Single-View 3D Scene Parsing by Attributed Grammar

Xiaobai Liu, Yibiao Zhao, Song-Chun Zhu
Dept. Statistics, University of California at Los Angeles, CA, USA 90095
{lxb, ybzhao}@ucla.edu, sczhu@stat.ucla.edu

Abstract

In this paper, we present an attributed grammar for parsing man-made outdoor scenes into semantic surfaces, and recovering its 3D model simultaneously. The grammar takes superpixels as its terminal nodes and use five production rules to generate the scene into a hierarchical parse graph. Each graph node actually correlates with a surface or a composite of surfaces in the 3D world or the 2D image. They are described by attributes for the global scene model, e.g. focal length, vanishing points, or the surface properties, e.g. surface normal, contact line with other surfaces, and relative spatial location etc. Each production rule is associated with some equations that constraint the attributes of the parent nodes and those of their children nodes. Given an input image, our goal is to construct a hierarchical parse graph by recursively applying the five grammar rules while preserving the attributes constraints. We develop an effective top-down/bottom-up cluster sampling procedure which can explore this constrained space efficiently. We evaluate our method on both public benchmarks and newly built datasets, and achieve state-of-the-art performances in terms of layout estimation and region segmentation. We also demonstrate that our method is able to recover detailed 3D model with relaxed Manhattan structures which clearly advances the state-of-the-arts of single-view 3D reconstruction.

1. Introduction

Automatically creating high-quality 3D model from single view provides the background context to other high-level vision tasks, e.g. human activities recognition. This is a challenging problem due to its ill-posed nature. However, for the image of man-made outdoor scenes, human can recognize 3D structure of the scene effortlessly. We conjecture that human make 3D inference with some commonsense knowledge, such as most of the objects placed on the ground due to gravity, building are most standing uprightly, man-made scenes usually have Manhattan type structure [4], or parallel lines in the words merge at vanishing points in images. Recently, researchers also tried to use the physics law to guide the 3D reconstruction [6]. Integrating these cues can definitely improve the system performance whereas an open problem is how to select the most useful knowledge during the inference.

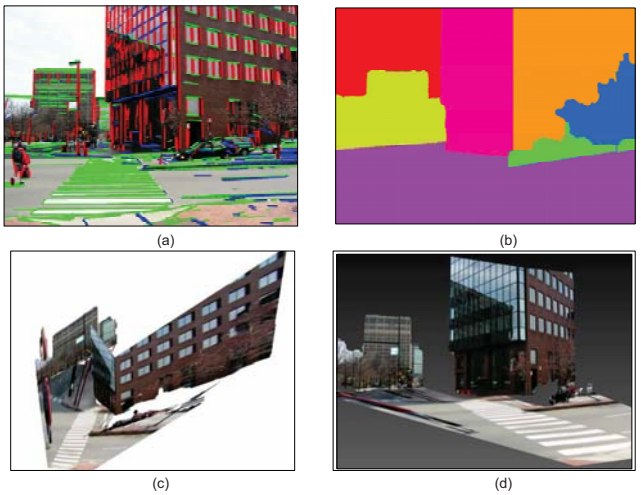


Figure 1. A typical result of our approach. (a) input image overlaid detected parallel lines; (b) segmentation of scene layout; (c) and d) synthesized images from novel viewpoints.

In this paper, we present a simple attributed grammar for the 3D parsing of man-made scenes. The basic observation is, like language where a large number of sentences are generated by a small set of words through a few of grammar rules, the visual patterns in the scene can be decomposed hierarchically into primitives through a few grammar rules. The grammar uses the superpixels as its terminal nodes.

Given one image, our goal is to build a hierarchical parse graph where each nonterminal node corresponds to a production rule. Our grammar uses attributes as switch variables to introduce constraints on nodes. Figure 2 illustrates a hierarchical representation of an outdoor scene. In this parse tree, the vertical links show the decomposition of the scene or one node into their components, and the horizontal links specify the spatial relationship between components. Both vertical and horizontal relationships are regularized by the local attributes of the nodes and the global attributes of the whole scene. The global attributes include the camera

parameters, e.g. focal length, and multiple Cartesian coordinate systems (CCS). Each CCS includes three or two orthogonal families of parallel lines. In contrast with the Manhattan world [4] which partitions all the parallel lines into three orthogonal families or one single CCS, we allow one scene to have multiple CCSs and further assume each surface belongs to one of the CCSs. Two CCSs may share at most one parallel family. These attributes are associated with the root node and will be inherited by all the nodes in the hierarchical parse graph. Every node, however, has its own constrain equations, which may use parts of these attributes.

