Single-View 3D Scene Reconstruction and Parsing by Attribute Grammar

Xiaobai Liu, Yibiao Zhao, *Member, IEEE*, and Song-Chun Zhu, *Fellow, IEEE*

Abstract—In this paper, we present an attribute grammar for solving two coupled tasks: i) parsing a 2D image into semantic regions; and ii) recovering the 3D scene structures of all regions. The proposed grammar consists of a set of production rules, each describing a kind of spatial relation between planar surfaces in 3D scenes. These production rules are used to decompose an input image into a hierarchical parse graph representation where each graph node indicates a planar surface or a composite surface. Different from other stochastic image grammars, the proposed grammar augments each graph node with a set of attribute variables to depict scene-level *global* geometry, e.g., camera focal length, or *local* geometry, e.g., surface normal, contact lines between surfaces. These geometric attributes impose constraints between a node and its off-springs in the parse graph. Under a probabilistic framework, we develop a Markov Chain Monte Carlo method to construct a parse graph that optimizes the 2D image recognition and 3D scene reconstruction purposes simultaneously. We evaluated our method on both public benchmarks and newly collected datasets. Experiments demonstrate that the proposed method is capable of achieving state-of-the-art scene reconstruction of a single image.

Index Terms—3D scene reconstruction, region partition, scene parsing, attribute grammar

1 Introduction

The goal of computer vision, as coined by Marr [32], is to compute *what* and *where*, which correspond to the tasks of recognition and reconstruction respectively. The former is often posed as parsing an image in a hierarchical representation, e.g., from sketches, semantic regions, objects, to scene categories. The latter recovers 3D scene structures, including camera parameters [55], surface normal orientations and depth [21], and local Manhattan world [6]. While the recognition and reconstruction problems are usually addressed separately or sequentially in the literature, it is mutually beneficial to solve them jointly in a tightly coupled framework for two reasons.

- 2D image recognition is capable of providing semantic contextual knowledge for pruning the uncertainties during 3D reconstruction. For example, if two neighboring pixels are classified as different labels, (e.g., ground and building), it is unlikely that they are projections of the same 3D plane. In addition, semantic region labels, e.g., building or ground-plane, often provide strong prior on surface normal.
- 3D scene models can provide additional geometric information to boost image recognition task. In the literature, there have been a number of efforts that utilize scene geometry to help region segmentation [16], [31], objection detection [21], visual tracking [39] or event classification [49], etc.

To couple the above two tasks, we propose an attribute grammar as a unified representation, which augments levels of geometric attributes (e.g., camera parameters, vanishing points, surface normal etc.) to the nodes in the parse graph. Thus the recognition and reconstruction tasks are solved in a joint parsing process simultaneously. Fig. 1 shows a typical result of the proposed approach.

1.1 Overview of Our Approach

We consider outdoor urban scenes that may contain multiple local Manhattan worlds (LMW) or 'mixture Manhattan world' [45], where, for example, buildings are composed of multiple planar surfaces and touch the ground on contact lines. In contrast to the widely used Manhattan world assumption [6], this paper considers more general scenarios where adjacent surfaces of a building may not be orthogonal to each other (see the main building in Fig. 1). Curved surfaces are approximated by piecewise linear splines. The surface is further decomposed into super-pixels and edge elements. These representational units can be naturally organized in a hierarchical parse graph with the root node being the scene and terminal nodes being the edges and super-pixels. Fig. 2 illustrates a parse graph.

Different from the widely studied appearance attributes of scenes in the vision literature [48] [49] [56], we aim to model *geometric attributes* of all types of scene entities or graph nodes in the parse graph. An edge segment has its associated vanishing point, and a super-pixel has a surface normal, a planar facet of a building has two vanishing points and a surface normal, and a building has 3 vanishing points, and finally the whole scene shares a set of camera

- *X. Liu is with San Diego State University (SDSU), San Diego, CA 92182. E-mail: xiaobai.liu@mail.sdsu.edu.*
- *Y. Zhao is with the Massachusetts Institute of Technology (MIT), Cambridge, MA 02139. E-mail: ybzhao@ucla.edu.*
- *S.-C. Zhu is with the University of California at Los Angeles (UCLA), Los Angeles, CA 90095. E-mail: sczhu@stat.ucla.edu.*

Manuscript received 13 Mar. 2015; revised 31 Aug. 2016; accepted 18 Dec. 2016. Date of publication 28 Mar. 2017; date of current version 13 Feb. 2018. Recommended for acceptance by S. Roth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2017.2689007

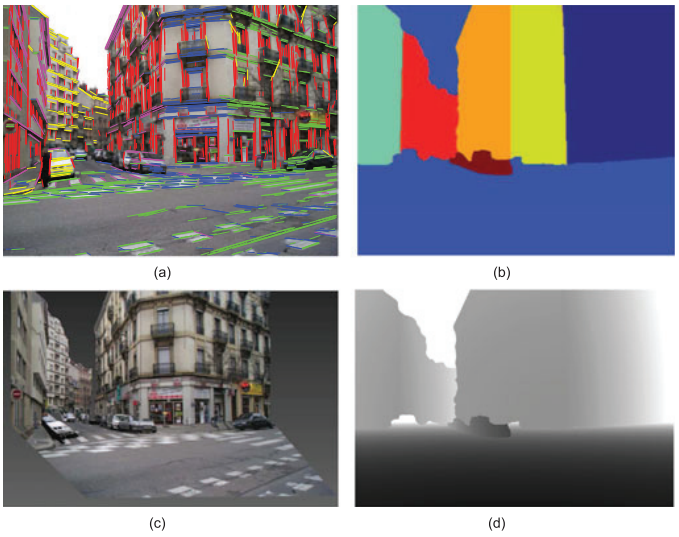


Fig. 1. A typical result of our approach. (a) Input image overlaid with parallel lines, where each color indicates a family of parallel lines; (b) surface normal map, where each color indicates a unique normal orientation; (c) a synthesized image from a novel viewpoint; and (d) depth map, where darker pixels are closer to the camera.

parameters (e.g., focal length). We amount these geometric attributes to the parse graph, as shown in Fig. 2. In this attribute parse graph, attributes of a node can be inherited by its offspring, and thus impose geometric constraints in the hierarchy. These constraints are expressed as additional energy terms in the parsing algorithm so as to maintain consistency in the hierarchy. Consequently, the parsing and reconstruction problems are solved in a tightly coupled manner. This attribute parse graph is different from, and can be integrated with, other scene parsing problems, e.g., fine-grained scene classification [48] that uses appearance attributes, e.g., "cast sky", "yellow field", etc.

To construct the attribute parse graph, we define an attribute grammar using a 5-tuple: $\mathbf{G} = (\mathbf{V}_T, \mathbf{V}_N, S, \mathbf{R}, P)$. The set of terminal nodes $\mathbf{V}_T$ include surface fragments or superpixels, the non-terminal nodes $\mathbf{V}_N$ include planar surfaces, composite surfaces, building block and Manhattan world, the root node S indicates the whole scene, and $\mathbf{R}$ is a set of production rules, and P is the probability models associated with the rules. Each node $a \in \mathbf{V}_T$ (or $A \in \mathbf{V}_N$) is associated with a set of geometric attributes.

We observe that a few production rules (or compositions) are capable of explaining most of the outdoor urban scenes. We introduce 5 production rules which are quite generic for urban scenes. Each rule $A \rightarrow A_1, \ldots, A_k$ represents a certain spatial arrangement between the children surfaces $A_1, \ldots, A_k$, and imposes constraints on the attributes of $X(A)$ and $X(A_1), \ldots, X(A_k)$.

These composition rules compete with each other to interpret the input image in a recursive way, which results in a parse graph as a valid interpretation of the scene. The parse graph includes both appearance models for 2D segmentation and geometric models for 3D reconstruction.

We formulate the inference of attribute parse graph from a single image in a probabilistic framework. The state space is the set of feasible attribute parse graphs with large structural variations. To efficiently sample this complex state space, we adopt the Data-Driven Markov Chain Monte Carlo paradigm [47]. In particular, our inference method starts with an initial parse graph constructed by a greedy method, and then simulates a Markov Chain in the state space by a set of diffusion-jump dynamics [2]. During the initialization stage, we utilize a heuristic search procedure for camera calibration, and introduce a belief propagation method to obtain region labeling which leads to an initial parse graph. During the following sampling stage, we introduce five dynamics that are paired with each other to

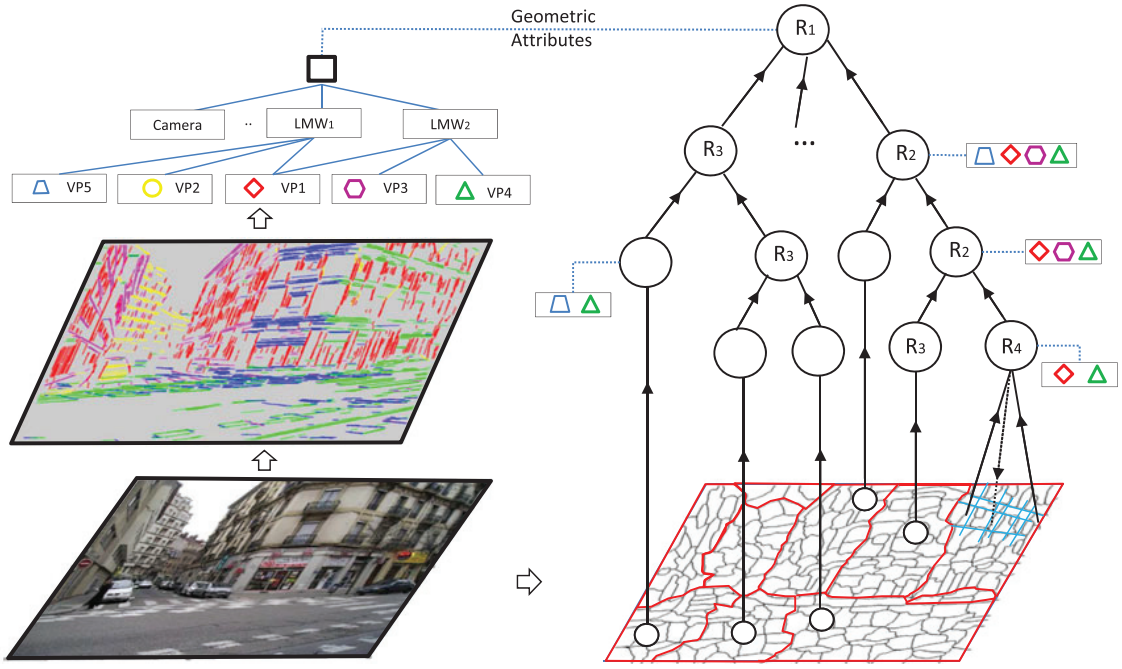


Fig. 2. Parsing an image using attribute grammar. *Left*: *global geometric attributes* are associated with the root node (scene) of the parse graph, including focal length of camera, and Cartesian Coordinate System defined by Manhattan frames. *Right*: parse graph augmented with *local geometric attributes*, such as surface normal orientations and vanishing points (VPs) associated with a surface, or multiple vanishing points for a building. $R_1, \ldots, R_5$ are the five grammar rules used for scene decomposition.

exploit the joint solution space periodically, which can guarantee nearly global convergence [47].

A short version of this work appeared in IEEE CVPR'2014 [31] and we extend it in both modeling and inference. In *modeling*, [31] uses geometric attributes to impose hard constraints that were used to switch on or off the corresponding probability models, whereas this work uses both semantic and geometric attributes to impose soft constraints to define a set of calibrated energies models, resulting in a more flexible model. In *inference*, this work introduces a stage-wise MCMC sampling method which is more effective than the method in [31]. Moreover, we collect and annotate a new dataset comprising of 950 images, and evaluate both methods on it. Results show that the newly proposed method achieves much better performance in terms of convergence and reconstruction/labeling accuracy.

1.2 Related Works

Our work is closely related to the following research streams in computer vision.

