Inferring 3D Layout of Building Facades from a Single Image

Jiyan Pan
Google Inc.[†]
jiyanpan@google.com

Martial Hebert
Carnegie Mellon University
hebert@ri.cmu.edu

Takeo Kanade
Carnegie Mellon University
Takeo.Kanade@cs.cmu.edu

Abstract

In this paper, we propose a novel algorithm that infers the 3D layout of building facades from a single 2D image of an urban scene. Different from existing methods that only yield coarse orientation labels or qualitative block approximations, our algorithm quantitatively reconstructs building facades in 3D space using a set of planes mutually related by 3D geometric constraints. Each plane is characterized by a continuous orientation vector and a depth distribution. An optimal solution is reached through inter-planar interactions. Due to the quantitative and plane-based nature of our geometric reasoning, our model is more expressive and informative than existing approaches. Experiments show that our method compares competitively with the state of the art on both 2D and 3D measures, while yielding a richer interpretation of the 3D scene behind the image.

1. Introduction

Given a single image of an urban scene, automatically inferring the underlying 3D layout of building facades in the scene would significantly benefit many tasks in fields such as autonomous navigation and augmented reality. It goes beyond depth map estimation [22, 23, 19, 15], because it provides a richer understanding of the scene, such as camera pose, locations of planes and blocks, and how they are related with each other in 3D [6].

Nevertheless, recovering building facades in 3D space is a particularly challenging task. The difficulty comes from the fact that building facades could have highly flexible combinations in 3D space, and therefore do not have a definitive shape either in 2D image or 3D space. In fact, they are "stuffs"rather than "objects", and are thus unable to be located by a single 3D coordinate. An example is shown in Figure 1, where the facades do not even follow the Manhattan world assumption.

Although we cannot locate building facades in the same way we locate objects, we observe that unlike other regions such as trees or sky, building facades are more structured, and can be decomposed into a set of planes that can be represented quantitatively. The orientations and locations of those planes are mutually constrained by their 3D geometric relationships derived from physical plausibility. In this paper, we model building facades as a set of planes with continuous orientations, and then quantitatively reason over their 3D locations using inter-planar geometric constraints. This approach would produce a richer interpretation of the facade scene than existing pixel/segment based approaches [11, 12, 19] and block-based approaches [6]. More specifically, our approach is able to provide critical scene understanding information (*e.g.* quantitative orientation, depth, and relationships of facade planes) that existing algorithms do not provide.

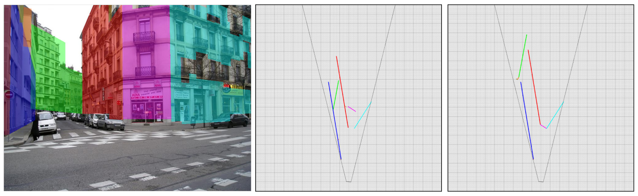

Figure 1. Our algorithm detects building facades in a single 2D image, decomposes them into distinctive planes of different 3D orientations, and infers their optimal depth in 3D space based on cues from individual planes and 3D geometric constraints among them. Left: Detected facade regions are covered by shades of different colors, each color representing a distinctive facade plane. Middle/Right: Ground contact lines of building facades on the ground plane before/after considering inter-planar geometric constraints. The coarser grid spacing is 10m.

The main contributions of this work are as follows. 1) We propose a plane-based fully quantitative model to infer the 3D layout of building facades, where each plane is represented by a continuous orientation vector and a distribution of depth values. 2) In such a model, multiple cues, such as semantic segmentation, surface layout, and vanishing lines, are utilized to detect and decompose the building region into distinctive planes. 3) The quality of an individual candidate plane is determined by its compatibility with both 2D evidence from image features and 3D evidence such as camera and building height. 4) We model different types of 3D geometric relationships among candi-

[†]Jiyan Pan was at the Robotics Institute, Carnegie Mellon University when the work was performed.

date planes, and apply a CRF to both determine their validity and infer their optimal depths. 5) We do not assume ground is horizontal or buildings are vertical with respect to the camera, nor do we take the Manhattan world assumption.

2. Related works

Pioneering works on modeling facade geometries, such as Geometric Context proposed by Hoiem *et al.*, focus on classifying super-pixels into different orientation labels (*e.g.* planar left, planar center, or planar right) [11, 14, 12, 5]. While this approach does not produce higher-level concepts such as planes and blocks, it generates useful cues of coarse surface orientations. Using such cues, Gupta *et al.* assemble blocks from two adjacent segments with different orientation labels, and locate those blocks by fitting their ground and sky contact lines [6]. This method generates a rich high-level interpretation of the scene, yet the interpretation is qualitative both in facade orientation and depth. Besides, approximating building facades by blocks cannot model more complex cases such as the one shown in Figure 1.

Quantitative modeling of surface orientation can be found in many works of indoor scene understanding [17, 3, 25, 8, 7, 24], where surface orientation is determined by the span of two orthogonal vanishing directions [1, 2, 4]. While we also use vanishing directions to compute plane orientations, the methods developed for indoor scene understanding cannot be applied directly to outdoor facade analysis. This is because those methods typically simplify the room as a box, and all the other vertical surfaces are confined within the box and parallel to the box walls. By contrast, building facades in outdoor scenes are located in an open space and usually have more flexible structures. While the algorithm in [18] does not simplify the room to a box, it heavily relies on a common ceiling to define vertical walls. This is not applicable to outdoor scenes either.

Instead of modeling surfaces, Ramalingam *et al.* constructs a wire frame model of building facades by lifting vanishing lines into 3D space using orthogonality constraints derived from a strong Manhattan-world assumption [21]. A limitation of this approach is that, even when the Manhattan-world assumption holds true, all the building facades have to be connected in a single wire frame model. In many cases, however, building facades occlude each other and do not form a connected structure. This is exemplified by the facades in red and green in Figure 1.