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

2. Related Work

Our work is closely related to the recent advances in semantic scene labeling, single view modeling, and scene grammar.

Semantic scene labeling The method of conditional random fields [12] are widely used to represent semantic relations by maximizing the affinity between object labels in computer vision. These works considered some qualitative context descriptions such as {inside, below, around, above}, which are proved to be helpful to recognize outdoor objects. Choi et al. [2] studied 2D context models that guide detectors to produce a semantically coherent interpretation of a scene. They demonstrated that 2D horizontal contexts are very sensitive to camera rotations.

Single-View 3D modeling Hoiem et al. [8] explored rich geometric features and context information to recognize the surface orientation labels of 2D regions. Gould et al. extend their model to recognize geometric surface labels and semantic labels simultaneously. Gupta et al. [6] consider 3D objects as blocks and infer their 3D properties such as occlusion, exclusion and stability. However, their methods are largely built on top of classification of the 2d segments, which does not directly reconstruct 3d or infer depth value. Payet and Todorovic [15] proposed a joint model to recognize objects and estimate scene shape simultaneously. Mobahi et al. [14] reconstruct a single view by extracting low rank textures on building facade. Saxena et al. [18] present a fully supervised method to learn a mapping between informative feature and depth value. It is hard to recover global 3D scene without an explicit representation of camera model and 3d geometric structures. In contrast, our method directly solve the optimal layout segmentation as well the spatial arrangement in the 3D space, which will give rises to a high-quality 3D model.

Scene grammar Koutsourakis et al. [11] proposed a shape grammar to explain build facades with increasing levels of details, but their model only works on rectified facade images does not handle the challenge of 3d geometry. Han and Zhu [7], Zhao and Zhu [19] and Pero et al. [16] built generative grammar to model the compositionality of Manhattan structures in the indoor scenes. By relaxing the Manhattan assumption, we generalize the grammar model to handle more complex and cluttered outdoor environment, and obtain better segmentation accuracy and reconstruction precision.

3. Attributed Grammar for 3D Scene

3.1. Local Manhattan World

In this work, we assume all the man-made scenes follow *Local Manhattan World* (LMW) assumption. the assumption suggests that each scene contains an ensemble of parallel lines or VPs which may not orthogonal to each other, each LMW only explain local regions. This observation goes beyond the classic Manhattan assumption [4] [13] which assume all parallel lines in a scene shall form one Cartesian coordinate system (CCS). We further assume that i) all the CCSs in the same scene share the same vertical axis, i.e. the vertical vanishing point (VP), and ii) every surface in the scene belongs to one and only one CCS. Thus, we could use VPs to indicate the surface orientation.

We adopt a simple procedure to discover the CCSs for the input image. Firstly, we use the method by Tretyak et al in [5] to detect the families of parallel lines and their associated vanishing points (VPs). This method also identifies one of the VPs as the vertical VP. All other VPs are considered as horizontal VPs. Second, based on the orthogonality between the vertical VP and horizontal VPs in the world space, we can estimate the camera focal length using the technique in [3]. Last, we adopt the proof by contradiction strategy to check if two horizontal VPs are orthogonal (see experiment for details).

3.2. Attribute Image Grammar

We first introduce the mathematic definition of attribute grammar used in our work. It is first proposed by Han et al. in [7] for image parsing and we extend it to integrate 3D spatial relationships between nodes. A attributed grammar is specified by a 5-tuple: $G = (V_N, V_T, S, R, P)$, where V_N is the set of non-terminal nodes, V_T is the set of terminal nodes, S is the initial root node for the whole scene, R is a set of production rules for spatial relationships, and P is the probability for the grammar. A non-terminal node is denoted by capital letter, $A_1, A_2 \in V_N$, and a terminal node is denoted by a lowercase letter , $a, b, c \in V_T$. Both non-terminal and terminal nodes have one vector of attributes $X(A)$ or $X(a)$ respectively.