Semantic scene labeling has been widely studied to deal with appearance variations, low-resolution and semantic ambiguities. A popular choice is the Conditional Random Field model [27] that describes qualitative contextual relations between region segments. Such relations are proved to be helpful in the recognition of outdoor objects. Choi et al. [5] further studied a 2D context model to guide detectors and produced a semantically coherent interpretation for the given image. Felzenszwalb and Vekser [10] applied the dynamic programming method for pixel labeling of 2D scenes with "tiered" structure. These methods formulate scene labeling as a pixel-wise labeling problem and ignore the hierarchical and recursive composition relations between semantic regions. In contrast, our method introduces a hierarchical parse graph that can explain the input image at different levels, from pixels to regions to scene layout.

Single-view 3D scene modeling has been extensively studied in previous literature. Han and Zhu [16] studied a generative model for reconstructing objects and plants from a single input image. Hoiem et al. [20] explored rich geometric features and context information to recognize normal orientation labels of 2D regions, and Heitz et al. [18] further proposed to recognize geometric surface labels and semantic labels simultaneously in a supervised way. Gupta et al. [13] considered 3D objects as blocks and inferred their 3D properties such as occlusion, exclusion and stability. Mobahi et al. [33] reconstructed a single view by extracting low rank textures on building facades. Saxena et al. [42] and Haene et al. [15] proposed a fully supervised model to build mappings between informative features and depth values. Schwing et al. [44] presented an exact inference method (i.e., branch-and-bound) for single-view indoor scene reconstruction. Pero et al. formulated the 3D reconstruction of room in a Bayesian framework and proposed a sampling method for inference [36], [37], [38]. Ladicky et al. [26] proposed a discriminatively trained boosting method for estimating surface normal. The above mentioned methods tried to recover global 3D scene without an explicit representation of camera model and 3D geometric structures. In contrast, our method introduces an attribute grammar as the unified representation of semantic regions and their geometric attributes, which can provide interpretable decompositions of the input image in both 2D and 3D. More importantly, the hierarchical representation is interpretable and explainable and can be used to facilitate higher level visual perception problems, e.g., object activity recognition.

Joint Recognition and Reconstruction has been investigated for multiple computer vision tasks. Haene et al. [15] presented a continuous-discrete formulation for jointly solving scene reconstruction and labeling of images of multiple views. Ladicky et al. [25] proposed to train a depth-wise classifier for each scene category, used to predict semantic categories and depth maps for a single image. Their method requires groundtruth depth maps for training. Carbral et al. [3] tried to recover planar-wise 3D scene model from panorama images of office areas, which extended the previous works by Xiao et al. [50].

The other studies include jointly solving object recognition and object modeling. Haene et al. [14] proposed to learn 3D shape priors from surface normal orientations, which has been proved to be very successful. Hejrati et al. [19] proposed to synthesize 2D object instances from 3D models and used the instances to help solve object recognition task. Schwing et al. [43] introduced a method for recovering 3D room layout and objects simultaneously. Xiao et al. presented a supervised method for localizing 3D cuboids in 2D images [52]. They also introduced a benchmark [51] for joint structure-from-motion and object labeling. Payet and Todorovic [34] proposed a joint model to recognize objects and estimate scene shape. Zhang et al. [54] proposed to reconstruct a room using panoramic images by exploiting both object parsing (e.g., table detection) and scene geometry (e.g., vanishing points).

Moreover, joint formulation has also been applied for simultaneous tracking and reconstruction [24] [53], joint object recognition and reconstruction [1] [29], floor-plan layout estimation [30] and video reconstruction [24]. Our work follows the same methodology and contributes an attribute grammar for joint image labeling and scene reconstruction. The developed techniques can be applied to the above mentioned joint tasks as well.

Scene Grammar. Koutsourakis et al. [23] proposed a shape grammar to reconstruct building facades, which focused on rectifying facade images but not recovering 3D geometry. Han and Zhu [17], Zhao and Zhu [56] and Pero et al. [35] developed various generative scene grammar models to explore the composition of scene entities of indoor images. Furukawa et al. [11] studied the reconstruction of Manhattan scenes from stereo inputs. In contrast, we relax the Manhattan assumption and generalize the scene grammar model to handle cluttered outdoor scenes. We contribute a hierarchical representation for urban scene modeling and augment it with both semantic and geometric attributes.

In summary, the three contributions of this work include: i) a grammar model with geometric attributes for solving the 2D image parsing and 3D scene reconstruction tasks simultaneously; ii) a stage-wise sampling inference method that is capable of exploiting the constrained space efficiently; iii) state-of-the-art computer vision systems that can

generate high-quality results for both recognition and reconstruction on images from both public datasets and newly collected datasets.

1.3 Paper Organization

The rest of this paper is organized as follows. We introduce a hierarchical scene representation in Section 2, present a probabilistic grammar model in Section 3, and discuss the inference algorithm in Section 4. We report the experimental results in Section 5, and conclude this paper with a discussion in Section 6.

2 Representation: Attribute Hierarchy

Given an input image, our goals include: i) recovering the scene geometry structure, ii) partitioning the image into semantic planar regions, and iii) reconstructing the planar-wise 3D scene model. These goals can be unified as solving the optimal parse graph with geometric and semantic attributes. In this section, we overview the hierarchical representation of scene entities and their attributes.

Camera Parameter We assume that there is no distortion, no skew, and that the camera optical center coincides with the image center. Thus we need to estimate camera focal length f and camera viewing directions. The viewing directions can be described by Manhattan frames since we consider Manhattan type urban scenes. We subtract the image center from the coordinate of each pixel to facilitate representation.

2.1 Geometric Attributes from Edge Statistics

In man-made scenes, texture gradients and edges are not arbitrarily oriented, but reflect camera orientations with respect to the scene and surface layout in 3D world. Hence, we can extract the geometric attributes from edge statistics.

2.1.1 Attributes of Edges and Parallel Lines

In pinhole camera model, a family of parallel lines share the same direction in 3D space and project to straight edges which all point to the same point on the image plane, i.e., the vanishing point. Thus each line segment in images has two geometric attributes:

- A vanishing point (x_i, y_i) on the image plane to which an edge points to. This can be directly obtained by clustering oriented edges based on their 2D directions in images [28].
- A 3D direction. Every family of parallel lines is associated with a vanishing point. As Fig. 3 illustrates, it follows from perspective geometry that a 3D ray from the camera center O to a vanishing point is parallel to the associated family of parallel lines. Therefore, the direction of an edge pointing to the vanishing point (x_i, y_i) is the unit vector by normalizing the triple vector (x_i, y_i, f) where f is the camera focal length.

2.1.2 Attributes of Local Manhattan World

Outdoor urban scene often contains a mixture of local Manhattan worlds [6]. Each local Manhattan world is a block of well aligned buildings with three sets of orthogonal parallel lines. Each set of parallel lines has a vanishing point (x_i, y_i) and 3D direction $\theta = (x_i, y_i, f)$. We refer to the rays from the camera origin O to the vanishing points as the *Manhattan axes*. Thus each local Manhattan world has the following geometric attributes:

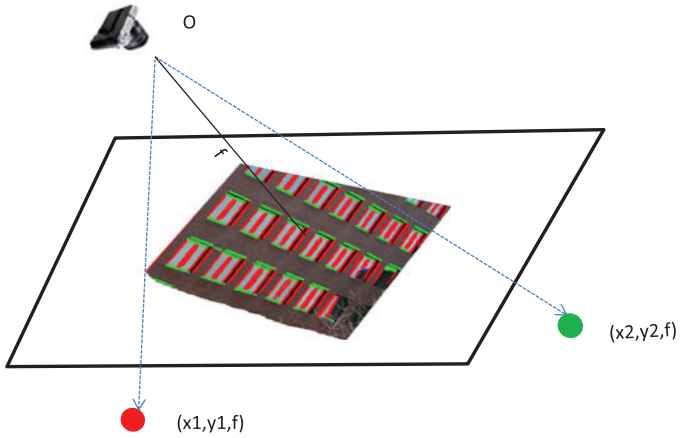


Fig. 3. Illustration of surface normal. A planar region often contains two sets of orthogonal parallel lines converging at two vanishing points (x_1, y_1) and (x_2, y_2), respectively. f is the camera focal length. Thus the surface normal is the cross-product of the two Manhattan axes (x_1, y_1, f) and (x_2, y_2, f) in the camera coordinate (taking camera position O as the origin).

- A Manhattan frame with three orthogonal Manhattan axes $\{(x_1, y_1, f), (x_2, y_2, f), (x_3, y_3, f)\}$.
- An estimated focal length f following [4]

$$f^2 = -(x_i, y_i) \cdot (x_j, y_j), \quad i \neq j \in \{1, 2, 3\}. \tag{1}$$

This follows the orthogonal condition that

$$(x_i, y_i, f) \cdot (x_j, y_j, f) = 0. \tag{2}$$

It is worth noting that the estimated focal length will be propagated to the scene node in the attribute parse graph. The equation $(x_i, y_i) \cdot (x_j, y_j) = (x_k, y_k) \cdot (x_j, y_j)$ poses consistency conditions among the attributes of Manhattan axes.

2.2 Attribute Planar Representation

The region-based hierarchy comprises of three representations: surface fragments, planar surfaces, and composite surfaces. Fig. 4 summarizes the attribute planar representation.

We augment every hierarchical entity with both semantic and geometric attributes. The semantic attribute of an entity, e.g., planar surface, is simply its semantic category. In this work, we consider a few semantic categories for outdoor scenes, including "building", "tree", "ground", "sky" and "other". A composite surface might include two or more categories. The geometric attributes of scene entities are used to describe the spatial properties of the hierarchy, which will be introduced in the rest of this section.

2.2.1 Geometric Attributes of Surface Fragment

We assume that each *super-pixel* in images is the projection of a *surface fragment* in 3D space. A super-pixel is a small region of pixels that are connected and share similar appearance features, and often have the same semantic label. Since these

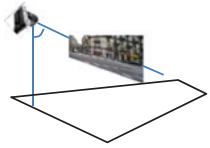
Representation	*Examples*	*Geometric Attributes*
Surface Fragment		Surface Normal; VPs;
Planar Surface		Surface Normal; Contact Line; Manhattan Axes; 2 VPs;
Composite Surface		Surface Normal ; Contact Spline; Manhattan Axes; K VPs;
Scene		Focal Length ; Vanishing Points; Manhattan Frames;

Fig. 4. Illustration of hierarchical scene entities in the attribute planar representation and their associated geometric attributes. Each entity is also entitled with a semantic attribute, e.g., building, sky etc.

super-pixels often correspond to regions in buildings or marked road/highways/ground, which have edges or texture gradients, from which we can extract short edges and estimate which vanishing points they point to.

As Fig. 3 illustrates, each super-pixel has two geometric attributes:

- Two vanishing points: $\{(x_1, y_1), (x_2, y_2)\}$ and thus two Manhattan axes $\{\mathbf{v}_1 = (x_1, y_1, f), \mathbf{v}_2 = (x_2, y_2, f)\}$;
- A surface normal direction which is the cross-product of the two Manhattan axes $\mathbf{n} = (x_1, y_1, f) \times (x_2, y_2, f)$.

For each superpixel, we extract its vanishing points and surface normal from local edge statistics, which might not be necessarily accurate. To improve robustness against noises, these statistics will be pooled together in a bottom-up process and propagated to other nodes in the attribute parse graph. For a super-pixel that does not contain sufficient number of edges, its surface normal will be inferred from the surrounding scene context.

2.2.2 Geometric Attributes of Planar Surface

We group spatially connected super-pixels into planar surfaces based on two types of features. i) Appearance features. We extract color and texture features to train a supervised classifier and assign a region to a few categories, e.g., 'building', 'tree', etc. ii) Geometry features. Superpixels in the same planar region should share the same surface normal. Both features are used in the iterative parsing process to form planar surfaces.