3. Plane-based 3D modeling of building facades

3.1. Problem formulation

Before formally stating the problem, we first define the geometric variables involved in our work. The coordinate systems we use are shown in Figure 2. Variables in red letters are defined with respect to the camera coordinate system $\{o, x, y, z\}$ whose origin is at the camera center. Variables in blue letters are defined with respect to the ground coordinate system $\{O, X, Z\}$, whose origin is on the ground plane. Variables in green letters are defined in the image plane. Note that the camera and ground coordinate systems are related in such a way that the Z axis of the ground coordinate system lies at the intersection between the ground plane and the plane spanned by $\mathbf{n_g}$ and the z axis of the camera coordinate system. The ground plane is parameterized by the ground orientation $\mathbf{n_g}$ and the ground distance h_g, and each facade plane involves the following geometric variables: surface orientation $\mathbf{n_s}$, distance from camera d_s, ground contact line orientation $\mathbf{n_s'}$, ground contact line distance d_s', 3D coordinates of the facade plane corners $\mathbf{X_s^{(1)}} \sim \mathbf{X_s^{(4)}}$ and their projections on the image plane $\mathbf{x_s^{(1)}} \sim \mathbf{x_s^{(4)}}$, ground coordinates $\mathbf{X_s^{'(1)}}, \mathbf{X_s^{'(2)}}$ of the two endpoints of the ground contact line, and real world height H_f.

Our model addresses the problem of detecting a set of distinctive facade planes and estimate their 3D orientations and locations given a single 2D image of urban scene. Here, a distinctive facade plane is defined as a facade plane whose orientation is different from the orientations of its adjacent facade planes; otherwise, two adjacent facade planes with the same orientation will be merged. More formally, our model infers the optimal 3D layout of building facades by maximizing the following objective function:

$$
\begin{aligned}
&V(\mathbf{o}, \mathbf{n_s}, \mathbf{d_s}, \mathbf{x_s} | I, h_g, \mathbf{n_g}, f, H_f) \\
&\quad = \sum_{i \in \mathbf{P}} \omega(o_i, \mathbf{n_{si}}, d_{si}, \mathbf{x_{si}} | I, h_g, \mathbf{n_g}, f, H_f) \\
&+ \sum_{(i,j) \in \mathbf{P_v}} \varphi_v(o_i, o_j, d_{si}, d_{sj}, \mathbf{n_{si}}, \mathbf{n_{sj}}, \mathbf{x_{si}}, \mathbf{x_{sj}} | h_g, \mathbf{n_g}, f) \\
&+ \sum_{(i,j) \in \mathbf{P_o}} \varphi_o(o_i, o_j, d_{si}, d_{sj}, \mathbf{n_{si}}, \mathbf{n_{sj}}, \mathbf{x_{si}}, \mathbf{x_{sj}} | h_g, \mathbf{n_g}, f) \\
&+ \sum_{(i,j) \in \mathbf{P_a}} \varphi_a(o_i, o_j, d_{si}, d_{sj}, \mathbf{n_{si}}, \mathbf{n_{sj}}, \mathbf{x_{si}}, \mathbf{x_{sj}} | h_g, \mathbf{n_g}, f),
\end{aligned}
\tag{1}
$$

where $\mathbf{o}, \mathbf{n_s}, \mathbf{d_s}, \mathbf{x_s}$ are variables that characterize facade planes: for each plane i, variables $o_i, \mathbf{n_{si}}, d_{si}$ and $\mathbf{x_{si}}$ represent its validity (binary indicator), orientation (continuous vector), distance from camera center (continuous scalar), and spatial extent (continuous coordinates specifying the corners of the plane in the image), respectively. Please see Figure 2 for an illustration of some of those variables. The optimization problem is conditioned upon image features I, ground distance h_g from the camera center, ground orientation $\mathbf{n_g}$ with respect to the camera, focal length f, and facade height H_f. In this work, H_f is obtained from the prior knowledge of a typical range of facade heights, and

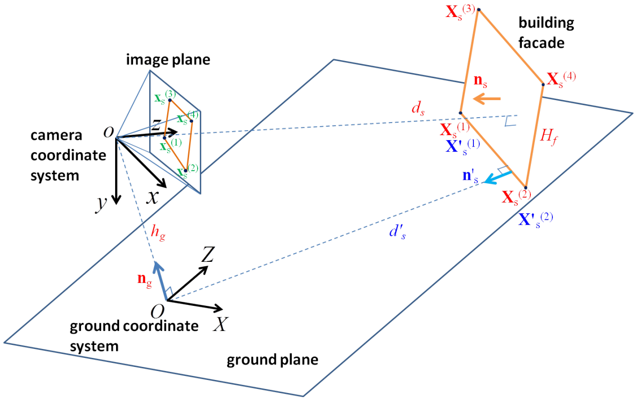

Figure 2. Geometric variables in the camera and ground coordinate systems. Please zoom in for a better view.

h_g is assumed to be 1.6m. $\mathbf{n_g}$, which gives rise to the horizon line, is automatically computed from vanishing lines using an approach similar to [16] except that we remove their Manhattan world assumption, resulting in the output of a vertical vanishing direction and multiple (could be more than 2) horizontal vanishing directions. $\mathbf{n_g}$ is then determined by the vertical vanishing direction, and f is estimated by maximizing the orthogonality between the vertical and horizontal vanishing directions.