We partition the input image into a set of superpixels and use them as the terminal nodes of the proposed attributed

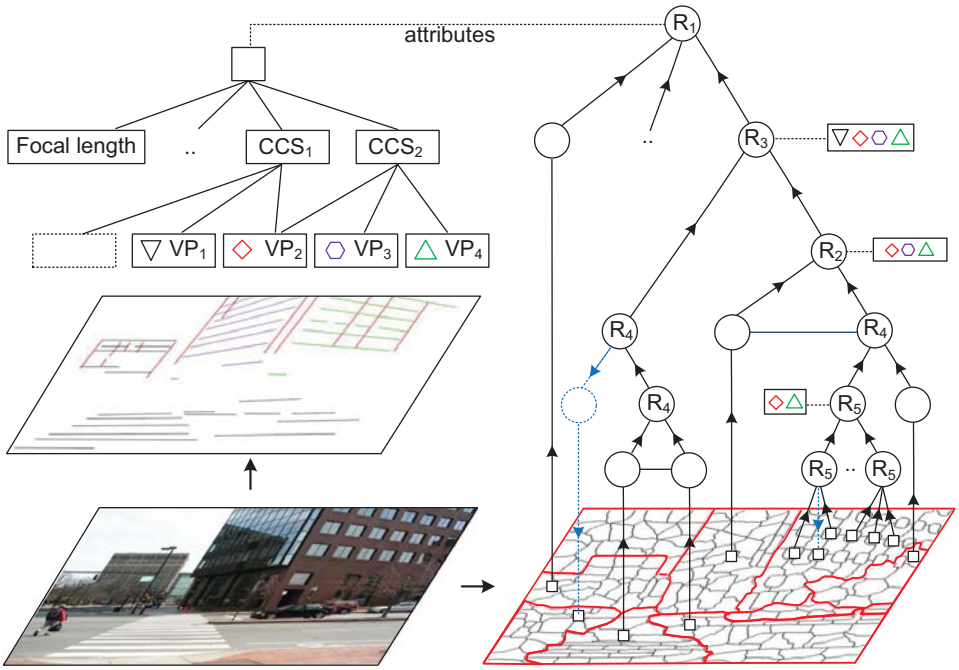


Figure 2. Parsing images using grammar rules

grammar, denoted as $V_T = \{(a, X(a) : X(a) \in \Omega_a\}$ The attributes of a terminal node is defined as: $X(a) = (u, v, h)$ where (u, v) is the central location im image, h is visual features extracted from this local region. We use the method by Ren et al. [17] to partition the image pixels into superpixels so each corresponds with only one geometric surface. There are about 200-300 suerpixels generated for each image.

3.3. Production Rules

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

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

The **siding rule** $R_2 : A \rightarrow (A_1, A_2)$ states two surfaces stand side by side in the 3D world. They usually belong to the same object (e.g. building). Since we assume all objects or surfaces in the scene stand on the ground, in the image, most of the siding surfaces are separated by a line passing through the vertical VP. Therefore, the attributes of A include: $X(A) = (u, v, \vec{l})$ where $\vec{l}$ is the parameters of the contact line between A_1 and A_2. The constraints for this rule include: 1) 1) A_1 and A_2 are spatially connected in image; 2) the normal direction of A_1 and A_2 are orthogonal to the vertical VP and 3) the contact line $\vec{l}$ should go through the vertical VP in image. Notice that this rule does not require the normals of these children surfaces to be orthogonal, in contrast with the work in [6].

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

The **affinity rule** $R_4 : A \rightarrow (A_1, A_2)$ states that two nodes have similar appearance and thus are likely to belong to the same surfaces. The attributes of A is defined as $X(A) = (u, v, h, \theta)$ where θ indicates the VP which is orthogonal to the normal of A. The default values of θ is