Each planar surface has three geometric attributes

- Two vanishing points: $\{(x_1, y_1), (x_2, y_2)\}$ and thus two Manhattan axes $\{(x_1, y_1, f), (x_2, y_2, f)\}$;
- Normal direction. As aforementioned, surface normal is simply the cross-product of the two Manhattan axes.
- A contact line and thus its 3D relative depth. The surface plane will intersect with other planes and form the contact lines. For example, Fig. 4 shows three planar surfaces of the building and their ground contact boundaries which can be approximated by straight lines respectively.

The contact lines may be occluded (e.g., between a building facade and the ground) or blurred (line between two surfaces of the building). Fortunately this can be solved by calculating the intersection line between adjacent surface planes, which usually points to one of the Manhattan axes associated with the surface planes. These geometric attributes are sufficient to reconstruct a planar-wise 3D scene model [21].

2.2.3 Geometric Attributes of Composite Surface

A composite surface consists of several planar surfaces that are physically connected. These surfaces might not belong to the same Manhattan frame. A composite surface has set of geometric attributes that pose consistency constraints between its children nodes in the parse graph. Its geometric attributes include:

- All vanishing points and surface normal of its planar surfaces.
- Contact lines between adjacent surfaces.
- A linear spline fit of the contact lines with the ground.

As planar surfaces, e.g., building facade, are usually occluded by foreground objects, e.g., vehicles and trees, and their boundaries to the ground plane are often partially visible. In Section 4 we shall introduce a robust method for estimating contact splines under these severe occlusions.

2.3 Geometric Attributes of Scene

The whole scene will pool over the geometric attributes from its components. As it is shown in Fig. 2, the root node S has the following geometric attributes.

- Camera parameters are shared by all nodes in the parse graph. Note that our model can be extended to reason other camera parameters, including skew, and optical center etc.
- m Manhattan frames $\{(x_{ij}, y_{ij}, f), i = 1, 2, \ldots, m, j = 1, 2, 3.\}$ for each local Manhattan world.

These global geometric attributes are used to constrain the geometric attributes of the entities in the parse graph. For example, the number of possible normal directions for planar surfaces are determined by the number of Manhattan axes detected for the global scene. In contrast, the past methods [22] [21] usually fix the number of surface normal orientations during inference.

3 Probabilistic Scene Grammar

In this section, we introduce a probabilistic treatment of the proposed attribute scene grammar.

3.1 Attribute Scene Grammar

Attribute grammar was first proposed by Han et al. in [17]. We extend it to model hierarchical scene representations in both 2D images and 3D scene space.

An attribute grammar is specified by a five-tuple: $\mathbf{G} = (\mathbf{V}_N, \mathbf{V}_T, S, \mathbf{R}, P)$, where $\mathbf{V}_N$ is a set of non-terminal

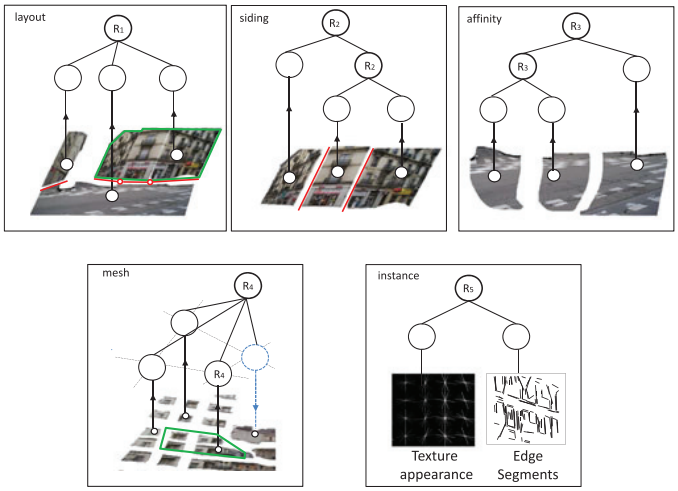


Fig. 5. Illustration of the five grammar rules. Each rule is associated with a set of geometric attributes that impose constraints over the parent node and children nodes. *Layout* rule: a children planar surface is supporting the other n children entities; *siding* rule: two children planar surfaces of the same semantic label are spatially connected; *affinity* rule: two children planar surfaces have the same appearance; *mesh* rule: multiple children surfaces appear in a 2D mesh structure; *instance* rule: links a terminal node and its image representation.

nodes, $\mathbf{V}_T$ is a set of terminal nodes, S is the root node for the whole scene, $\mathbf{R}$ is a set of production rules for spatial relationships, and P is a probability for the grammar.

These production rules can be recursively applied to generate a hierarchical representation of the input scene, namely the *Parse Graph*. A parse graph is a valid interpretation of the input 2D image and the desired 3D scene. A grammar generates a large set of valid parse graphs for one given image of the scene.

Terminal Nodes. We partition the input image into a set of superpixels and use them as terminal nodes. Each superpixel is the projection of a surface fragment in space. We denote all terminal nodes as $\mathbf{V}_T = \{a, X(a)\}$, where $X(a)$ denotes a set of attribute variables.

Non-Terminal Nodes are sequentially produced by merging terminal nodes or other non-terminals with grammar rules. Each node represents a planar surface or composite surface in space. There is one root node for the whole scene, i.e., S, and five production rules. 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 non-terminal nodes as $\mathbf{V}_N = \{(S, X(S)), (A, X(A))\}$ where S denotes the root node for the whole scene, A non-terminal node and $X(A)$ the attributes of A. Fig. 5 illustrates these five rules and Fig. 2 shows one parse graph that is capable of generating the input image.

Global and Local Attributes. Each graph node is associated with a number of attributes, which are either globally or locally defined.

Global attributes $X(S)$ are defined for the root node S and inherited by all graph nodes. It includes i) a list of possible categories (e.g.,'building') that appear in the input image, denoted as C; ii) geometric attributes, including the camera focal length f and Manhattan frames detected in the input image. Formally, we have $X(S) = (f, m, \{\mathrm{M}_i\}, C)$, $i = 1, .., m$. As aforementioned, each Manhattan frame M_i contains three orthogonal axes. *Local attributes* are defined over properties of scene entities, e.g., normal orientation for planar regions. These attributes are usually inherited from the global attributes and thus should be consistently assigned. Fig. 2 illustrates global geometric attributes in the left panel and local geometric attributes in the right panel. Semantic attributes are not included in the figure. Both global or local attributes are used to impose constraints to confine the inference process.

TABLE 1
Definitions of Grammar Rules and Their Geometric Attributes

Rules	Notations	Geometric Attributes
R_1: layout	$A \rightarrow (A_0, A_1, \ldots, A_m)$	$X(A) = (f, m, \mathrm{M}_i, \theta_0, \theta_{ij}, \vec{l}_k, C)$
R_2: Siding	$A \rightarrow (A_1, A_2)$	$X(A) = (\theta_i, \mathrm{M}_i, \vec{l}_k, c)$
R_3: Affinity	$A \rightarrow (A_1, A_2)$	$X(A) = (\theta, \mathrm{M}, c)$
R_4: Mesh	$A \rightarrow (A_1, A_2, \ldots)$	$X(A) = (\theta, \mathrm{M}, \mathbf{v}_1, \mathbf{v}_2, c)$
R_5:Instance	$A \rightarrow a$	$X(A) = (\theta, \mathrm{M})$

In the parse graph, each graph node corresponds to a grammar rule, and the attributes of a graph node are inherited by its offspring nodes.

3.2 Probabilistic Formulation for 3D Scene Parsing

We utilize a hierarchical parse graph to explicitly encode the attribute hierarchy (introduced in Section 2) for joint recognition and reconstruction purposes. In particular, terminal nodes with semantic attributes are used to derive the semantic region partition of the input image; the whole set of graph nodes with geometric attributes can be used to derive a full 3D scene model for reconstruction purpose.

Formally, let $\mathbf{G}$ denote the parse graph to solve, $\mathcal{A}$ all attributes in $\mathbf{G}$. Given an input image $\mathbf{I}$, we aim to compute a scene interpretation W in a joint solution space

$$W = (\mathcal{A}, \mathbf{G}). \tag{3}$$

The optimal solution W^* can be obtained by maximizing a posterior probability (MAP)

$$P(W|\mathbf{I}) \propto \exp\{-|\mathbf{V}_N| - \lambda^{\mathrm{gra}} E(\mathbf{I}, \mathbf{G}, \mathcal{A})\}, \tag{4}$$

where $|\mathbf{V}_N|$ indicates the number of non-terminal nodes. The first item is used to encourage compact parse graphs. λ^{gra} is a weight constant.

The energy $E(\mathbf{I}, \mathbf{G}, \mathcal{A})$ is defined over the hierarchy of $\mathbf{G}$, specifying how well $\mathbf{G}$ can generate the input image $\mathbf{I}$. Let $r(A)$ indicate the grammar rule used for the graph node A. We have,

$$E(\mathbf{I}, \mathbf{G}, \mathcal{A}) = \sum_{A \in V_N} \beta_{r(A)} E^t(\mathbf{I}, X(A)|r(A)), \tag{5}$$

where $r(A) \in [1..5]$ indicate the grammar rule associated with A, $\beta_{r(A)}$ is a weight constant that is dependent on $r(A)$. The energy term $E^t(\mathbf{I}, X(A)|r(A))$ is associated with the nonterminal node A and conditioned on the corresponding grammar rule $r(A)$.

Table 1 summarizes the definitions of grammar rules. In the rest of this section, we introduce the definitions of the five grammar rules.

3.2.1 Grammar Rule R_1: Layout

The *Layout* rule $R_1 : A \rightarrow (A_0, A_1, \ldots, A_n)$ states that a planar surface A_0 is supporting n entities. In this work we

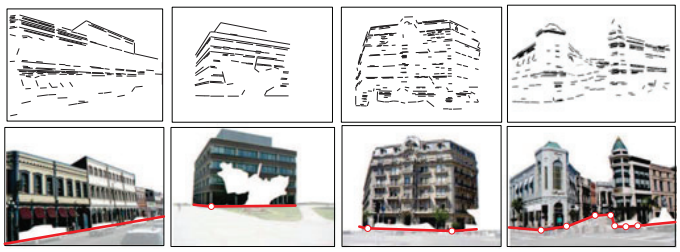


Fig. 6. Illustration of piece-wise linear spline model for the contact boundaries of composite surfaces. Each spline consists of several control points and the straight lines between these points. Note that each straight line correlates with one planar region in the composite surface.

assume that all stuff (objects, building, etc.) are standing upright on ground. A_0 indicates the ground region in images (e.g., grass, road, side walk etc.), and $A_1, \ldots, A_n$ indicate the n children surfaces or composite surfaces produced by the other grammar rules. Fig. 5 illustrates the use of R_1, which merges two building blocks/surfaces and the ground. The rule R_1 is used to generate the root node S.

The geometric attributes of S include both global and local attributes. Global attributes include a list of semantic categories, camera focal length f and m Manhattan frames. Each Manhattan frame includes three axes in space that are orthogonal to each other. The latter includes the normal directions of children surfaces, e.g., θ_0 for A_0, and the contact lines between A_0 and each of the m entities, denoted as $\vec{l}_k$. Formally, we have $X(S) = (f, m, \mathrm{M}_i, \theta_0, \theta_{ij}, \vec{l}_k, C), i, k = 1, \ldots, n$, where θ_{ij} represents one of the normal orientations in the ith children node, C a list of category labels.

We use continuous splines $\vec{l}_k$ to represent contact boundaries between A_0 and $\{A_k\}s$, which are assumed to be piecewise linear. Fig. 6 illustrates four typical scenes where contact splines are highlighted in red. A piece-wise linear spline consists of several control points and straight lines between them. Each straight line corresponds to the contact boundary of a planar region. In urban images, a contact line is usually parallel to one of the parallel families falling on the support region. This gives rise to a useful observation: if we can detect local edges in the given planar region and cluster these edges to parallel families, the direction of a contact line can be simply determined. With this observation, we will develop an effective search algorithm for discovering contact splines in Section 4.