The first term in the objective function is a unary potential for each individual plane, and it is summed over all candidate planes $\mathbf{P}$. The remaining three terms are pairwise potentials for planes with mutual constraints, and they are summed over a subset (*i.e.* $\mathbf{P_v}$, $\mathbf{P_o}$, or $\mathbf{P_a}$) of candidate planes involved in those constraints. We will describe these potentials in more detail in the remainder of this section.

As directly optimizing the objectivie function in Equation 1 is intractable, we first generate a set of candidate facade planes using the quadrilateral-based sampling algorithm described in Section 4, where each candidate facade plane has a fixed normal $\mathbf{n_{si}}$ and boundary $\mathbf{x_{si}}$. With given $\mathbf{n_{si}}$ and $\mathbf{x_{si}}$, we only need to optimize over the validity o_i and depth d_{si} of each candidate facade plane. The total number of valid facade planes, which is unknown in advance, will also be obtained in this process.

3.2. Individual compatibility

The unary potential ω is based on the product of two scores. The first score is the **image feature compatibility score**, measuring how well the 2D location of a facade plane in the image agrees with image features:

$$S_r(\mathbf{n_{si}}, \mathbf{x_{si}}|I) = \frac{1}{N(\mathbf{Q})} \sum_{p \in \mathbf{Q}} (s_p^{(bldg)} + g_p(\mathbf{n_{si}}) + < \mathbf{n_{si}}, \mathbf{v}_p >), \tag{2}$$

where $\mathbf{Q}$ is the image region of the facade plane defined by its corners $\mathbf{x_{si}}$, and N is the number of pixels within $\mathbf{Q}$. We consider three sources of information. 1) $s_p^{(bldg)}$ is the semantic label score of pixel p belonging to “building”region.

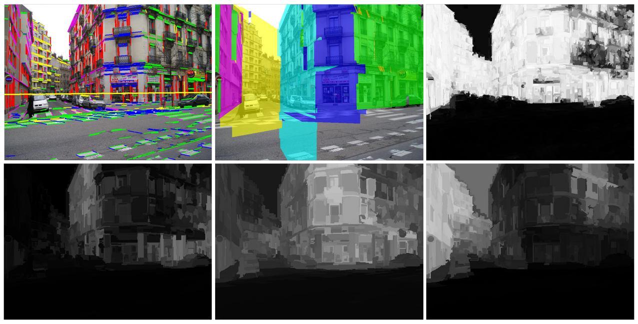

Figure 3. Image features utilized by our method. Top row from left to right: 1) vanishing lines and horizon line, where the thick yellow line is the horizon line, and the thiner lines in different colors represent vanishing lines belonging to different vanishing directions, 2) orientation map from the line-sweeping algorithm where each color represents an orientation corresponding to a specific horizontal vanishing direction, 3) semantic segmentation score for the “building”region. Bottom row from left to right: surface orientation scores for “planar left”, “planar center”, and “planar right”.

We use Stacked Hierarchical Labeling proposed by Munoz *et al.* [20] to perform semantic segmentation. 2) $g_p(\mathbf{n_{si}})$ is the orientation label score of pixel p having an orientation label consistent with the orientation $\mathbf{n_{si}}$ of the plane. Orientation labels are obtained from the surface layout algorithm proposed by Hoiem *et al.* [11]. The orientation labels of interest here include “planar left”, “planar center”, and “planar right”. Plane orientation $\mathbf{n_{si}}$ is quantized into one of these three labels before computing the score g_p using the orientation label distribution of pixel p produced by Hoiem’s surface layout algorithm. 3) $< \mathbf{n_{si}}, \mathbf{v}_p >$ computes the inner product between the plane orientation $\mathbf{n_{si}}$ and the orientation vector $\mathbf{v}_p$ at pixel p derived from vanishing lines using the line-sweeping algorithm proposed by Lee *et al.* [18]. An example of the image features we use are shown in Figure 3 [*]. Note that the three terms in the right-hand side of Equation 2 are comparable because they all range from 0 (totally implausible) to 1 (totally plausible).

The intuition behind Equation 2 is that if an image region indeed belongs to a building facade, then it should 1) be supported by the semantic cue that it belongs to the “building”region, 2) be supported by the surface layout cue that its orientation agrees with the dominant orientation label within it, and 3) be supported by the vanishing line cue that its orientation is consistent with the dominant horizontal vanishing direction within it.

The second score is the **geometric compatibility score**. Although the facade detection algorithm to be described in Section 4 returns the image region occupied by a facade plane, its ground contact line is often occluded (*e.g.* by cars parked along the road). As a result, the bottom boundary of the facade plane in the image is usually invisible and is

[*] Here we do not limit the line-sweeping region to be contained within the “building”semantic region.

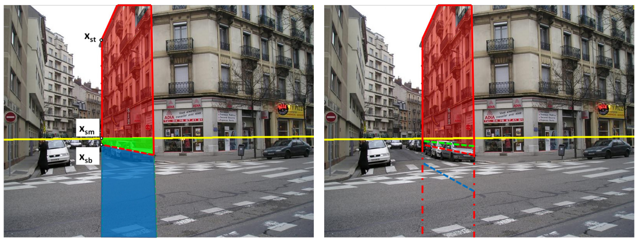

Figure 4. Illustration of the situation when computing the geometric compatibility score S_d. The yellow line is the horizon line. The left image show the relevant regions we consider. The right image shows the ground contact lines when the plane is hypothetically placed at different depths.