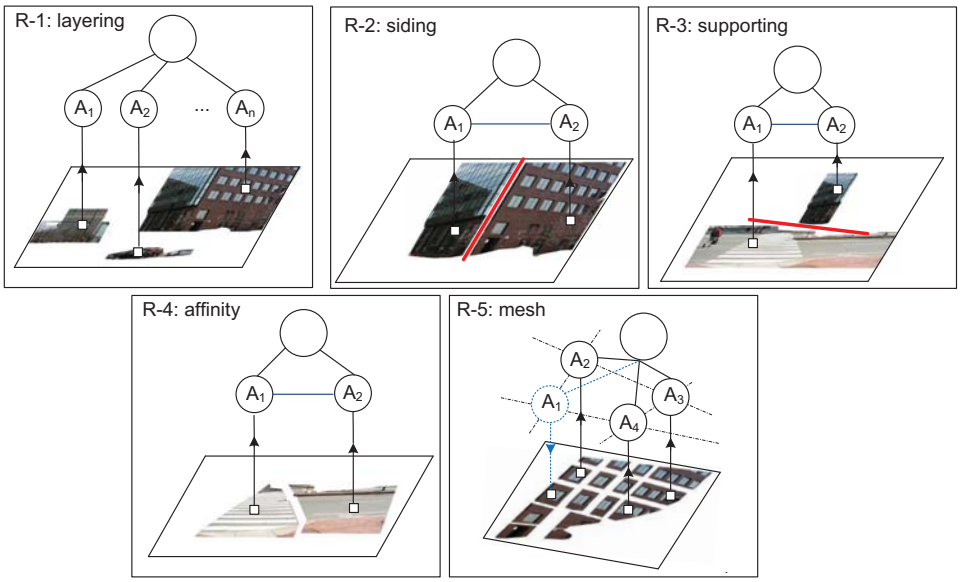


Figure 3. Illustration of the proposed five grammar rules

unknown. This rule requires that: 1) A_1 and A_2 are spatially connected in image; 2)A_1 and A_2 have the same surface normal except the normal of A_1 or A_2 is labeled as unknown;

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

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

This simple grammar can generate a large number of parse graphs for generic scenes. Every graph determines one layout segmentation by clustering the superpixels together according to the nodes of R_4, R_5 in the constructed parse graph. Fig. 2 shows one example of one layout segmentation in the bottom line. In addition, for two surfaces applied by the siding or the supporting rule, the union image regions of these two surfaces will be partitioned by the contact line into two parts each corresponding to one of the two surfaces. This projection from the parse graph to the configuration would help reducing the errors produced by the pre-step of superpixel partition. To obtain the optimal parse graph, we need to enforce constraints over the attributes of the parent nodes and those of the children nodes. In the next section, we introduce a Bayesian treatment for this problem to maximize a posterior probability.

4. Bayesian Formulation

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

$$G^* = \arg\max p(I|C)p(G)p(C) \tag{1}$$

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

4.1. Prior Model

The probability $p(C)$ is used to encourage the typical Ising/Potts prior used in the grouping problem [1], i.e., two neighbor sites tend to be grouped together. Let $\mathbf{c}(a)$ denote the region index or the color of a terminal node a in the layout segmentation, we define $p(C)$ as,

$$p(C) \propto exp\{\beta \sum_{<a,b>} \mathbf{1}(\mathbf{c}(a) = \mathbf{c}(b))\} \tag{2}$$

where $\mathbf{1}(\mathbf{c}(a) = \mathbf{c}(b)) = 1$ if $\mathbf{c}(a) = \mathbf{c}(b)$ for two adjacent superpixels otherwise it is zero. The highest probability is achieved when all vertices are the same color, or being

merged into one single region. β and Z are constants both of which can be determined from the training data.

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

$$p(G) \propto \prod_{A \in V_N} p(\ell(A))p(ch(A)|X(A), \ell(A)) \tag{3}$$

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

4.2. Likelihood Model

The likelihood model are defined on the layout segmentation of terminal nodes (i.e. superpixels) $p(I|C)$. We adopt the supervised model in [8] to classify regions to be one of the three geometric labels: ground, sky or vertical. Let j index the color of semantic region in C (clusters of superpixels), h_j the region features, v_j the region label determined by the method in [8], $\mathbf{c}(a)$ the color of the superpixel a, h_a the superpixel features. The probability $p(I|C)$ can be factorized as,

$$P(I|C) \propto \prod_j p(v_j|h_j) \prod_{\mathbf{c}(a)=j, \mathbf{c}(b)=j} p(\mathbf{c}(a) = j, \mathbf{c}(b) = j|h_a, h_b) \tag{4}$$

where $p(v_j|h_j)$ is the label confidence in the geometric label v_j, $p(\mathbf{c}(a) = j, \mathbf{c}(b) = j|h_a, h_b)$ is the probability that superpixels a and b have the same geometric label, i.e. homogeneity likelihood. We follow the work in [8] to implement these two terms.