We define the energy function for R_1 from two aspects. First, the normal direction of A_0 and normal directions of the children surfaces should be as distinct as possible. This is different from the previous works [13] [22] which assume orthogonality between adjacent surfaces. Second, contact lines are likely to go through vanishing points that have edges falling on the children planar region A_k. Thus, we have,

$$E^t(\mathbf{I}, X(A)|R_1) = \sum_{i,j} \mathrm{D}^{\cos}(\theta_0, \theta_{ij}) + \lambda^{\mathrm{lay}} \sum_{l \in \vec{l}_k} \min \mathrm{D}^{\cos}(l, \mathbf{v}) \quad \forall \mathbf{v} \in \mathrm{M}, \mathrm{M} \in X(A_k), \tag{6}$$

where l indexes the line segments included by the spline $\vec{l}_k$, M the Manhattan world in $X(A_k)$, $\mathrm{D}^{\cos}$ the cosine distance between two directions in 3D space. Note that $\mathbf{v}$ indicates one of the Manhattan axes in the Manhattan world associated with A_k.

3.2.2 Grammar Rule R_2: Siding

The *siding rule* $R_2 : A \rightarrow (A_1, A_2)$ states that two planar surfaces or composite surfaces of the same label are spatially connected in the scene. The parent node A is a composite surface and the children nodes A_1, A_2 could be planar surfaces. It requires that children surfaces share the same semantic label (e.g., building) but have different normal orientations. These surfaces are usually, but not necessarily, orthogonal with each other.

The attributes of R_2 include $X(A) = \{(\theta_i, \mathrm{M}_i), \vec{l}_k, c\}$, where θ_i is normal direction of the children surface A_i, $i = 1$, or 2, M_i the Manhattan frame associated with A_i, $\vec{l}_k$ the contact line between children surfaces, and c_i the semantic label.

The energy function for R_2 is derived from two aspects. First, the normal directions of two siding surfaces should be as distinct as possible, which is the case for most urban images. Second, the contact line of A is likely to point to the vertical vanishing point, denoted as $\mathbf{v}_0$, as Fig. 5 illustrates. Formally, we have,

$$E^t(\mathbf{I}, X(A)|R_2) = \sum_{i \neq j} \mathrm{D}^{\cos}(\theta_i, \theta_j) + \lambda^{\mathrm{sid}} \sum_k \mathrm{D}^{\cos}(\vec{l}_k, \mathbf{v}_0), \tag{7}$$

where λ^{sid} is a weight constant. Note that the semantic attributes c are used as hard-constraints: a graph node of R_2 is only applicable for children surfaces that have the same label.

3.2.3 Grammar Rule R_3: Affinity

The *affinity rule* $R_3 : A \rightarrow (A_1, A_2)$ states that two planar surfaces have similar appearance and thus should belong to the same planar surface. The children surfaces A_1 and A_2 should be spatially connected in 3D scene. In practice, since they could be disjoint in images due to occlusions, we allow the grouping of disjoint regions by this rule if they have high affinity in appearance. The attributes of A are defined as $X(A) = (\theta, \mathrm{M}, c)$ where θ is the normal direction, M the related Manhattan frame, and c the semantic label, which are shared by the two children surfaces.

The grammar rule R_3 requires that the children surfaces A_1 and A_2 should have the same surface normal. Thus, the geometric attributes serve as hard constraints and we only utilize the appearance information to define the energy function $E^t(\mathbf{I}, X(A)|R_3)$.

The energy function for R_3 include both unary and pairwise terms, defined over superpixel partition of the parent surface A. Let s and t index two neighbor superpixels, c_s the semantic label of superpixel s. We have,

$$E^t(\mathbf{I}, X(A)|R_3) = \sum_s \phi_s(c_s) + \lambda^{\mathrm{aff}} \sum_{s,t} \mathbf{1}(c_s = c_t), \tag{8}$$

where $\phi_s(c_s)$ returns the negative class likelihood ratio, and $\mathbf{1}()$ is an indicator function. Like [46], we estimate $\phi_s(c_s)$ by applying a non-parametric nearest neighbor estimator over training data. The second term is defined as a Potts/Ising model [47] to encourage homogeneousness of the labeling.

For a planar region, we estimate its surface normal based on the statistics of the straight edges falling on this region, as introduced in Section 2. However, if an image region does not contain any straight edge, there is no cue to tell its normal direction directly, and we need to infer its normal from the scene context. In Section 4, we shall introduce a robust inference method to deal with these uncertainties.

3.2.4 Grammar Rule R_4: Mesh

The *mesh rule* $R_4 : A \rightarrow (A_1, A_2, A_3, \ldots)$ states that multiple surfaces are arranged in a mesh structure. Children surfaces should be spatially connected to each other and share the same normal direction. In perspective geometry, a mesh structure in images can be described by two orthogonal vanishing points. Formally, the attributes of A include: $X(A) = (\theta, \mathrm{M}, \mathbf{v}_1, \mathbf{v}_2, c)$, where $\mathbf{v}_1 = (x_1, y_1, f), \mathbf{v}_2 = (x_2, y_2, f)$ are the coordinates of two vanishing points, $\theta = \mathbf{v}_1 \times \mathbf{v}_2$ is the normal direction of A, c the semantic label. The children surfaces share the same normal direction θ with A.

The energy function for R_4 is defined over edge statistics. As Fig. 6 illustrates, straight edges in a mesh region usually merge at two vanishing points. Let $\mathcal{E}(A)$ denote the set of local edges in A, and $\mathbf{l}_j = (x_j, y_j, \vec{d}_j) \in \mathcal{E}$ an edge at the position (x_j, y_j) with the orientation $\vec{d}_j$. Let v_i denote the image coordinate of the vanishing point $\mathbf{v}_i$. If an edge $\mathbf{l}_j$ points to $\mathbf{v}_i$, we have $(x_j, y_j) + \lambda_j^{\mathrm{mes}} \vec{d}_j = v_i$. Thus, we define $E^t(\mathbf{I}, X(A)|R_5)$ as

$$E^t(\mathbf{I}, X(A)|R_4) = \sum_{\mathbf{l}_j \in \mathcal{E}(A)} \min_{i, \lambda_j^{\mathrm{mes}}} \|v_i - (x_j, y_j) - \lambda_j^{\mathrm{mes}} \vec{d}_j\|^2, \quad (9)$$

where $i = 1, 2$. The least squares energy term is minimized while all edges in the mesh region exactly point to one of the two VPs, i.e., $\mathbf{v}_1$ or $\mathbf{v}_2$.

3.2.5 Grammar Rule R_5: Instance

An instance rule $R_5 : A \rightarrow a$ instantiates a terminal node to its image representation, including both texture appearances and edge segments. Fig. 5 illustrates how the grammar rule R_5 links a non-terminal node to two image representations: texture map that is represented as histogram of oriented gradient (HoG), and straight edge map.

The potential $E^t(\mathbf{I}, X(A)|R_5)$ is defined over two aspects: i) the region A should be perceptually homogeneous in appearance; ii) the directions of local straight edges should be consistent with the Manhattan frame assigned or inherited from the parent nodes of A. Let $\mathbf{I}_i$ and $\mathbf{I}_k$ denote two neighboring pixels in region A, we have

$$E^t(\mathbf{I}, X(A)|R_5) = \sum_{i,k} g(\mathbf{I}_i, \mathbf{I}_k) + \lambda^{\mathrm{ins}} \sum_{l \in \mathcal{E}(A)} \mathrm{minD}^{cos}(l, \mathbf{v}) \quad \forall \mathbf{v} \in \mathrm{M}, \mathrm{M} \in X(A), \quad (10)$$

where $g(\mathbf{I}_i, \mathbf{I}_k)$ returns the negative confidences of two pixels being homogeneous, λ^{ins} is a weight constant. The model $g(\mathbf{I}_i, \mathbf{I}_k)$ is directly estimated by the superpixel partition method [40] with both HoG features and edge features. The second term is used to encourage that all edges in A should be parallel to one of the Manhattan axes in $X(A)$.

Fig. 2 shows an exemplar parse graph generated by the proposed grammar. Each grammar rule describes a kind of spatial relationships, e.g., R_1 for supporting, R_2 for being co-block, R_3 and R_4 for being co-planar. These simple rules are capable of producing a large number of tree structured representations. It is worth noting that the tree structures are augmented to be parse graphs by linking spatially connected nodes of the same layer. This graph representation encodes both 2D appearance and 3D geometric properties of the hierarchical scene entities (as introduced in Section 2).

4 Inference

Our inference algorithm aims to construct an optimal parse graph by sequentially applying five grammar rules to maximize a posterior $P(W|\mathbf{I})$. This task is challenging because: a) the optimal parse graph does not have a pre-defined structure; b) the attribute constraints over attribute hierarchy are of high-order.

We develop a stage-wise method to solve the optimal parse graph, which includes three major stages. First, we introduce a heuristic search method to calculate camera parameters, i.e., the geometric attributes of the root node S, and fix the parameters $\mathcal{A}$ during inference. Note that the semantic attributes of S (i.e., possible category labels) are manually set for all outdoor scenes. Second, we solve region labeling to optimization by minimizing the energy function of Eq. (8) w.r.t. superpixel labels c_s. Eq. (8) is a typical Markov Random Field (MRF) type energy function that consists of a unary term and a regularization term of Potts/Ising prior. It can be efficiently solved by the loopy belief propagation (LBP) method [9]. We use the results of region labeling to initialize the desired parse graph. *Finally*, we introduce a data-driven Markov Chain Monte Carlo (DDMCMC) method to sample the posterior probability $P(\mathbf{I}|W)$.

Algorithm 1 summarizes the proposed inference algorithm. It includes two bottom-up computation steps and an iterative sampling step. The first two steps are used to narrow the search space and thus speed up the sampling procedure. We introduce these steps in the rest of this section.

Algorithm 1. Building Parse Graph via Attribute Grammar

1: **Input:** Single Image **I**;
2: Pre-processing: partition **I** into superpixels; detect vanishing points [28];
3: Bottom-up: calibration by heuristic search (Section 4.1);
4: Bottom-up: region labeling by belief propagation method (Section 4.2);
5: Initialize the parse graph **G**;
6: Iterate until convergence,
-Randomly select one of the five MCMC dynamics
-Make a proposal accordingly to reconfigure the current parse graph;
-Accept the change with probability;

4.1 Bottom-Up Computation: Calibration by Heuristic Search

We develop a stochastic heuristic search procedure to solve the optimal camera focal length and Manhattan frames. We first utilize the Hough Transform based voting method by

Li et al. [28] to detect families of parallel lines and their associated vanishing points (VPs). Next, we apply Eq. (1) over every pair of parallel families to estimate the camera focal length, by assuming they are orthogonal to each other. Let S denote the number of pairs of parallel families. We associate a binary variable to every pair of parallel families, denoted as $d_i \in [0, 1]$. Let $d_i = 1$ if the ith pair of families is orthogonal otherwise $d_i = 0$. Thus, we can solve the camera focal length $\hat{f}$ by minimizing the following objective

$$\min_{\hat{f},\{d_i\}} \frac{1}{S} \sum_{i=1}^{S} \|d_i f_i - \hat{f}\|, \tag{11}$$

where f_i is the estimation of the camera focal length from the ith pair of parallel families. To estimate f_i, we assume the ith pair of parallel families to be orthogonal and apply Eq. (1). To optimize Eq. (11), we introduce a heuristic search procedure. It starts with initializing at random $\{d_i\}$ followed by two iterative steps. Step 1: estimate focal length f_i from the ith pair if $d_i = 1$ and average over all estimations to get $\hat{f}$; Step-2: assign d_i to be 1 with the probability of $1/\{1 + \exp(|f_i - \hat{f}|)\}$. We iterate these two steps until convergence.

4.2 Bottom-Up Computation: Belief Propagation for Region labeling

The goal of this step is to assign every superpixel of the input image to one of five semantic labels, including 'sky', 'building', 'ground', 'trees' and 'other'. This is equal to estimating the optimal superpixel label assignment so as to minimize the energy function of Eq. (8) w.r.t. c_s.