therefore flexible, as is shown in the right image of Figure 4. When we place the facade plane (with a fixed orientation $\mathbf{n_{si}}$ returned by the facade detection algorithm in Section 4) at different depths, it would result in a series of ground contact lines sweeping the blue and green regions illustrated in the left image of Figure 4. Not all depths are equally plausible. For example, if the facade plane is placed too close to the camera, the ground contact line would be the blue line in the right image of Figure 4. Such a depth is geometrically implausible because there exists ground region above the ground contact line. On the other hand, if the ground plane is placed too far away from the camera, the ground contact line would be the green line in the right image of Figure 4. Such a depth is also geometrically implausible because the resulting 3D height of the facade would be unreasonably large. Based on these intuitions, we compute the geometric compatibility score of a facade plane with orientation $\mathbf{n_{si}}$ placed at a certain depth d_{si} by checking if the ground contact line of the facade plane in the image is *both* above the ground region *and* below the building region. In addition, it checks if the distance between the ground contact line and the horizon line in the image yields a reasonable 3D height of the facade:

$$S_d(\mathbf{n_{si}}, d_{si}, \mathbf{x_{si}}|I, h_g, \mathbf{n_g}, f, H_f) =$$
$$\min\{1 - \frac{1}{N(\mathbf{A})}\sum_{p\in\mathbf{A}} s_p^{(gnd)}, 1 - \frac{1}{N(\mathbf{B})}\sum_{p\in\mathbf{B}} s_p^{(bldg)},$$
$$H_f(\frac{\|\mathbf{x_{st}} - \mathbf{x_{sb}}\|}{\|\mathbf{x_{sm}} - \mathbf{x_{sb}}\|} \cdot h_g))\}, \quad (3)$$

where $\mathbf{A}$ is the region between the ground contact line and the horizon line as is illustrated by the green region in Figure 4. This regions is not supposed to contain ground pixels. $\mathbf{B}$ is the region between the ground contact line and the bottom of the image as is illustrated by the blue region in Figure 4. This region is not supposed to contain building pixels. h_g, $\mathbf{n_g}$, and f are used to project the 3D ground contact line of the facade plane with orientation $\mathbf{n_{si}}$ at depth d_{si} to the 2D ground contact line in the image. $\mathbf{x_{si}}$ defines the boundaries of the facade plane in the image. As Figure 4 illustrates, $\mathbf{x_{st}}, \mathbf{x_{sb}}$ and $\mathbf{x_{sm}}$ are where the left boundary intersects with the top boundary, the (projected) ground contact line, and the horizon line, respectively. $\|\mathbf{x_{st}} - \mathbf{x_{sb}}\|/\|\mathbf{x_{sm}} - \mathbf{x_{sb}}\| \cdot h_g$ is an estimate of the facade height in 3D. It should be within a reasonable range H_f of typical facade heights. Note that the three terms in the right-hand side of Equation 3 are comparable because they all range from 0 (totally implausible) to 1 (totally plausible).

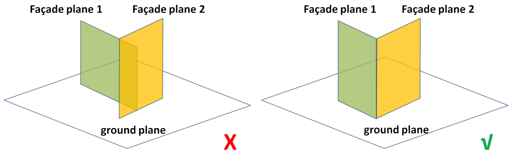

Figure 5. Facade planes that are adjacent in the image and form a convex corner in 3D must connect with each other along the convex fold.

After the two scores S_r and S_d are obtained, the unary potential ω is defined as

$$\omega(o_i, \mathbf{n_{si}}, d_{si}, \mathbf{x_{si}}) =$$
$$\begin{cases} S_r(\mathbf{n_{si}}, \mathbf{x_{si}}) \cdot S_d(\mathbf{n_{si}}, d_{si}, \mathbf{x_{si}}) - 0.25 & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases} \quad (4)$$

Here, the conditioning variables are ommitted for the sake of clarity. The product of S_r and S_d is subtracted by 0.25 because the neutral values of both S_r and S_d are 0.5. When $S_r \cdot S_d < 0.25$, the unary potential penalizes the objective function if $o_i = 1$, indicating the candidate plane is unlikely to be valid based on its individual compatibility cues.

3.3. Mutual compatibility

As building facades are structured, geometric constraints exist among facade planes. The first type of constraints we consider is **convex-corner constraint**: if two facade planes are adjacent in the image *and* their orientations form a convex corner in 3D, then their depths should be such that they connect in 3D along the convex fold [6], as is illustrated in Figure 5. As a result, all the facade planes connected by convex corners (such as the red, magenta, and cyan planes in Figure 1) could only move in space as a whole. To enforce such a constraint, the pairwise potential φ_v is defined as

$$\varphi_v(o_i, o_j, d_{si}, d_{sj}, \mathbf{n_{si}}, \mathbf{n_{sj}}, \mathbf{x_{si}}, \mathbf{x_{sj}}|h_g, \mathbf{n_g}, f) =$$
$$\begin{cases} -\infty & \text{if } o_i \cdot o_j \neq 0 \text{ and } \mathbf{x_{si}}|\mathbf{x_{sj}} \text{ and } \mathbf{n_{si}} \vee \mathbf{n_{sj}} \text{ and } d_{si} \veebar d_{sj} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\mathbf{x_{si}}|\mathbf{x_{sj}}$ means facade planes i and j are adjacent in the image, $\mathbf{n_{si}} \vee \mathbf{n_{sj}}$ means the orientations of facade planes i and j form a convex corner, and $d_{si} \veebar d_{sj}$ denotes the situation in which the depths of the two planes are such that they fail to connect along the convex fold in 3D. Negative infinity penalty is applied if this situation happens. In other

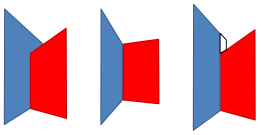

Figure 6. Three cases for determining the occlusion ordering between two facade planes that are adjacent in the image and form a concave corner in 3D. Please see text for details.

words, the convex-corner constraint is a hard constraint, so that physical plausibility is always strictly maintained.