5. Inference

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

Our algorithm bases on the Swendsen-Wang Cut algorithm [1] by Barbu and Zhu, which is essentially a cluster sampling procedure for graph coloring problem. It works on a adjacent graph iteratively following the MCMC design. At each step, it generates a cluster of connected component (CCP) by turning off the edges probabilistically, selects one CCP and changes the colors of the nodes in the selected CCP. The changes will be accepted with probability that usually integrates both the posterior probability and the proposal probability. The keys to the success of this clustering sampling algorithm include how to design propeor adjacent graphs and how to make the informative solution proposals effectively .

5.1. Ensemble of Augmented Adjacent Graphs

In this work, we use superpixels or terminal nodes as graph vertices, and link every two adjacent vertices to construct the adjacent graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$. Like SWCut method, we introduce an augment variable for each edge to indicate how likely two adjacent vertices should be assigned to the same color. We measure edge probabilities using different metrics following the grammar rules R_4 or R_5. These gives rise to an ensemble of augmented graphs which share the same graph structure $\{\mathbf{V}, \mathbf{E}\}$ but use different measurement of the augment variables.

For the appearance rule R_4 , the augment variables $\varphi = \{\varphi_e\}$represent the appearance similarity of two adjacent superpixels. We let $\varphi_e = exp(-\|h_a - h_b\|)$ where h_a and h_b are the visual features extracted from superpixel a and b, and $\|\|$ is the Euclidean norm of a vector.

For the mesh ruleR_5, an augment variable indicates if two superpixels are aligned to one of the VPs detected in image. If so, these two superpixels are likely to be merged by the mesh rule. Therefore, we introduce a set of augment variables denoted as $\phi_e = \{\phi_e^s\}$ for the s^{th} VP. For the edge $e =< a, b >$, ϕ_e^s is measured using following steps: i) estimate the boundary pixels between two superpixels a and b; ii) fit one straight line, denoted as $\vec{ab}$, from the boundary pixels by the typical Hough transform method; iii) compute the Cosine distance d_e^s between the line $\vec{ab}$ and the line linking the s^{th} VP to one of the end points of $\vec{ab}$. We set $\phi_e^s = 1 - \frac{1}{1-d_e^s}$ and normalize it by $\sum_{s=1}^{m} \phi_e^s$.

All the augmented graphs share the same graph structure $\{\mathbf{V}, \mathbf{E}\}$ as well as edge status (on or off). For each augmented graph, if two adjacent nodes have different labels, the edge between them will be turned off deterministically; otherwise the edge is turned off with the edge probability. The edges are turned on and off to group nodes into clusters in a dynamic way so that nodes in every cluster are strongly coupled.

Algorithm 1 Building Parsse Graph via Attributed Grammar .

1: **Input:** Single Image;
2: Initialization: partition input image into superpixels; detect families of parallel lines and VPs
3: Link every two neighbor superpixels with an edge to construct graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$
4: Calculate edge probability φ_e and $\{\phi^s\}$ where s is index of the horizontal VPs;
5: Initialize solution state randomly;
6: Iterate until convergence,

- For each CCP at the current status, estimate its normal and geometric class;
- Randomly select one of the grammar rules R_2, R_3, R_4,and R_5
- Make proposals according the select rule to change solution state
- Accept the change with a probability

5.2. Bottom-to-up and Top-down Proposals by Grammar Rules

Algorithm 1 summarizes our inference algorithm. Each iteration includes four major steps. In the first step, we turn off all the edges linking vertices of different colors to form multiple CCPs. For each CCP, we estimate its normal direction by i) firstly accumulating all the straight edges falling in this CCP and ii) selecting one VP that has the most supports in these edges. We also use the method in [10] to recognize geometrical class (ground,sky, or vertical) of each CCP. These two attributes are used for making proposals and computing the posterior probability. Other steps include randomly selecting one grammar rule, making proposals based on the rule, and accepting the change probabilistically. Let $q(G \to G')$ denote the proposal probability of moving from state G to G', the acceptance probability of the new state G' is defined based on the proposal probability and the posterior probability,

$$\min\left(1, \frac{q(G' \to G)p(G'|I)}{q(G \to G')p(G|I)}\right) \tag{5}$$

We make different proposals for different grammar rules. For the appearance rule R_4, we use the augmented graph $\{\mathbf{V}, \mathbf{E}, \varphi\}$ and sample the status of every edge e according to the edge probability φ_e. The edges of being 'on' will form multiple CCPs. Let V^c denote the selected CCP, $Cut(V^c|W)$ denote the set of edges turned off probabilistically around V^c, the proposal probability ratio for selecting V^c at states G and G' is defined as,

$$\frac{q(G' \to G)}{q(G \to G')} = \frac{\prod_{e \in Cut(V^c|G')}(1 - q_e)}{\prod_{e \in Cut(V^c|G)}(1 - \varphi_e)} \tag{6}$$