We estimate the unary term in Eq. (8) as follows. Each superpixel is described using 20 different features, including shape, location, texture, color and appearance [46]. We first extract these features for training images. Next, we associate a semantic label with a training superpixel if 50 percent or more of the superpixel overlap with the ground truth segment mask of that label. Last, we compute class likelihood ratio for each superpixel in the test image, using the nearest neighbor estimator [46].

We use the efficient loopy belief propagation algorithm by Felzenszwalb et al. [9] to finalize the labeling. We use the min-sum algorithm that works by passing messages around the graph defined by the connected grid of superpixels. Each message is a vector of dimension given by the number of possible labels, 5 in this work. Since the smoothing term $\varphi_{<s,t,>}$ is semi-metric, the propagation algorithm can converge in $O(|C|NT)$ time where $|C|$ is the number of labels, N is the number of superpixels, and T is the number of iterations. Each iteration is very fast since we only have $|C| = 5$ candidate labels. We fix the maximal iteration number to be 10.

4.3 Iterative MCMC Sampling

Following the computations of camera calibration and region labeling, we design a data-driven Markov Chain Monte Carlo sampling algorithm (DDMCMC) [47] to search for the optimal parse graph. It starts with an initial parse graph that includes one root node and a set of terminal nodes, as illustrated in Fig. 7 (a). Next, we merge neighbor terminal nodes or superpixels that have the same semantic label to obtain non-terminal nodes using the grammar rule R_3. This step is greedily conducted and the resulting parse graph will be refined by the later iterative steps.

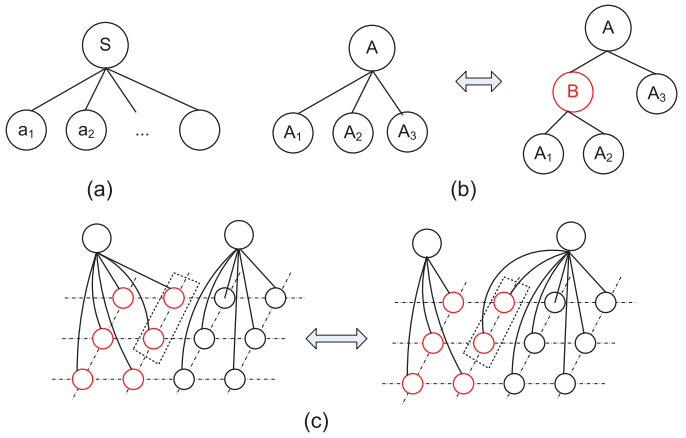


Fig. 7. Diffusion and jump dynamics. (a) An initial parse graph that includes a root node and terminal nodes; (b) jump dynamic: birth (from left-hand to right-hand) or death (from right-hand to left-hand) of non-terminal nodes; (c) diffusion dynamic: re-grouping superpixels.

In the following, we reconfigure the graph using a set of Markov Chain Monte Carlo (MCMC) dynamics. These dynamics are either *jump* moves, e.g., creating new graph nodes or deleting graph nodes, or *diffusion* moves, e.g., changing node attributes. Diffusion dynamics move the solution in a subspace of a fixed dimension whereas jump dynamics walk between subspaces of varying dimensions. These dynamics are paired with each other to make the solution change reversible [47], e.g., creating nodes paired with deleting nodes, changing attributes paired with itself. These stochastic dynamics are able to guarantee convergence to the target distribution $p(W|\mathbf{I})$.

Formally, a dynamic is proposed to drive the solution state from W to W', and the new solution W' is accepted with a probability, following the Metropolis-Hastings strategy [47]. The acceptance probability is defined as,

$$\alpha(W \to W') = \min\left(1, \frac{P(W'|\mathbf{I})Q(W \to W')}{P(W|\mathbf{I})Q(W' \to W)}\right), \tag{12}$$

where $Q(W' \to W)$ is the proposal probability.

We introduce five types of MCMC dynamics. The dynamics 1-2 are jump moves and the other dynamics are diffusion moves.

Dynamics 1-2: *birth/death of nonterminal nodes* are used to create or delete a nonterminal node and thus transition the current parse graph into a new graph as illustrated in Fig. 7.

The proposal for creating a nonterminal node was made by first selecting at random one of the four grammars, $R_1, \ldots,$ or R_4. Next, for the selected grammar rule, we obtain a list of candidates that are plausible according to the predefined constraints. Taking R_2 as example, two children nodes should i) have different normal orientations; ii) be spatially connected; and iii) be assigned to the same semantic label. Each candidate in this list is represented by its energy. Let B_i^k denote the ith candidate for the grammar rule R_k, $E^t(\mathbf{I}, X(B_i^k)|R_k)$ the related energy, and $|B_i^k|$ the

number of graph nodes in the sub-tree rooted at the node B_i^k. The list has the following form

$$\mathbf{L}^b = \{B_i^k, E^t(\mathbf{I}, X(B_i^k)|R_k), i = 1, 2, \ldots\}. \quad (13)$$

The proposal probability for selecting B_i^k is calculated as,

$$Q(W \to W') = 1 - \frac{E^t(\mathbf{I}, X(B_i^k)|R_k)/|B_i^k|}{\sum_j E^t(\mathbf{I}, X(B_j^k)|R_k)/|B_j^k|}. \quad (14)$$

Similarly, we obtain another set of candidate nodes to delete based on their energies,

$$\mathbf{L}^d = \{D_i^k, E^t(\mathbf{I}, X(D_i^k)|R_k), i = 1, 2, \ldots\}. \quad (15)$$

The proposal probabilities for deleting the node D_i^k is calculated as follows:

$$Q(W \to W') = \frac{E^t(\mathbf{I}, X(D_i^k)|R_k)/|D_i^k|}{\sum_j E^t(\mathbf{I}, X(D_j^k)|R_k)/|D_j^k|}. \quad (16)$$

Dynamics 3-4: *Merging/splitting regions* are used to relabel superpixels around boundaries between adjacent semantic regions. These jumps are used together to polish the results of image labeling obtained by the bottom-up computation in Section 4.2. Fig. 7c illustrates a typical example.

We obtain a list of candidate proposals for the merge/split dynamics as follows. *First*, we take superpixels located on boundaries of neighbor regions as graph nodes. These superpixels are usually with big ambiguities and the discriminative methods [9] do not necessarily work well. Second, we link all adjacent graph nodes to form an adjacent graph, and measure the links between adjacent nodes with appearance similarities. *Third*, we sample edge status of 'on' or 'off' based on edge similarities to obtain connected components (CCPs). We select one of the CCPs and change its semantic label to get a new solution state W'. This procedure is similar to that used by Barbu et al. [2] for the graph labeling task. Let CCP_i^k denote the ith CCP, $h(\mathrm{CCP}_i^k|W)$ denote its label confidence in the solution W, the list of proposals is denoted as follows,

$$L^m = \{\mathrm{CCP}_i^k, h(\mathrm{CCP}_i^k|W), h(\mathrm{CCP}_i^k|W'), i = 1, 2, \ldots, \}. \quad (17)$$

The proposal probability for selecting the ith candidate is defined as follows:

$$Q(W \to W') = \frac{h(\mathrm{CCP}_i^k|W')/h(\mathrm{CCP}_i^k|W)}{\sum_j h(\mathrm{CCP}_j^k|W')/h(\mathrm{CCP}_j^k|W)}. \quad (18)$$

Dynamic 5: *Switching Geometric Attributes* is used as diffusion dynamics to sample the geometric attributes of graph nodes. As aforementioned, the geometric attributes of the root node, including camera focal length and Manhattan frames, are calculated and fixed during inference. The geometric attributes of nonterminal nodes mainly include their respect *normal orientation*and *contact splines*.

Switching Normal θ. In a local Manhattan world, every normal direction corresponds to a Manhattan axis or a family of parallel lines. To determine the normal of a surface region, we cluster straight edges within this region to form families of parallel lines, each corresponding with a Vanishing Point (VP) [28], and find two orthogonal VPs that are mostly supported by straight edges. These two VPs can be used to determine the surface normal. Fig. 3 illustrates the geometric relations between surface normal and local edge statistic. Since edge directions include many noises, we use the estimated surface normal orientations to initialize geometric attributes of graph nodes at the beginning and refine them in the iterative sampling step. In particular, during inference we select at random one of the planar surfaces and change its normal randomly. We set the proposal probability to be constant so the acceptance probability is simply based on the posterior probability ratio.

Estimating Contact Spline $\vec{l}$. This dynamic is used to greedily estimate contact splines of composite surfaces. We take the grammar rule R_1: $A \to (A_0, A_1, A_2, \ldots, A_n)$ for instance to introduce our method for estimating contact spline. As aforementioned, a contact spline consists of multiple control points and straight lines between them, representing the boundary between the children surface A_i and the supporting surface A_0. Our method is motivated by the following observation: a contact line of A is likely to go through one of the vanishing points associated with A.

Let $\mathbf{V}$ denote the set of vanishing points (VPs) detected in the input image, $\mathbf{E}^i$ the set of edges with two end points: $< \mathbf{l}_s^i, \mathbf{l}_t^i > \in \mathbf{E}^i$ in the children surface A_i. Let $\mathbf{B}^i$ denote the set of boundary points and $b^{ij} \in \mathbf{B}^i$ the point coordinate. Our goal is to infer $n+1$ control points $\{\mathbf{c}^i\}$, and search for the associated VPs to which the contact lines point to, denoted as $\mathbf{v}^i$. We have the the following objective function

$$\begin{aligned} \min_{\{\mathbf{c}^i, \mathbf{v}^i\}} & \sum_{i,j,k} \mathrm{Dist}(\mathbf{c}^{i-1}, \mathbf{c}^i, \mathbf{v}^i) + \lambda^{\mathrm{bd}} \mathrm{Dist}(\mathbf{c}^{i-1}, \mathbf{c}^i, b^{ij}) \\ & + \lambda^{\mathrm{ed}} \mathrm{Dist}(\mathbf{l}_s^i, \mathbf{l}_t^i, \mathbf{v}^i) \\ s.t. \quad & \mathbf{v}^i \in \mathbf{V}, b^{ij} \in \mathbf{B}^i, < \mathbf{l}_s^i, \mathbf{l}_s^i > \in \mathbf{E}^i, \end{aligned} \quad (19)$$

where the function $\mathrm{Dist}(\mathbf{c}^{i-1}, \mathbf{c}^i, \mathbf{v}^i)$ returns the minimal distance between the point $\mathbf{v}^i$ and the contact line $< \mathbf{c}^{i-1}, \mathbf{c}^i >$. λ^{bd} and λ^{ed} are two constants. Eq. (19) minimizes the following three types of distances.

1) $\mathrm{Dist}(\mathbf{c}^{i-1}, \mathbf{c}^i, \mathbf{v}^i)$, the distance between the desired contact line $< \mathbf{c}^{i-1}, \mathbf{c}^i >$ and the VP it points to ;
2) $\mathrm{Dist}(\mathbf{c}^{i-1}, \mathbf{c}^i, b^{ij})$, the distance between the desired contact line and each of the boundary points, or fitting errors of the desired contact line;
3) $\mathrm{Dist}(\mathbf{l}_s^i, \mathbf{l}_t^i, \mathbf{v}^i)$, the distance between an edge segment in A^i and the desired VP $\mathbf{v}^i$.

In general, Eq. (19) is a NP-hard optimization problem. Fortunately, the feasible space is not huge and thus even an exhaustive search method is computationally acceptable. In order to deal with outliers and noises, we use the RANSAC technique to search for the approximate solution. We greedily solve the optimal contact spline, in order to reduce the computational complexity of inference.

Fig. 6 shows four exemplar results of our approach. It is worth noting the ground boundaries could be partially occluded or even fully occluded by objects (e.g., vehicles) or stuff (e.g., tree). The proposed method can predict the correct contact lines since edge statistics from the regions are

used for reasoning, whereas the alternate methods[22] [21] usually fail.

