In *ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2006.

[MZY11] Hossein Mobahi, Zihan Zhou, Allen Y. Yang, and Yi Ma. "Holistic 3D Reconstruction of Urban Structures from Low-rank Textures." In *Proceedings of the International Conference on Computer Vision - 3D Representation and Recognition Workshop*, pp. 593–600, 2011.

[NF12] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. "Indoor Segmentation and Support Inference from RGBD Images." In *ECCV*, 2012.

[NIH11] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-Time Dense Surface Mapping and Tracking." In *ISMAR*, 2011.

[NXS12] Liangliang Nan, Ke Xie, and Andrei Sharf. "A Search-classify Approach for Cluttered Indoor Scene Understanding." *ACM Trans. on Graph. (TOG)*, **31**(6), 2012.

[OM08] L. Oakes and K. Madole. "Function revisited: How infants construe functional features in their representation of objects." *Advances in Child Development and Behavior*, **36**:135185, 2008.

[OT01] A. Oliva and A. Torralba. "Modeling the shape of the scene: a holistic representation of the spatial envelope." *International Journal of Computer Vision (IJCV)*, 2001.

[PF05] S. Petti and T. Fraichard. "Safe Motion Planning in Dynamic Environments." In *IROS*, 2005.

[PGS12] Bojan Pepik, Peter Gehler, Michael Stark, and Bernt Schiele. "3D2PM - 3D Deformable Part Models." In *ECCV*, Firenze, Italy, 2012.

[PJZ11] Mingtao Pei, Yunde Jia, and Song-Chun Zhu. "Parsing video events with goal inference and intent prediction." In *ICCV*, 2011.

[PL11] M. Phillips and M. Likhachev. "SIPP: Safe Interval Path Planning for Dynamic Environments." In *ICRA*, 2011.

[Pla99] John C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods." In *Advances in large margin classifiers*, 1999.

[POF12] S. N. Parizi, J. Oberlin, and P. Felzenszwalb. "Reconfigurable models for scene recognition." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[PSF12] A. Prest, C. Schmid, and V. Ferrari. "Weakly Supervised Learning of Interactions between Humans and Objects." *TPAMI*, **34**(3), 2012.

[PT11] Nadia Payet and Sinisa Todorovic. "Scene Shape from Textures of Objects." In *CVPR*, 2011.

[PVB08] Jann Poppinga, Narunas Vaskevicius, Andreas Birk, and Kaustubh Pathak. "Fast plane detection and polygonalization in noisy 3D range images." In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[PZ10] J. Porway and S. C. Zhu. "Hierarchical and Contextual Model for Aerial Image Understanding." *IJCV*, **88**(2):254–283, 2010.

[RM03] X. Ren and J. Malik. "Learning a classification model for segmentation." In *ICCV*, 2003.

[SB91] L. Stark and K. Bowyer. "Achieving generalized object recognition through reasoning about association of function to structure." *PAMI*, **13**:10971104, 1991.

[SCH14] Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. "SceneGrok: Inferring Action Maps in 3D Environments." *ACM Trans. on Graph. (TOG)*, **33**(6), 2014.

[SF83] Q. Y. Shi and King-sun Fu. "Parsing and Translation of (Attributed) Expansive Graph Languages for Scene Analysis." *TPAMI*, **PAMI-5**(5):472–485, 1983.

[SFC11] Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. "Real-time human pose recognition in parts from single depth images." In *CVPR*, 2011.

[SFP13] Alexander G. Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. "Box In the Box: Joint 3D Layout and Object Reasoning from Single Images." In *ICCV*, 2013.

[SH13] Scott Satkin and Martial Hebert. "3DNN: Viewpoint Invariant 3D Geometry Matching for Scene Understanding." In *ICCV*, 2013.

[SHP12] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun. "Efficient Structured Prediction for 3D Indoor Scene Understanding." In *CVPR*, 2012.

[SLH12] Scott Satkin, Jason Lin, and Martial Hebert. "Data-Driven Scene Understanding from 3D Models." In *BMVC*, 2012.

[SM00] Jianbo Shi and Jitendra Malik. "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8):888–905, 2000.

[SMZ14] Tianjia Shao, Aron Monszpart, Youyi Zheng, Bongjin Koo, Weiwei Xu, Kun Zhou, and Niloy J. Mitra. "Imagining the Unseen: Stability-based Cuboid Arrangements for Scene Understanding." *ACM Trans. on Graph. (TOG)*, 2014.

[SNI05] Ryusuke Sagawa, Ko Nishino, and Katsushi Ikeuchi. "Adaptively Merging Large-Scale Range Data with Reflectance Properties." *IEEE Trans. on Patt. Anal. Mach. Intell (TPAMI)*, **27**:392–405, 2005.

[SSN09] A. Saxena, M. Sun, and A. Ng. "Make3D: Learning 3D Scene Structure from a Single Still Image." *PAMI*, **31**(5):824–840, 2009.

[SU12] Alexander G. Schwing and Raquel Urtasun. "Efficient Exact Inference for 3D Indoor Scene Understanding." In *ECCV*, 2012.

[SX14] S. Song and J. Xiao. “Sliding Shapes for 3D Object Detection in Depth Images.” In *ECCV*, 2014.