The second type of constraints is **occlusion constraint**. If two facade planes are adjacent in the image *and* their orientations form a concave corner, we check if one of them is occluded by the other. Figure 6 illustrates three cases when two facade planes form a concave corner (other cases are symmetrical to one of these three cases). In the left case, it can be immediately determined that the blue plane is occluded by the red one. In the center case, we cannot say for sure which plane is occluded. In the right case where the blue plane does not extend beyond the red one, we check the vanishing lines and orientation map within a small region (outlined by the black lines in Figure 6) right next to the blue plane. If the region has the same orientation as the blue plane, then it becomes the left case; otherwise it becomes the center case. Although we cannot unambiguously determine the occlusion ordering for the center case, we could still reasonably assume that the plane whose orientation is more frontal is usually occluded, such as the red-green facade plane pair in Figure 1. While such an assumption could be violated, it holds true for most of the outdoor urban scenarios we have encountered. Also, as we will see in the equation below, the occlusion constraint is a soft constraint, meaning it could be overridden by other stronger evidence. Without loss of generality, let's suppose facade plane i is occluded by facade plane j. Then the pairwise potential φ_o related to occlusion ordering is defined as

$$\varphi_o(o_i, o_j, d_{si}, d_{sj}, \mathbf{n_{si}}, \mathbf{n_{sj}}, \mathbf{x_{si}}, \mathbf{x_{sj}} | h_g, \mathbf{n_g}, f) = \\ \begin{cases} -\frac{\max\{0, d'_{j1}, d'_{j2}\}^2}{2\sigma_1^2} & \text{if } o_i \cdot o_j \neq 0 \text{ and } \mathbf{x_{si}} | \mathbf{x_{sj}} \text{ and } \mathbf{n_{si}} \wedge \mathbf{n_{sj}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Here, $\mathbf{n_{si}} \wedge \mathbf{n_{sj}}$ means the orientations of facade planes i and j form a concave corner. d'_{j1} and d'_{j2} are defined in Figure 7, where the solid blue and red lines are the ground contact lines of facade planes i and j, respectively, in the *ground* coordinate system. According to equation 6, the pairwise potential φ_o is 0 (*i.e.*, no penalty) when facade plane i is totally behind facade plane j; otherwise, a soft penalty is applied depending on the degree of violation. The soft constraint here serves to handle noise in the estimation of occlusion ordering.

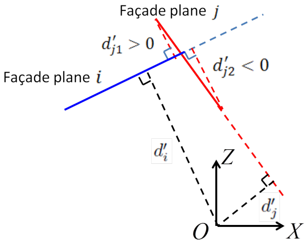

Figure 7. Computing the pairwise potential φ_o related to occlusion ordering. The solid red and blue lines are the ground contact lines of the two facade planes in the *ground* coordinate system. d'_{j1} and d'_{j2} are signed distances from the two endpoints of the red line to the blue line.

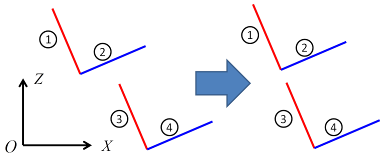

Figure 8. Illustration of the alignment constraint. The configuration on the right is preferred over the one on the left. Note that facade planes 2 and 4 are not encouraged to be aligned because there exists facade plane 3 that results in the violation of the non-occlusion criterion.

The third type of constraints is **alignment constraint**: a pair of nearby facade planes are encouraged to reside in a common plane if they have the same orientation, except when there exists a third facade plane that is 1) connected to the closer plane in the pair through convex corners, and 2) occludes the farther plane in the pair. Please see Figure 8 for illustration. Such a constraint is based on the observation that street-facing facades of different buildings are usually aligned along the street. If facade planes i and j satisfy both the same-orientation and non-occlusion criteria, the pairwise potential φ_a encoding the alignment constraint between them is defined as

$$\varphi_a(o_i, o_j, d_{si}, d_{sj}, \mathbf{n_{si}}, \mathbf{n_{sj}}, \mathbf{x_{si}}, \mathbf{x_{sj}} | h_g, \mathbf{n_g}, f) = \\ \begin{cases} -\frac{(d'_i - d'_j)^2}{2\sigma_2^2} & \text{if } o_i \cdot o_j \neq 0 \text{ and } \mathbf{n_{si}} = \mathbf{n_{sj}} \text{ and } Q_s(i,j) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $Q_s(i,j) = 0$ means the two facade planes satisfy the non-occlusion criterion. d'_i and d'_j are the distances from the ground origin to the ground contact lines of facade planes i and j, respectively, as is illustrated in Figure 7. The alignment constraint is also a soft constraint since we are just *encouraging* eligible planes to be aligned; other strong evidence could override such an encouragement.

The benefit of incorporating geometric constraints among facade planes is evident in Figure 1. After interplanar geometric constraints are imposed, planes that form convex corners are correctly connected together, and planes in the background are correctly pushed backward.

4. Implementation details

Optimizing the objective function in Equation 1 is challenging, because it is generally non-convex and involves binary, discrete and continuous variables, resulting in a huge search space. In addition, how do we acquire the candidate facade planes in the first place? In this section, we describe a quadrilateral-based sampling algorithm that detects a set of candidate facade planes, each of which comes with a 3D orientation $\mathbf{n_{si}}$ and corner coordinates $\mathbf{x_{si}}$. Using the output of this algorithm, we are able to optimize Equation 1 in a tractable way.

4.1. Quadrilateral-based sampling algorithm

We detect candidate facade planes via a sequential sampling of "quadrilaterals". A sample of a quadrilateral is formed by a random pair of vertical vanishing lines and a random pair of horizontal vanishing lines in the image (please see Figure 9 for examples). The 3D orientation of a quad (short for quadrilateral) is equal to the cross-product of the vertical and horizontal vanishing directions. A quad itself could be a facade plane, or multiple adjacent quads with the same orientation could merge and form a larger facade plane.