5 Experiments

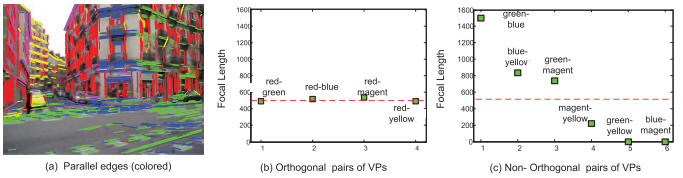
In this section, we apply the proposed method to recover 3D scene model from a single image, and evaluate it in both qualitative and quantitative ways.

5.1 Evaluation Protocols

Datasets. We use four datasets for evaluations. The first one is the CMU dataset collected by Hoiem et al. [22], which provides annotations of occlusion boundaries and surface normal orientations for each image. We use a subset of 100 images provided by Gupta et al. [13]. The surfaces are labeled with three main classes: 'ground', 'sky' and 'vertical', and the 'vertical' class is further divided into five subclasses: 'left', 'center', 'right', 'porous', and 'solid'. There are only three possible orientations for vertical surfaces. Note that our method associates normal orientations with Manhattan frames and a scene might have more than three Manhattan frames. To utilize these datasets, we arrange the discovered surface normal orientations from left-hand to centroid to right-hand and link them with the labels of 'left', 'center' or 'right'. We used 50 images for training and the rest for testing as [13].

We further collect three sets of images that do not follow the Manhattan assumption, e.g., having more than 3 vanishing points or including planar regions that are not orthogonal nor parallel to each other. For every image, we manually annotated vanishing points (VPs), region labels and surface normal orientations. The first dataset *LMW-A* consists of 50 images from the collections in [22], and there are 4.6 VPs per image on average. The second dataset *LMW-B* consists of 50 images from the dataset of EurasianCities in [7] with 4.2 VPs per image on average. The third one *LMW-C* consists of 950 images selected from the PASCAL VOC [8] and Labelme projects [41]. There are 3.5 VPs per image on average. These three datasets are used for testing only since our model is learned on the training subset of the CMU dataset.

Model Training. We utilize an empirical study of log-probability over training samples to estimate the optimal parameters in the model $p(W|\mathbf{I})$, including λs, βs and kernel widths used for the exponential functions. For each of these parameters we empirically quantize its possible values, e.g., $0.1, 0.3, \ldots, 1$ for β_1. Our goal is to select the optimal value for each parameter, i.e., determine the optimal parameter configuration. With each possible parameter configuration, we need to simulate a parse graph for every image from the annotations. To do so, we revise Algorithm 1 as follows: i) skip the step 4 for region labeling, since we have access to the annotated region label map and normal map; ii) only use the dynamics 1-2 (birth/death of non-terminal nodes) during MCMC sampling. This revised Algorithm 1 usually converges within a hundred of iterations (with dozens of graph nodes). After convergence, we calculate the log-probability $\log p(W|\mathbf{I})$. Thus, we select the parameter configuration that achieves the maximum log-probability. Similar simulation based estimation method has been used in previous works [47] [56].



(a) Parallel edges (colored) (b) Orthogonal pairs of VPs (c) Non- Orthogonal pairs of VPs

Fig. 8. Focal length estimation. (a) Input image overlaid with parallel families of edges (colored), each of which corresponds to a vanishing point on the image plane; (b) focal length estimated by orthogonal pairs of vanishing points; (c) focal length estimated by non-orthogonal pairs of vanishing points. The true focal length is 500 (red dotted lines).

Implementation of Algorithm 1. We resize each image so its maximal dimension is 500 pixels. We use the method by Ren et al. [40] to partition an image into 200-300 superpixels. We set the maximal iteration numbers to be 2000. It costs 5-6 minutes for Algorithm 1 to converge on a Dell Workstation (i7-4770 CPU@3.4 GHZ with 16 GB memory).

Baselines. We compare our method to two previous methods: i) the geometric parsing method by Hoiem et al. [22], ii) the method by Gupta et al. in [13]. Both methods can recover the three main geometric classes and the five vertical subclasses. We use the default parameters in their source codes.

We further implement three variants of the proposed method to verify the effects of individual grammar rules. i) *Ours-I*, that uses the grammar rules $R1$ (layout), $R2$ (siding), $R4$ (mesh), and $R5$ (instance) ; ii) *Ours-II*, that uses the grammar rules $R1$ (layout), $R3$ (affinity) and $R5$ (instance); and iii) *Ours-III*, that uses all five grammar rules. All these implementations include the grammar rules $R1$ and $R5$ and use all the five dynamics in inference. Note that the rule $R3$ encodes geometric information as well since it requires children nodes to have the same normal orientation. In addition, we include the region labeling results of the Belief Propagation algorithm for comparisons, denoted as *BP*.

5.2 Results

Camera Calibration. We first demonstrate how the orthogonality conditions of parallel families can be used to estimate camera focal length, as introduced in Section 4.1. We use the image shown in Fig. 8a, where one vertical VP and four horizontal VPs are detected. For each pair of VPs, we solve the Eq. (1) to estimate the camera focal length. Fig. 8b plots the estimated focal length (vertical direction) from four orthogonal pairs of VPs, i.e., the vertical VP and each of the four horizontal VPs. Fig. 8c plots the focal length estimated from non-orthogonal pairs VPs. The true focal length for this image is 500, plotted as red dotted lines in both figures. We can observe that i) in Fig. 8b, the estimations of focal length are roughly same with each other (low variance) and the average focal length 510 is quite close to the true value (high accuracy); ii) in Fig. 8c, in contrast, the estimations are with large variance, and most of them are not close to the true value. Therefore, we need to jointly estimate focal length and orthogonality conditions between parallel families. To do so, we use the heuristic search method (see Section 4.1) to minimize Eq. (11).

Qualitative Evaluations. Fig. 9 visualizes the results of normal map estimation over iterations of Algorithm 1. There are three main stages, stage-1: camera calibration, stage-2: region labeling and stage-3: iterative MCMC sampling. In the first row of Fig. 9, from left to right we plot the input

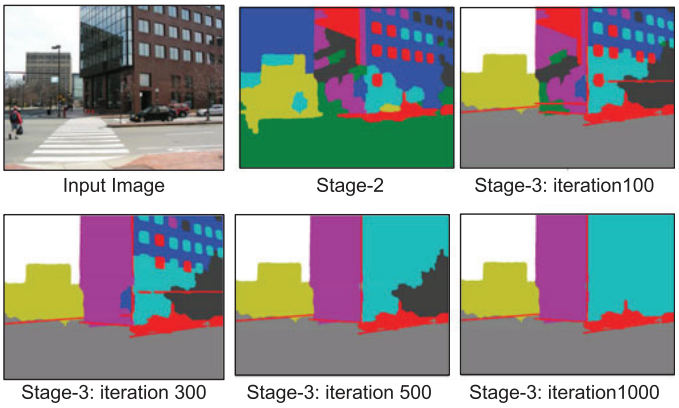


Fig. 9. Results of normal map estimation over iterations in Algorithm 1. Row-1: input image (left), surface normal map (middle) obtained during the stage-2, and that during the stage 3 (right). Row-2: three normal maps obtained during the stage 3 after 300, 500 and 1,000 iterations, respectively. Each color indicates a unique normal orientation.

image, surface normal map obtained during the stage-2 and surface normal map during the stage-3 (after 100 iterations). In the second row we plot three surface normal maps estimated during the stage-3 after 300, 500 and 1,000 iterations, respectively. The figures are overlaid with contact splines when applicable. We can observe that the surface normal map is continuously refined over iterations. Fig. 10a plots the convergence curve of Algorithm 1 on this input image. The curve shows how the energy $E(\mathbf{I}, \mathbf{G}, \mathcal{A})$ changes w.r.t. iterations. We also plot the convergence curve of our previous inference algorithm [31] for side-by-side comparisons. Note that we scale the two curves so that they start from the same energy. We can observe that Algorithm 1 converges after 1,000 iterations which is a lot faster than [31]. In addition, Fig. 10b visualizes the changes of average energy on all the 50 images from the LMW-A dataset. We plot the average optimal energy calculated from groundtruth annotations for comparisons. These two sub-figures clearly demonstrate the advantages of the proposed inference algorithm over [31]. The reasons are two-fold: i) the bottom-up computation step for region labeling in Algorithm 1 provides good initializations to the MCMC sampling process; and ii) the newly introduced five dynamics are more effective than the dynamics used in [31].

Fig. 11 shows exemplar results of *Ours-III* on the CMU dataset [22]. In each cell, we plot (a) the input image

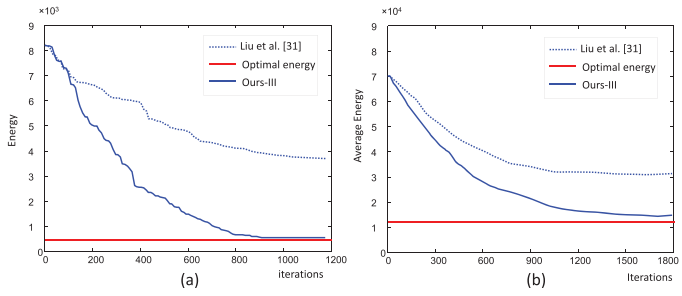


Fig. 10. Convergence of Algorithm 1. (a) Convergence curve calculated for the image in Fig. 9, i.e., the energy at each iteration. The solid red line indicates the optimal energy calculated from the ground-truth annotation. (b) Convergence curve calculated on 50 images from the dataset LMW-A, which plots the average energy at each iteration. We plot the average optimal energy (red line) of these images for comparisons.

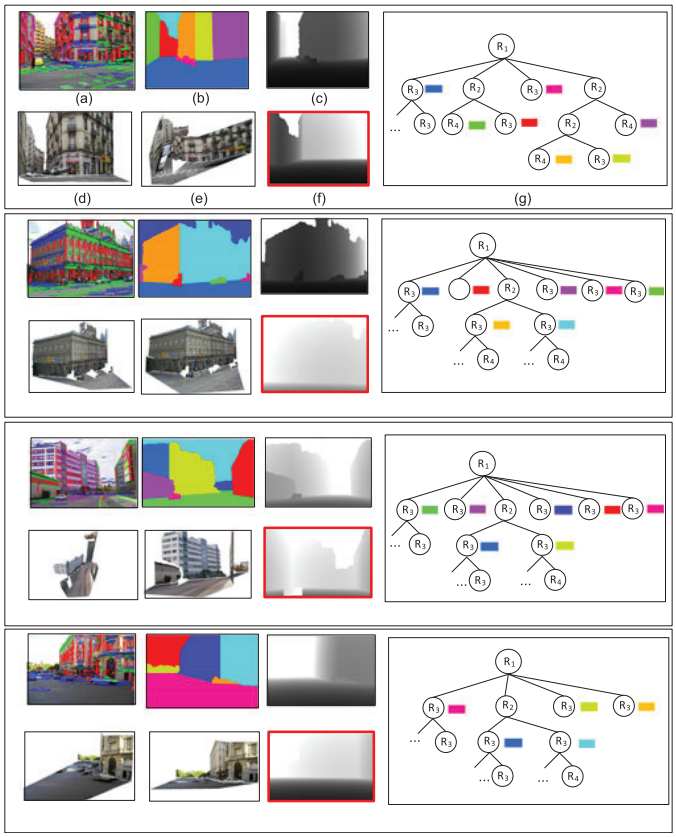


Fig. 11. Exemplar results on CMU dataset. (a) Input image overlaid with families of parallel lines; (b) surface normal map; (c) estimated depth map; (d-e) newly synthesized views;(f) depth map by Hoiem et al. [22]; (g) estimated parse graph where the colored rectangles correspond with the semantic region in subfigures (b).

overlaid with families of parallel lines, where each color indicates a parallel family; (b) the layout partition where each color indicates one planar surface; (c) the estimated depth map where darker pixels indicate being closer to the camera and vice versa; (d-e) three synthesized images from novel viewpoints; (f) the depth map estimated by [22]; and (g) the parse graphs created during inference. In Fig. 11g we only show the top levels of the parse graph where each colored rectangle corresponds to one planar surface in subfigure (b) with the same color. Our results are very promising considering that only a single viewpoint of the scene is available and no groundtruth 3D scene models are used for training. Taking the first example for instance, since the far-right building region in purple is occluded by vehicles and trees, none of the previous methods can tell where the contact line between the facade and ground is. Our approach, however, is able to infer the contact line from edge statistics extracted from this region.