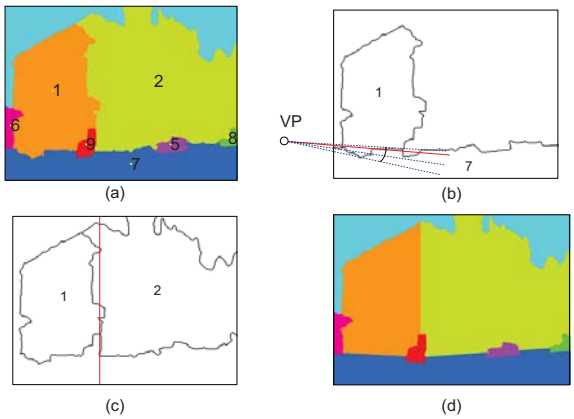


Figure 4. Line snapping for grammar rules R_2 and R_3

For the mesh rule R_5, we first randomly select one of the VPs, indexed by s and use the related augmented graph $\{\mathbf{V}, \mathbf{E}, \phi^s\}$ to sample the edge status. The proposals are made in the similar way as that for R_4.

To make proposals for the siding grammar rule R_2, we first collect all pairs of adjacent CCPs (or surfaces) at the current status, and randomly select one pair to generate one new node of R_4 in the parse graph. The attribute $\vec{l}$ of the new node is set as a line going through the vertical VP as well as one of the boundary pixels between these two surfaces. To speed up the sampling, we adopt two strategies: i) we down-sample the boundary pixels with an constant step (of 5 pixels) to reduce the number of candidate contact lines; ii) at each iteration, we greedily select the optimal contact line that achieves the highest acceptance probability.

For the supporting grammar rule R_3 , we first collect all pairs of adjacent CCPs and select one pair in which one CPP is classified to be ground, and the other is classified to be vertical, as aforementioned. Then, we adopt the same strategy as that for R_2 to add a new node in the parse graph or to change the attribute $\vec{l}$ of an existing node.

Fig. 4 illustrates the effects of contact line snapping used for R_3 and R_2. Fig. 4(a) show a layout segmentation for the photo shown in Fig. 6, where every surface is identified by a color and a number. Fig. 4(b) shows the candidates of contact lines (dotted lines) for a node of siding rule (surface 1 and the surface 7) in the parse graph. The optimal contact line at the current iteration is plotted in red. Fig. 4(c) shows one example for the supporting rule. Fig. 4(d) shows the segmentation map after applying these snapping process for all nodes of R_2 and R_3.

In addition to making proposals by the above bottom-to-up steps, we can also make proposals using the the top-down predictions by the grammar rules. For a non-terminal node A, the top-down prediction is essentially changing one of its children nodes (or removing/adding/changing a new children node for R_5) to other candidates . This top-down strategy can make proposals that are difficult to detect.

6. Experiments

In this section, we apply proposed algorithm to parse 3D scenes from single-view images, and evaluate it in both qualitative and quantitative ways.

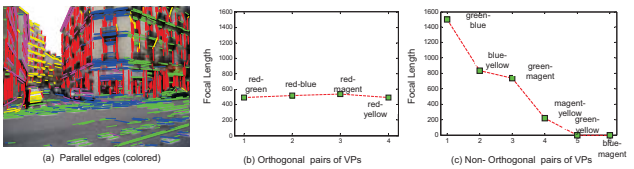
We use three datasets for evaluations. The first one is the CMU dataset collected by Hoiem et al [9] which consists of 300 outdoor images. Like [6], we use a subset of 100 images where the groundtruths of both occlusion boundaries and surface orientation are provided. The surfaces are labeled with three 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. Our method, in contrast, uses attributes of assigned VPs in Local Manhattan World to indicate the surface orientations, which relaxed the Manhattan assumption of 'left', 'center', 'right' labels. To compare with the baselines, we simply projected our detailed surface normal to 'left', 'center' and 'right' labels in the groundtruth. We used the first 50 images for training and the rest for testing as [6] did.