In the quadrilateral-based sampling algorithm, the pool of selected quads is first initialized as an empty set. To add a new quad to the pool, we sample a large number of quads and evaluate each sample using the image feature compatibility score S_r by Equation 2. The top k quads, which come with known orientation $\mathbf{n_{si}}$ and spatial extent $\mathbf{x_{si}}$, are selected and further evaluated by computing the maximum geometric compatibility score $\hat{S}_d$ over a set of quantized hypothetical depth d_{si} using Equation 3. The quad with the highest hybrid score $S_r \cdot \hat{S}_d$ is added to the pool of selected quads, and the region occupied by the quad is no longer available for subsequent sampling. This process is repeated until the hybrid score of the best available quad is below 0.5. We finally merge those adjacent quads with the same orientation into a single distinctive candidate facade plane. An example of the quadrilateral-based sampling process is shown in Figure 9.

Note that the purpose of keeping the top k (more than one) quads after evaluating S_r scores is to reduce the risk of being trapped in local extrema while reducing computational complexity. Empirically, we found that when the search width k is greater than or equal to 4, performing multiple starts does not bring additional improvement over a single start.

4.2. Tractable inference

In our approach, the key to make the inference tractable is to decouple $\mathbf{n_s}, \mathbf{x_s}$ from $\mathbf{o}, \mathbf{d_s}$ in Equation 1. When the quadrilateral-based sampling process is completed, we already have a set of candidate facade planes with known orientation $\mathbf{n_{si}}$ and spatial extent $\mathbf{x_{si}}$. The only variables that remain to be inferred are the validity indicator o_i and depth d_{si}. The objective function in Equation 1 is therefore reduced to

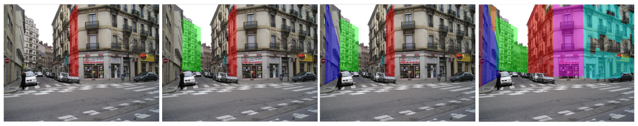

Figure 9. When detecting candidate facade planes using the quadrilateral-based sampling algorithm, the best quads are selected and added to the candidate pool one by one. From left to right, the four images show the first three selected quads, as well as all the selected quads when the search process is completed, respectively. Quads belonging to the same candidate distinctive facade plane share the same color.

$$V(\mathbf{o}, \mathbf{d_s}) = \sum_{i \in \mathbf{P}} \omega(o_i, d_{si}) + \sum_{(i,j) \in \mathbf{P_v}} \varphi_v(o_i, o_j, d_{si}, d_{sj}) + \sum_{(i,j) \in \mathbf{P_o}} \varphi_o(o_i, o_j, d_{si}, d_{sj}) + \sum_{(i,j) \in \mathbf{P_a}} \varphi_a(o_i, o_j, d_{si}, d_{sj}), \tag{8}$$

where we have omitted the conditioning variables for the sake of clarity. Suppose we have quantized depth d_{si} into M bins, then the number of possible states over o_i and d_{si} is only $M + 1$. This is significantly more tractable than the original optimization problem. We construct a CRF to encode all the individual and mutual compatibilities according to Equations 4, 5, 6, and 7, and perform max-product belief propagation [9] to obtain the validity of each candidate facade plane and the optimal depth of each *valid* plane, as well as a MAP distribution over its possible depths.

5. Experiments

We evaluate the performance of our facade reasoning algorithm on three challenging datasets. All the parameters of our algorithm are fixed throughout our experiments, where the camera height h_g is assumed to be 1.6m, $\sigma_1 = 0.5$ in Equation 6, $\sigma_2 = 50$ in Equation 7, the number of random quad samples in Section 4.1 is 2000, the number k of top quads in Section 4.1 is set as 4, and the reasonable range of building height H_f is assumed to be between 5m and 30m, with a standard deviation of 1m on the lower end, and 10m on the higher end. On average, it takes about 20 minutes to process a 800-by-600 image with Matlab code.

5.1. 3D facade layout

The key strength of our approach is that it is able to produce a richer interpretation of a facade scene – a 3D layout of facade planes with continuous orientations. We apply our facade reasoning algorithm on the LabelMe dataset [13]. As the dataset does not provide the ground truth for a plane-wise decomposition of building facades or their 3D layout,