Figs. 12 and 13 show results of our method on the datasets LMW-A and LMW-B, respectively. We also include the results by Hoiem et al. [22] for comparisons. In each cell, we show (a) input image overlaid with parallel families; (b) superpixel partition overlaid with VPs; (c) surface normal map by our method; (d) depth map by our method; (e-g): three novel viewpoints synthesized; and (h) depth map by [22]. From the comparisons between (d) and (h), we can observe that our method is capable of creating better 3D models than [22]. Fig. 14 visualizes exemplar results of our method on the dataset LMW-C. While the recovered 3D

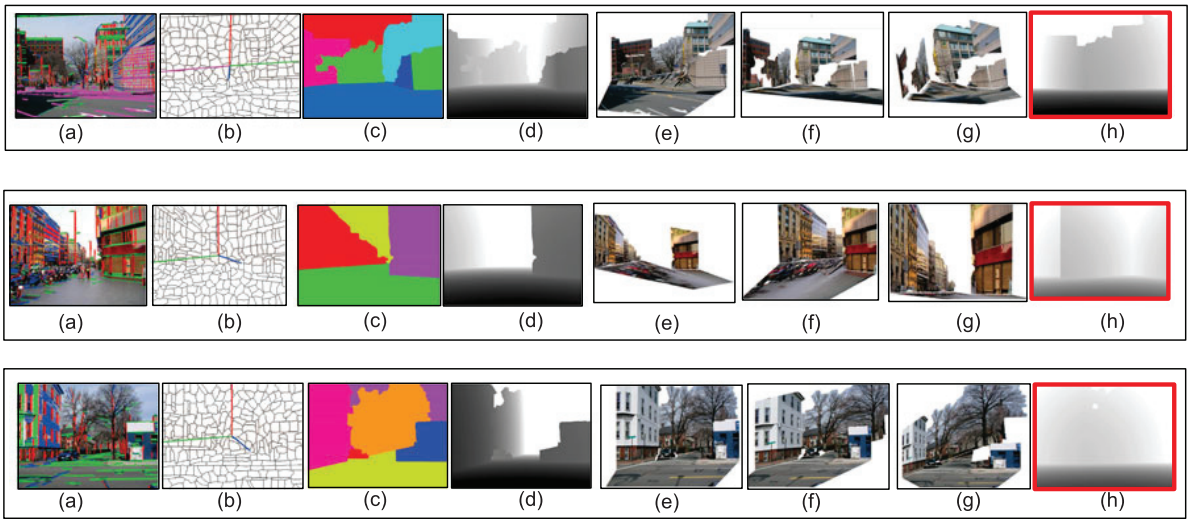


Fig. 12. Exemplar results on LMW-A dataset. For each cell, we show (a) an input image overlaid with families of parallel edges; (b) the superpixel partition overlaid with vanishing points; (c) the obtained surface normal map; (d) the depth map by the proposed method Ours-III; (e-g) three newly synthesized views; (h) the depth map estimated by Hoiem et al. [22].

scene models are considerably accurate, these results demonstrate a few drawbacks of the proposed method. First, our current model does not deal with foreground objects, e.g., vehicle in the first image and the pedestrian in the second image. Second, our method cannot work well for isolated structure-less regions, e.g., tree or plants in the third and fourth images, which do not include rich geometric regularizations. Third, our method assumes surface regions to be planar which might not be true, e.g., the image in the second row. In addition, for the fifth image, our approach does not group the two building regions to be a single block because of the large occlusions by a chunk of plants; for the sixth image, our method fails to identify the normal orientations of the facades on the left-hand side building because all straight edges are grouped into two parallel families. These two failure examples call for a more principled way for dealing occlusions and failures of pre-processing steps (e.g., vanishing point detections), which will be discussed in Section 6.

We further apply the proposed method *Ours-III* over indoor images to demonstrate its generalization capability. We use the dataset collected by Zhao and Zhu [56] that comprises of 220 indoor images for training. Note that the dataset includes foreground objects (e.g., sofas, tables), and we only focus on the reconstruction of room layout. We consider three indoor categories: floor, ceiling, and wall. The implementation details remain the same as *Ours-III*. Fig. 15 plots several exemplar results, including input images, layout segmentation and newly synthesized images. From these results, we can observe that the recovered surface normal maps are very accurate even when there are clutters in front of the scene entities, e.g., walls are occluded by sofas

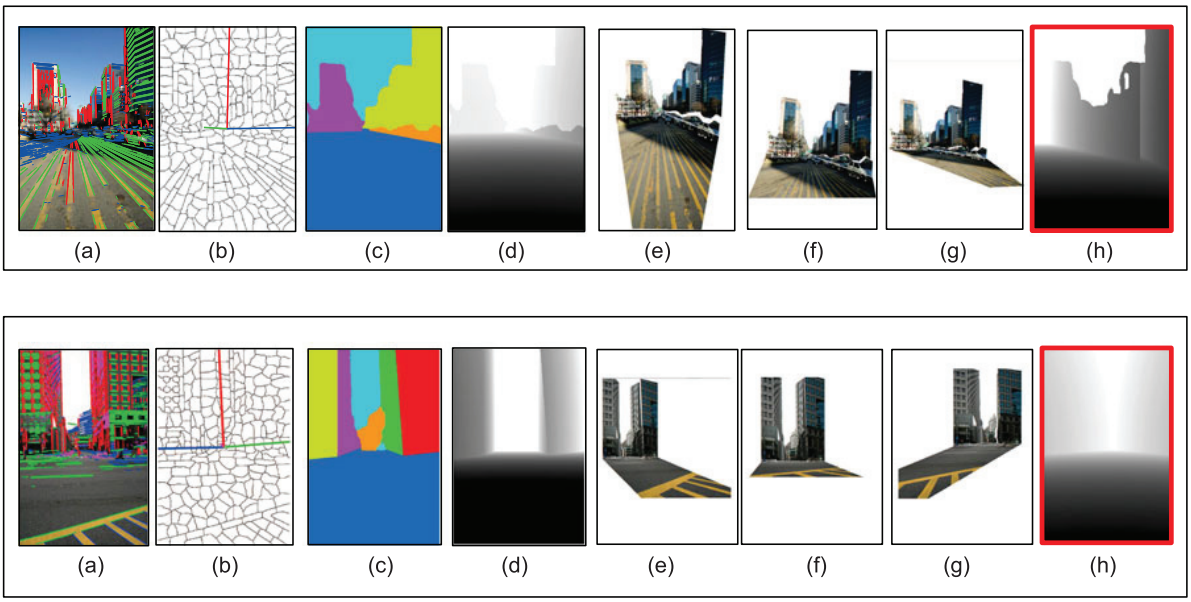


Fig. 13. Results on the LMW-B dataset. For each cell, we show (a) an input image overlaid with families of parallel edges; (b) the superpixel partition overlaid with vanishing points; (c) the obtained surface normal map; (d) the depth map by the proposed method Ours-III; (e-g) three newly synthesized views; (h) the depth map estimated by Hoiem et al. [22].

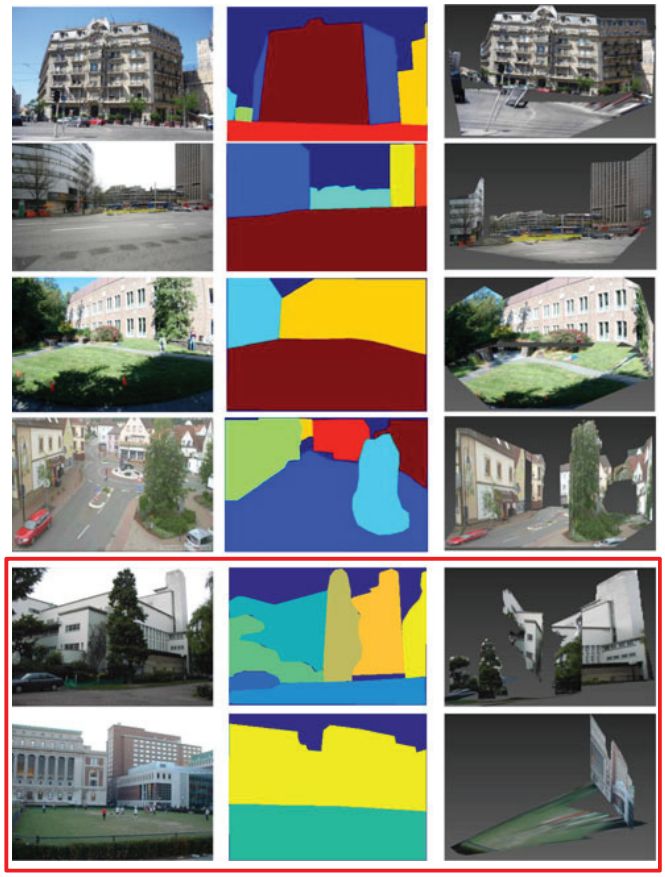


Fig. 14. Results on the LMW-C dataset by the proposed Ours-III. Column-1: input image; Column-2: surface normal map; Column-3: newly synthesized view. In Rows 1-4, we plot four images for which Ours-III can successfully estimate the 3D scene models. In Rows 5-6, we visualize two failure examples. For the image in the 5th row, our method fails to group the two building regions to be a single block because of the chunk of occlusions by plants; for the image in the 6th row, our method fails to identify the normal orientations of the facades on the left-hand side building.

(first image) or tables (third image). The obtained 3D models, however, can be further improved by reconstructing foreground objects, e.g., persons, tables, pillars etc.

Quantitative Results. We report the numerical comparisons of the various methods in terms of *normal orientation estimation* and *region labeling*.

For normal orientation estimation, we use the metric of *accuracy*, i.e., percentage of pixels that have the correct normal orientation label, and average accuracies over test images for every dataset. On the estimation of main geometric classes, i.e., 'ground', 'vertical', and 'sky', both our method and baseline methods can achieve high-quality

Fig. 15. Results on indoor images by the proposed Ours-III. Each row shows an exemplar result. Column-1: input images; Column-2: surface normal maps; Column-3: newly synthesized views.

results with accuracy 0.98 or more. Therefore, we focus on the vertical subclasses, like [13], and discard the superpixels belonging to ground and sky while calculating the accuracies for all methods.

Table 2 reports the numerical comparisons on four datasets. From the results, we can observe the following. First, the proposed *Ours-III* clearly outperforms the other baseline methods on all the four datasets. Taking the CMU dataset for instance, the method by Gupta et al. [13] has an average accuracy of 73.72 percent, whereas ours performs at 79.53 percent. On the other three datasets that have accurate normal orientation annotations, the improvements by our method are even more. As stated by Gupta et al. [13], it is hard to improve vertical subclass performance. Our method, however, can improve these two baselines with large margins. Second, *Ours-III* clearly outperforms the other two variants, i.e., *Ours-I* and *Ours-II*, that use less types of grammar rules. These comparisons justify the effectiveness of the proposed grammar model. Third, *Ours-III* has good margins over our previous method [31], which follows the same methodology. As summarized in Section 1.2, we improve [31] with a sophisticated representation model and an advanced inference algorithms, which are justified to be effective.