We further collect two datasets with manual annotation of vps, surface segmentation and surface orientation (represented by the correspondent VPs for each surface). The first one LMW-A consists of 50 images from the collections in [9], and there are 4.6 VPs per image on average in this dataset. The other dataset LMW-B consists of 50 images from the dataset of EurasianCities in [5] with 4.2 VPs per image on average. These two datasets, to our knowledge, are the first public benchmark to provide high-quality annotations for surface orientations in outdoor environment. Note that these two datasets are used for testing only while our model is trained on the CMU dataset.

To implement Algorithm 1, we use the training set to estimate the normalization parameters in $p(G|I)$ through the MLE method. We fix the maximum iterations to be 2000 in practice, and apply the top-down strategy to make proposals after 1000 iterations. For each superpixel or CCP (cluster of superpixels), we extracted the 78-bin feature descriptor (i.e. h_a) used in [9] which contains color, texture, location and shape, and geometry features.

We compare out method to two previous works, including the geometric parsing method by Hoiem et al. [9], the method by Gupta et al. in [6]. Both methods can recover the three main geometric classes and the five vertical subclasses, whereas only [9] ever reported the results of 3D reconstruction extensively. We use the default parameter configuration in their source codes

We first illustrate how to check the orthogonality between two horizontal VPs. We use the image shown in Figure 5(a), where one vertical VP and four horizontal VPs are detected. Figure 5(b) plots the focal length (vertical direction) estimated from the vertical VP and each of the four horizontal VPs. Observe that the estimated focal lengths



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

Figure 5. Focal length estimation. (a) Input image overlaid with parallel groups of edges (colored); (b) Focal length estimated by Orthogonal pairs of VPs; (c) Focal length estimated by non-orthogonal pairs of VPs.

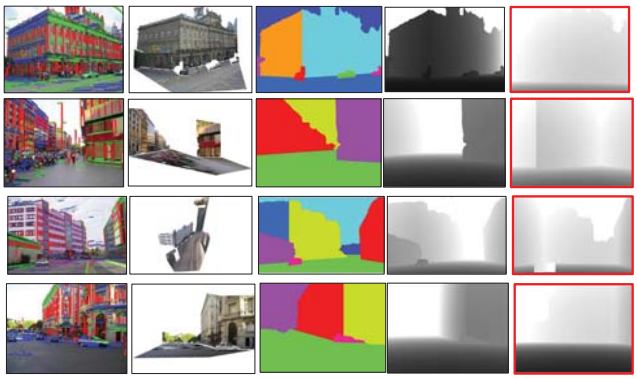


Figure 6. Exemplar results on the CMU dataset(first row) and the LMW-A dataset (othe rows). Columns 1-4 show our results, including: families of parallel lines; newly synthesized view; layout segmentation; and depthmap. Column-5 shows the depth map by Hoiem et al. [9] .

are roughly the same and the average focal length is 510 (unit). Figure 5(c) plots the focal length estimated from pairs of horizontal VPs by assuming they are orthogonal VPs, which are actually not. One could observe that none of them are close to the true focal length (i.e. 510 in this example). Therefore, for any pairs of horizontal VPs, we can estimate the focal length the method in [3] and then check if it is close enough to the previously estimated value.

Qualitative Evaluations We first compare the 3D models recovered by our method to that by [9]. Fig. 6 shows the results for two image in the CMU dataset [9] and dataset LMW-A. As shown, the image in the first row follows the typical Manhattan World assumption, while other images only follow the Local Manhattan World assumption as they contain more then 2 horizonal VPs or the horizontal VPs are not orthogonal with each other. For the first image, both [9] and our method can produce reasonable depth maps. For the other images, [9] tends to assign the same depth to the surfaces of 'vertical', whereas our method can still produce high-quality depth maps. These exemplar results well demonstrate how the LMW knowledge propagate through the grammar rules to create accurate 3D models.