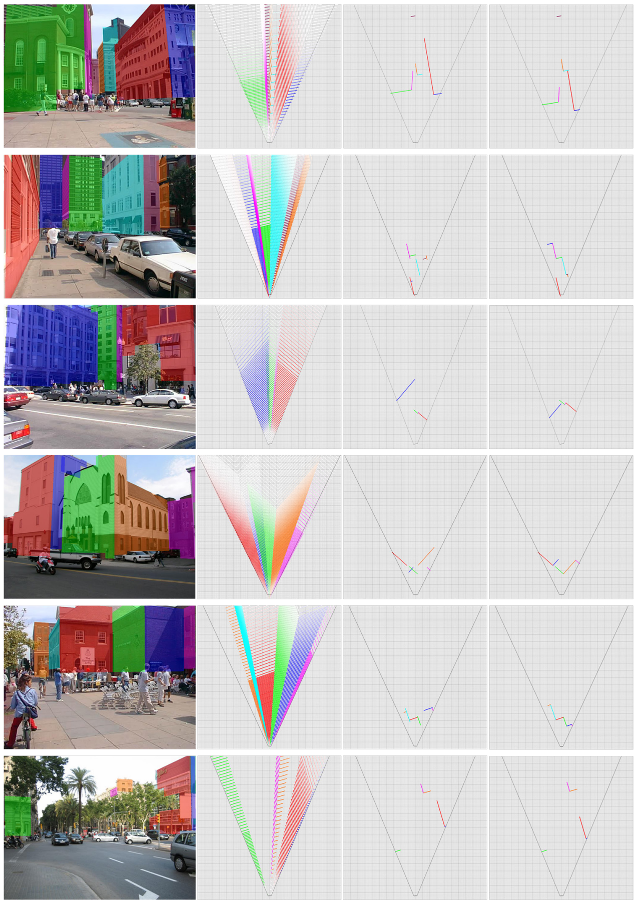

Figure 10. 3D facade layout estimation by our method on images from the LabelMe dataset [13]. Left to right: 1) Original image overlaid with color shades representing quads from distinctive facade planes. Quads from the same distinctive facade plane share the same color. 2) Depth distribution of candidate distinctive facade planes before running the CRF inference. 3) Best locations of candidate distinctive facade planes before running the CRF inference. 4) Optimal locations of valid distinctive facade planes after running the CRF inference. The viewing boundary is marked with black lines, and the coarser grid spacing is 10m.

we evaluate our approach qualitatively. Typical examples of success cases are shown in Figures 10. Comparing columns 3 and 4, we can see that imposing inter-planar geometric constraints significantly regulates the depths of distinctive facade planes and results in a meaningful interpretation.

In the LabelMe dataset [13], many complex facade structures can be found – many building facades are occluded by other objects or mutually occluded, and in some cases they are not Manhattan (*e.g.* row 1 in Figure 10 and Figure 1). As our method makes no assumption on those conditions, they do not pose a problem. We also do not assume all building facades are inter-connected as [21] does. Therefore, many street scenes where adjacent buildings are separated by streets can be modeled by our approach (*e.g.* rows 1 3, and 4 in Figure 10). In row 6 of Figure 10, we could also see that distinct facade planes having the same orientation (the red and magenta planes) are aligned after CRF inference. However, when other stronger evidence is present as is the case in row 5 of Figure 10, they are not blindly aligned. Figure 11 shows examples when major failures occur. In the first row, the farther left-facing facade is not detected due to lack of vanishing lines. In the second row, severe mistakes in semantic segmentation result in false facade planes being detected. These failures can be mitigated when vanishing line detection and semantic segmentation are improved.

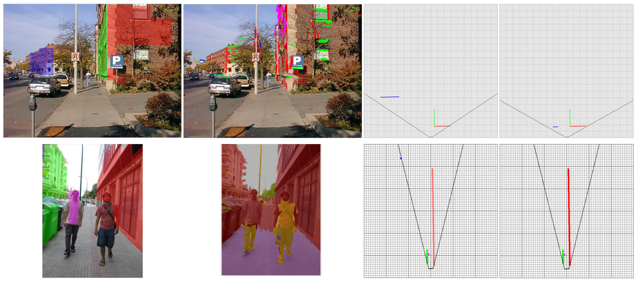

Figure 11. Major failure cases by our method during 3D facade layout estimation on the LabelMe dataset [13]. The conventions in this figure are the same as in Figure 10, except that the top and bottom images in column 2 are detected vanishing lines and semantic segmentation, respectively.

5.2. Surface layout estimation

To our knowledge, our algorithm is among the first to be able to generate a 3D layout of building facades consisting of mutually-constrained planes with continuous orientations. While our algorithm allows for a greater flexibility and produces a richer interpretation than existing methods, does it negatively affect existing quantitative measures in the literature?

To answer this question, we first evaluate our algorithm in estimating surface layout on the Geometric Context dataset [11]. Among the 250 test images, 55 of them contain building facades. Therefore, we perform evaluation only on those 55 images. As the dataset does not provide ground truth for semantic segmentation, we train the Stacked Hierarchical Labeling model [20] on the Stanford Background Dataset [5] and use the resulting model to compute soft semantic labels. We also use the publicly available pre-trained Geometric Context classifier [10] to obtain the soft surface layout labels.

The dataset comes with the ground truth of seven surface layout labels (“support”, “sky”, “planar left”, “planar center”, “planar right”, “non-planar porous”, “non-planar solid”). We compare our method with the block-based approach proposed by Gupta *et al.* [6] and the segment-based approach proposed by Hoiem *et al.* [12] on surface layout accuracy. To generate surface layout labels from the output of our algorithm, we quantize our continuous 3D orientation of facade planes into soft labels of “planar left”, “pla-

	Hoiem	Gupta	Ours	Ours w/o CRF	Orien. Map
Surface Layout Accuracy	72.87%	73.59%	74.82%	73.89%	71.20%

Figure 12. Comparison of surface layout accuracy. The methods from left to right are the algorithm from Hoiem *et al.* [12], Gupta *et al.* [6], our algorithm, our algorithm without CRF inference, and orientation map from the line-sweeping algorithm [18].

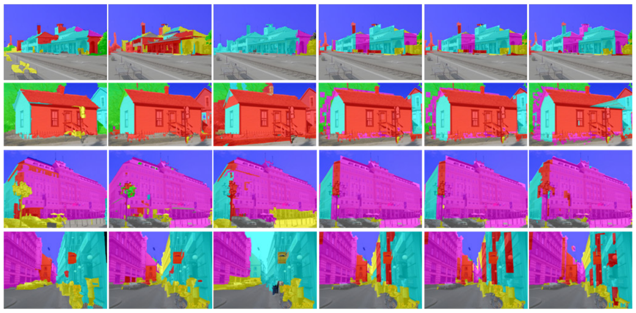

Figure 13. Qualitative comparisons of surface layout estimation. From left to right: Ground truth; Hoiem *et al.* [12]; Gupta *et al.* [6]; Ours; Ours w/o CRF; Orientation map from the line-sweeping algorithm [18]. Surface layout color code: Magenta – planar right; Cyan – planar left; red – planar center; green – non-planar porous; yellow – non-planar solid; blue – sky; grey – support.

nar center", or "planar right", and fuse them with the labels returned by the original Geometric Context classifier.