[SXZ12] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. “An Interactive Approach to Semantic Modeling of Indoor Scenes with an RGBD Camera.” *ACM Trans. on Graph. (TOG)*, **31**(6), 2012.

[TCY05] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-Chun Zhu. “Image Parsing: Unifying Segmentation, Detection, and Recognition.” *Int. J. Computer Vision (IJCV)*, 2005.

[TKG11] J. Tenenbaum, C. Kemp, T. Griffiths, and N. Goodman. “How to grow a mind: Statistics, structure, and abstraction.” *Science*, **331**(6022):1279–1285, 2011.

[TL13a] J. Tighe and S. Lazebnik. “Finding things: image pars- ing with regions and per-exemplar detectors.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[TL13b] J. Tighe and S. Lazebnik. “Superparsing: scalable non- parametric image parsing with superpixels.” *International Journal of Computer Vision (IJCV)*, 2013.

[TZ02] Zhuowen Tu and Song-Chun Zhu. “Image Segmentation by Data-Driven Markov Chain Monte Carlo.” *PAMI*, **24**(5):657–673, 2002.

[VJM10] R.G. Von Gioi, J. Jakubowicz, J. M Morel, and G. Randall. “LSD: A Fast Line Segment Detector with a False Detection Control.” *TPAMI*, **32**(4):722–732, 2010.

[Wal04] D. Wales. *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*. Cambridge Molecular Science. Cambridge University Press, 2004.

[WGK10] Huayan Wang, Stephen Gould, and Daphne Koller. “Discriminative learning with latent variables for cluttered indoor scene understanding.” In *ECCV*, pp. 497–510, 2010.

[WLS14] Chenxia Wu, Ian Lenz, and Ashutosh Saxena. “Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception.” In *Robotics: Science and Systems (RSS)*, 2014.

[WWZ12] S. Wang, Y. Wang, and S.C. Zhu. “Hierarchical Space Tiling in Scene Modeling.” In *Asian Conf. on Computer Vision (ACCV)*, 2012.

[WZ11] Tianfu Wu and Song Chun Zhu. “A Numerical Study of the Bottom-Up and Top-Down Inference Processes in And-Or Graphs.” *IJCV*, **93**(2):226–252, 2011.

[WZZ13] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. “Modeling 4D Human-Object Interactions for Event and Object Recognition.” In *ICCV*, 2013.

[XHE10] Jianxiong Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. “SUN database: Large-scale scene recognition from abbey to zoo.” In *CVPR*, pp. 3485 –3492, 2010.

[XRT12] Jianxiong Xiao, Bryan Russell, and Antonio Torralba. “Localizing 3D cuboids in single-view images.” In *NIPS*, pp. 755–763, 2012.

[XS12] Yu Xiang and Silvio Savarese. "Estimating the Aspect Layout of Object Categories." In *CVPR*, 2012.

[YF12] Bangpeng Yao and Li Fei-Fei. "Recognizing Human-Object Interactions in Still Images by Modeling the Mutual Context of Objects and Human Poses." *TPAMI*, **34**(9), 2012.

[YMF13] B. Yao, J. Ma, and L. Fei-Fei. "Discovering Object Functionality." In *ICCV*, 2013.

[YYW12] Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D. Goodman, and Pat Hanrahan. "Synthesizing open worlds with constraints using locally annealed reversible jump MCMC." *ACM Trans. Graph.*, **31**(4):56:1–56:11, July 2012.

[ZM07] S. C. Zhu and D. Mumford. "A stochastic grammar of images." *Foundations and Trends in Computer Graphics and Vision*, **2**(4):259–362, 2007.

[ZST14] Y. Zhang, S. Song, P. Tan, and J. Xiao. "PanoContext: A Whole-room 3D Context Model for Panoramic Scene Understanding." In *ECCV*, 2014.

[ZTH13] Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. "Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(3):582–596, 2013.

[ZTI10] B. Zheng, J. Takamatsu, and K. Ikeuchi. "An Adaptive and Stable Method for Fitting Implicit Polynomial Curves and Surfaces." *PAMI*, **32**(3):561–568, 2010.

[ZWM98] Song Chun Zhu, Yingnian Wu, and David Mumford. "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling." *IJCV*, **27**(2):107–126, 1998.

[ZZ11] Yibiao Zhao and Song-Chun Zhu. "Image Parsing via Stochastic Scene Grammar." In *NIPS*, 2011.

[ZZ13] Yibiao Zhao and Song-Chun Zhu. "Scene Parsing by Integrating Function, Geometry and Appearance Models." In *CVPR*, 2013.

[ZZY13] Bo Zheng, Yibiao Zhao, Joey C. Yu, K. Ikeuchi, and Song-Chun Zhu. "Beyond Point Clouds: Scene Understanding by Reasoning Geometry and Physics." In *CVPR*, 2013.

[ZZY14] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S. C. Zhu. "Detecting Potential Falling Objects by Inferring Human Action and Natural Disturbance." In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[ZZY15] Bo Zheng, Yibiao Zhao, Joey C. Yu, K. Ikeuchi, and Song-Chun Zhu. "Scene Understanding by Reasoning Stability and Safety." *IJCV*, 2015.