Table 3 reports the region labeling performance on the four datasets. We use the *best spatial support* metric as [13],

TABLE 2
Numerical Comparisons on Normal Orientation Estimation

	CMU dataset [22]	LMW-A	LMW-B	LMW-C
Gupta et al. [13]	73.72 %	62.21 %	59.21 %	58.39
Hoiem et al. [22]	68.80 %	56.30 %	52.70 %	53.28
Liu et al. [31]	76.34 %	67.90 %	64.30 %	62.34
Ours-I	74.24 %	67.35 %	63.18 %	60.41
Ours-II	75.87 %	68.39 %	64.29 %	62.78
Ours-III	**79.53** %	**71.40** %	**68.51** %	**65.92**

TABLE 3
Numerical Comparisons on Region Labeling

	CMU dataset [22]	LMW-A	LMW-B	LMW-C
BP	65.23%	55.23%	58.72 %	56.34
Gupta et al. [13]	68.85%	59.21%	60.28%	60.19
Hoiem et al. [22]	65.32 %	58.37%	57.70 %	59.25
Liu et al. [31]	72.71%	66.45%	65.14 %	63.17
Ours-I	69.34%	68.09 %	63.75 %	62.32
Ours-II	75.69%	70.15 %	65.91 %	65.47
Ours-III	**78.32**%	**71.23** %	**70.24** %	**68.79**

which first estimates the best overlap score of each ground truth labeling and then averages it over all ground-truth labeling. We discard the superpixels belonging to ground and sky while calculating the accuracies of all methods. Our method outperforms the method [13] with the margins of 9.47, 12.02, 9.96, 8.60 percentages on the four datasets, respectively. It is worth noting that all the three variants of our methods outperform the baseline *BP* that provides initializations of region labeling. These comparisons show that jointly solving recognition and reconstruction can bring considerable improvements over recognition accuracies.

6 Conclusions

This paper presents an attribute grammar for 3D scene reconstruction from a single view. We introduce a unified grammar model that includes five grammar rules to generate a hierarchical image representation for both 2D recognition and 3D reconstruction purposes. The developed inference method can efficiently search the constrained space by optimizing both the 2D surface layout and geometric attributes required for estimating 3D scene model. Extensive evaluations on challenging image collections show that our method outperforms the other popular methods for single-view 3D scene reconstruction.

This work contributes a generic probabilistic framework for jointly solving 2D recognition problems, e.g., classification, detection, tracking, etc., and 3D reconstruction problems, e.g., camera calibration, depth estimation, geolocalization, etc. There are two particular directions to exploit in the future: i) developing new solution for existing joint tasks, e.g., calibration from tracking [12]; ii) motivating novel vision tasks, e.g., jointly solving tracking and geolocalization. Taking the failure examples in Fig. 14 for example, we might jointly solve vanishing point detection and 3D reconstruction to avoid pre-mature decisions in pre-processing steps.

Our method is currently limited to the reconstruction of background scene entities, e.g., building, ground, tree, etc. The developed representation and algorithms, however, can be easily extended to parse foreground objects as well, e.g., car, human, etc. This is actually equal to jointly solving object detection, region labeling and 3D scene reconstruction. The 3D position or pose of an object shall be regularized by the global geometric attributes, e,g., camera focal length. Another limitation of our approach is that it aims at reconstructing major surfaces, e.g., building, road, etc., rather than fine-grained scene entities, e.g., edges, corners, rectangles, which leads to a coarse 3D scene model. A research direction in the future is to jointly reconstruct these scene entities to obtain a fine-grained scene model.

In addition, a critical problem in the line of this research is how to accelerate the inference algorithm which is slow because of its stochastic nature. Since the goal is to create a hierarchical parse graph, a promising way is to utilize the composition principle of the hierarchy with a more efficient inference algorithm, e.g., dynamic programming.

Acknowledgments

The project was supported by ONR MURI N00014-16-1-2007, DARPA SIMPLEX project N66001-15-C-4035, NSF IIS 1423305 and NSF IIS 1657600, and NSF IIS 1423305. The three image datasets were collected and annotated by the first author and his students at San Diego State University (SDSU). Xiaobai Liu is the corresponding author.

References

[1] Y. Bao, M. Chandraker, Y. Lin, and S. Savarese, "Dense object reconstruction using semantic priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1264–1271.
[2] A. Barbu and S.-C. Zhu, "Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1239–1253, 2005.
[3] R. Cabral and Y. Furukawa, "Piecewise planar and compact floorplan reconstruction from images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 628–635.
[4] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *Int. J. Comput. Vis.*, vol. 4, no. 2, pp. 127–140, 1990.
[5] M. Choi, A. Torralba, and A. Willsky, "A tree-based context model for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 240–252, Feb. 2012.
[6] J. Coughlan and A. Yuille, "Manhattan world: Orientation and outlier detection by Bayesian inference," *Neural Comput.*, vol. 15, no. 5, pp. 1063–1088, 2003.
[7] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky, "Geometric image parsing in man-made environments," *Int. J. Comput. Vis.*, vol. 97, no. 3, pp. 305–321, 2012.
[8] M. Everingham, S. Eslami, L. V. Gool, C. Williams, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.
[9] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, pp. 41–54, 2006.
[10] P. Felzenszwalb and O. Veksler, "Tiered scene labeling with dynamic programming," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, pp. 3097–3104, 2010.
[11] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, pp. 1422–1429, 2009.
[12] A. Gillbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 125–136.
[13] A. Gupta, A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 482–496.
[14] C. Haene, N. Savinov, and M. Pollefeys, "Class specific 3D object shape priors using surface normals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 652–659.
[15] C. Haene, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3D scene reconstruction and class segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 97–104.
[16] F. Han and S.-C. Zhu, "Bayesian reconstruction of 3d shapes and scenes from a single image," in *Proc. Int. Workshop High Level Knowl. 3D Model. Motion*, Oct. 2003, pp. 12–20.
[17] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 59–73, Jan. 2009.
[18] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded classification models: Combining models for holistic scene understanding," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 641–648, 2008.
[19] M. Hejrati and D. Ramanan, "Analysis by synthesis: Object recognition by object reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2449–2456.
[20] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, pp. 654–661.
[21] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2137–2144.
[22] D. Hoiem, A. Efros, and M. Hebert, "Closing the loop on scene interpretation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[23] P. Koutsourakis, L. Simon, O. Teboul, G. Tziritas, and N. Paragios, "Single view reconstruction using shape grammars for urban environments," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 1795–1802.

[24] A. Kundu, Y. Li, F. Daellert, F. Li, and J. Rehg, "Joint semantic segmentation and 3D reconstruction from monocular video," in *Proc. Eur. Conf. Comput. Vis.*, pp. 703–718, 2014.
[25] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 89–96.
[26] L. Ladicky, B. Zeisl, and M. Pollefeys, "Discriminatively trained dense surface normal estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 468–484.
[27] J. Lafferty, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[28] B. Li, K. Peng, X. Ying, and H. Zha, "Vanishing point detection using cascaded 1D Hough transform from single images," *Pattern Recog. Lett.*, vol. 33, pp. 1–8, 2012.
[29] J. Lim, H. Pirsiavash, and A. Torralba, "Parsing IKEA objects: Fine pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2992–2999.
[30] C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler, "Rent3D: Floor-plan priors for monocular layout estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 3413–3421, 2015.
[31] X. Liu, Y. Zhao, and S.-C. Zhu, "Single-view 3D scene parsing by attributed grammar," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 684–691.
[32] D. Marr, "Vision: A computational investigation into the human representation and processing of visual information," *Henry Holt Co. Inc.*, New York, NY, vol. 2, pp. 4–2, 1982.
[33] H. Mobahi, Z. Zhou, A. Yang, and Y. Ma, "Holistic 3D reconstruction of urban structures from low-rank textures," in *Proc. 11th Asian Conf. Comput. Vis.*, pp. 593–600, 2012.
[34] N. Payet and S. Todorovic, "Scene shape from textures of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 2017–2024.
[35] L. Pero, J. Bowdish, E. Hartley, B. Kermgard, and K. Barnard, "Understanding Bayesian rooms using composite 3D object models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 153–160.
[36] L. Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard, "Sampling bedrooms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 2009–2016.
[37] L. Pero, J. Guan, E. Hartley, B. Kermgard, and K. Barnard, "Bayesian geometric modeling of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 2719–2726.
[38] L. Pero, J. Guan, E. Hartley, B. Kermgard, and K. Barnard, "Understanding Bayesian rooms using composite 3D object models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 153–160.
[39] J. Prokaj and G. Medioni, "Using 3D scene structure to improve tracking," in *Proc. IEEE Conf. Computer Vis. Pattern Recog.*, 2011, pp. 1337–1344.
[40] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 10–17.
[41] B. Russell, et al., "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 157–173, 2008.
[42] A. Saxena, M. Sun, and A. Ng, "Make3d: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.
[43] A. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun, "Box in the box: Joint 3D layout and object reasoning from single images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 353–360.
[44] A. Schwing and R. Urtasun, "Efficient exact inference for 3D indoor scene understanding," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 299–313.
[45] J. Straub, G. Rosman, O. Freifeld, J. Leonard, and J. F. III, "A mixture of Manhattan frames: Beyond the Manhattan world," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3770–3777.
[46] J. Tighe and S. Lazebnik, "SuperParsing: Scalable nonparametric image parsing with superpixels," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 329–349, 2013.
[47] Z. Tu and S.-C. Zhu, "Image segmentation by data-driven markov chain Monte Carlo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 657–673, May 2002.
[48] S. Wang, J. Joo, Y. Wang, and S.-C. Zhu, "Weakly supervised learning for attribute localization in outdoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3111–3118.
[49] P. Wei, Y. B. Zhao, N. Zheng, and S.-C. Zhu, "Modeling 4D human-object interactions for event and object recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3272–3279.
[50] J. Xiao and Y. Furukawa, "Reconstructing the world's museums," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 668–681.
[51] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SFM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1625–1632.
[52] J. Xiao, B. Russell, and A. Torralba, "Localizing 3D cuboids in single-view images," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 746–754, 2012.
[53] H. Zhang, A. Geiger, and R. Urtasun, "Understanding high-level semantics by modeling traffic patterns," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3056–3063.
[54] Y. Zhang, S. Song, P. Tan, and J. Xiao, "PanoContext: A whole-room 3D context model for panoramic scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 668–686.
[55] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[56] Y. Zhao and S.-C. Zhu, "Image parsing via stochastic scene grammar," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 73–81.

Xiaobai Liu received the PhD degree from the Huazhong University of Science and Technology (HUST), China. He is currently an assistant professor of computer science with San Diego State University (SDSU), San Diego. His research interests focus on the development of theories, algorithms, and models for the core computer vision problems. He has published more than 30 peer-reviewed articles in top-tier conferences (e.g., ICCV, CVPR, etc.) and leading journals (e.g., the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Image Processing* etc.). He received a number of awards for his academic contributions, including the 2013 outstanding thesis award by CCF (China Computer Federation). He is a member of the CVF.

Yibiao Zhao received the PhD degree from the University of California, Los Angeles (UCLA). He is currently a postdoctoral researcher in the Massachusetts Institute of Technology (MIT). His research interests include vision and cognitive science, statistical learning, and inference. He is a co-chair of the Int'l Workshop on Vision Meets Cognition: Functionality, Physics, Intent, and Causality 2014. He is a member of the IEEE.

Song-Chun Zhu received the PhD degree from Harvard University, in 1996. He is currently a professor of statistics and computer science with UCLA, and director of the Center for Vision, Cognition, Learning and Autonomy (VCLA). He has published more than 200 papers in computer vision, statistical modeling, learning, cognition, and visual arts. In recent years, his interests has also extended to cognitive robotics, robot autonomy, situated dialogs, and commonsense reasoning. He received a number of honors, including the Helmholtz Test-of-time award in ICCV 2013, the Aggarwal prize from the Int'l Association of Pattern Recognition in 2008, the David Marr Prize in 2003 with Z. Tu et al., twice Marr Prize honorary nominations with Y. Wu et al. in 1999 for texture modeling and 2007 for object modeling respectively. He received the Sloan Fellowship in 2001, a US NSF Career Award in 2001, and a US ONR Young Investigator Award in 2001. He is a fellow of the IEEE since 2011, and served as general co-chair for CVPR 2012.