The comparison results are listed in Figure 12. We can see that our method achieves better accuracy than the state-of-the-art block-based and segment-based approaches. We also observe that if we remove the inter-planar geometric constraints from our algorithm, the accuracy drops as is shown in the 'Ours w/o CRF' column. Accuracy from the orientation map generated by the line-sweeping algorithm [18] is the lowest, as it is highly susceptible to noise in vanishing lines. While our algorithm takes in the orientation map as an important cue, the plane-level reasoning in our algorithm reduces much noise.

We also show qualitative comparisons in Figure 13. The ground-truth surface layout is shown in column 1. We can see that in the first row, the segment-based approach (column 2) decomposes the facade into many unstructured orientation segments, while the block-based approach (column 3) represents the entire facade as a block. Our plane-based approach (column 4) approximates the true facade using a set of mutually-constrained planes and therefore generates a much better approximation. In row 2, our approach identifies distinctive facades missed out by the other approaches. In row 3 where a non-Manhattan facade exists, our approach identifies it unambiguously. In row 4, an invalid plane labeled with "planar center"(red) in column 5 was removed after the CRF inference.

5.3. Depth map estimation

Another existing quantitative measure in the literature is recovering absolute depth values. To this end, we perform evaluation on the Make3D dataset of Saxena *et al.* [23, 19] which provides ground truth in pixel-wise depth value. We train the Stacked Hierarchical Labeling model [20] on the 400 training images of the dataset. As for the Geometric Context classifier, we still use the publicly available pre-trained one [10]. Among the 134 test images, 51 images contain building facades and are used in evaluation.

	Liu	Ours	Ours w/o CRF
Entire image	**0.144**	0.152	0.154
Exclude tree/foregnd.	0.130	**0.125**	0.127

Figure 14. Comparison of log depth error. The methods from left to right are the algorithm from Liu *et al.* [19], our algorithm, and our algorithm without CRF inference.

We compare our method with the super-pixel based approach proposed by Liu *et al.* [19] on average log error of pixel-wise depth values. For pixels located on trees and foreground objects, our method is not intended to infer their depths. However, for the sake of comparison, we roughly estimate the depths of those pixels from the ground contact point of the semantic region they reside in. Note that this is a very rough estimation compared with the sophisticated model in [19] specifically trained on 400 training images to predict the depths of such pixels.

To make a fair comparison, we report results on the average log depth error of pixels over the entire image as well as over the image regions *excluding* trees and foreground objects. The results are shown in Figure 14. We can see that when evaluating on the entire image, Liu *et al.* achieves the best performance. However, if we exclude the tree and foreground regions, our method outperforms Liu's super-pixel based approach. Note that our method does not require any training in terms of depth prediction. We can also see that, again, removing the inter-planar geometric constraints from our approach degrades performance.

While our approach achieves comparable performance with the state of the art in terms of existing quantitative measures, the output of our system is much richer than a surface layout map or a depth map.

6. Conclusion

We propose a plane-based 3D facade reasoning system that incorporates multiple image cues and geometric constraints into a probabilistic reasoning framework to achieve a coherent reconstruction of building facades in 3D space from a single 2D image of an urban scene. Our method yields a more informative interpretation of building facades while maintaining a competitive performance on several quantitative measures. Compared with the block-based approach [6] that yields a coarse and qualitative reconstruction, our method returns a numeric parameterization of a set of planes that compose facades. Compared with super-pixel based approaches that return a depth map [19], our method enables reasoning over planes and blocks and provides a higher-level understanding of the scene.

Acknowledgment: This work was supported in part by U.S. Army Research Laboratory (ARL) under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016, and by by ONR MURI Grant N000141010934.

References

[1] M. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. *CVPR*, 2000.

[2] B. Brillaut-O'Mahony. New method for vanishing point detection. *CVGIP: Image Understanding*, 54(2):289–300, September 1991.

[3] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. *ICCV*, 2013.

[4] R. Collins. Vanishing point calculation as statistical inference on the unit sphere. *ICCV*, pages 400–403, 1990.

[5] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. *ICCV*, 2009.

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

[7] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. *CVPR*, 2011.

[8] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. *CVPR*, 2012.

[9] T. Heskes. Stable fixed points of loopy belief propagation are minima of the bethe free energy. *Advances in NIPS*, 2003.

[10] D. Hoiem, A. A. Efros, and M. Hebert. Code for recovering surface layout from an image. http://www.cs.illinois.edu/homes/dhoiem/.

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

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

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

[14] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *IJCV*, 2011.

[15] K. Karsch, C. Liu, and S. B. Kang. Depthtransfer: Depth extraction from video using non-parametric sampling. *PAMI*, 2014.

[16] J. Kosecka and W. Zhang. Video compass. *ECCV*, 2002.

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

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

[19] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. *CVPR*, 2010.

[20] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. *ECCV*, 2010.

[21] S. Ramalingam and M. Brand. Lifting 3d manhattan lines from a single image. *ICCV*, 2013.

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

[23] A. Saxena, M. Sun, and A. Y. Ng. Make3d: learning 3-d scene structure from a single still image. *PAMI*, 2009.

[24] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: joint 3d layout and object reasoning from single images. *ICCV*, 2013.

[25] Y. Zhao and S. C. Zhu. Scene parsing by integrating function, geometry and appearance models. *ICCV*, 2013.