Quantitative Results We further report the numerical comparisons between various methods in term of surface orientation estimation and region segmentation. For surface

orientation estimation, we use the metric of accuracy, calculated by the percentage of pixels that have the correct label and averaged over the test images. Since both our method and the two baselines can achieve high performance on the estimation of main geometric classes ('ground','vertical', and 'sky'), we focus on the vertical subclasses, like [6]. We discard the superpixels belonging to ground and sky and evaluate the performance of all methods. Table 1 reports the numerical comparisons. The method by Gupta et al. [6] has an average performance of 73.72%, whereas ours performs at 76.34% on the dataset by Hoiem et al [9]. On the other two datasets that have accurate surface orientation annotations, the improvements by our method are even more, i.e. 5.18 percentages and 4.1 percentages respectively. As stated by Gupta et al. [6] , improving vertical subclass performance is known to be hard. Our method, however, can improve these two baselines with large margins.

We also evaluate the segmentation performance on the three datasets. We use the best spatial support metric in [6], which first estimates the best overlap score of each ground truth segment and then averages it over all ground-truth segments. Table 2 report the numerical comparisons on the three datasets. Our method improves the method [6] with the margins of 3.86,5.24,4.86 percentages on the three datasets, respectively.

	dataset in [9]	LMW-A	LMW-B
Our method	76.34 %	67.9 %	64.3 %
Gupta et al. [6]	73.72 %	62.21 %	59.21 %
Hoiem et al. [9]	68.8 %	56.3 %	52.7 %

Table 1. Numerical comparisons on surface orientation

	dataset in [9]	LMW-A	LMW-B
Our method	72.71%	66.45%	65.14 %
Gupta et al. [6]	68.85%	59.21%	60.28%
Hoiem et al. [9]	65.32 %	58.37%	57.7 %

Table 2. Numerical comparisons on segmentation

7. Conclusions

This paper presents an attributed grammar for 3D scene parsing from a single view. We uses five grammar rules to generate the scene recursively, and introduce constraints equations on attributes of the rules to guide the construction of a parse graph. The developed inference method can fully exploit the constrained space efficiently by optimizing both the 2D surface segmentation and the attributes required for creating full 3D model. Hence, the obtained parse graph along its attributes can achieve high-quality 3D reconstruction results. Extensive evaluations on public benchmarks show that our method can achieve comparable performance to the state-of-the-art methods.

8. Acknowledgement

This work was supported by DARPA MSEE project FA 8650-11-1-7149, MURI grant ONR N00014-10-1-0933, and NSF IIS1018751.

References

[1] A. Barbu and S. Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *TPAMI*, 2007.

[2] M. Choi, A. Torralba, and A. Willsky. A tree-based context model for object recognition. *TPAMI*, 34(2):240–252, 2012.

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

[4] J. Coughlan and A. Yuille. Manhattan world: Orientation and outlier detection by bayesian inference. *Neural Computation*, 15(5):1063–1088, 2003.

[5] E.Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *IJCV*, 97(3):305–321, 2012.

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

[7] F. Han and S. Zhu. Bottom-up/top-down image parsing with attribute grammar. *TPAMI*, 31(1):59–73, 2009.

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

[9] D. Hoiem, A. Efros, and M. Hebert. Closing the loop on scene interpretation. In *CVPR*, 2008.

[10] D. Hoiem, A. Efros, and M. Hebert. automatic photo pop-up. *TOG*, 31(1):59–73, 2009.

[11] P. Koutsourakis, L. S. O. Teboul, G. Tziritas, and N. Paragios. Single view reconstruction using shape grammars for urban environments. In *ICCV*, pages 1795–1802. IEEE, 2009.

[12] J. Lafferty. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

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

[14] H. Mobahi, Z. Zhou, A. Yang, and Y. Ma. Holistic 3d reconstruction of urban structures from low-rank textures. In *Proceedings of the Internetional Conference on Computer Vision - 3D Representation and Recognition Workshop*, pages 593–600, 2011.

[15] N. Payet and S. Todorovic. Scene shape from textures of objects. In *CVPR*, 2011.

[16] L. D. Pero, J. Bowdish, E. Hartley, B. Kermgard, and K. Barnard. Understanding bayesian rooms using composite 3d object models. In *CVPR*, 2013.

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

[18] A. Saxena, M. Sun, and A. Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 2009.

[19] Y. Zhao and S. Zhu. Image parsing via stochastic scene grammar. In *NIPS*, 2011.