2013 IEEE Conference on Computer Vision and Pattern Recognition

# Manhattan Junction Catalogue for Spatial Reasoning of Indoor Scenes

Srikumar Ramalingam[1] Jaishanker K. Pillai[2*] Arpit Jain[2] Yuichi Taguchi[1]
[1]Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA
[2]Dept. of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA
{ramalingam,taguchi}@merl.com, {jsp,ajain}@umiacs.umd.edu

## Abstract

*Junctions are strong cues for understanding the geometry of a scene. In this paper, we consider the problem of detecting junctions and using them for recovering the spatial layout of an indoor scene. Junction detection has always been challenging due to missing and spurious lines. We work in a constrained Manhattan world setting where the junctions are formed by only line segments along the three principal orthogonal directions. Junctions can be classified into several categories based on the number and orientations of the incident line segments. We provide a simple and efficient voting scheme to detect and classify these junctions in real images. Indoor scenes are typically modeled as cuboids and we formulate the problem of the cuboid layout estimation as an inference problem in a conditional random field. Our formulation allows the incorporation of junction features and the training is done using structured prediction techniques. We outperform other single view geometry estimation methods on standard datasets.*

## 1. Introduction

Consider Figure 1 of a living room. Such man-made structures predominantly contain a large set of line segments. Two or more line segments intersect at different points and we refer to these intersections as junctions. Based on the patterns formed by the incident line segments, we can classify them into different categories such as **L**, **Y**, **W**, **T** and **X** junctions [28]. Indoor scenes are full of such junctions and a careful observer can even spot a **K** junction hidden in Figure 1. The types and locations of these junctions provide several geometrical cues about the scene. For example, the four corners of the room form **Y** junctions. **L** and **X** junctions generally occur on planar surfaces like the walls, ceiling and floor. **W** junctions are common on furniture boundaries and do not frequently appear on the walls and ceiling. In this paper, we detect these junctions automatically from a given image and use them to improve the spatial understanding of indoor scenes.

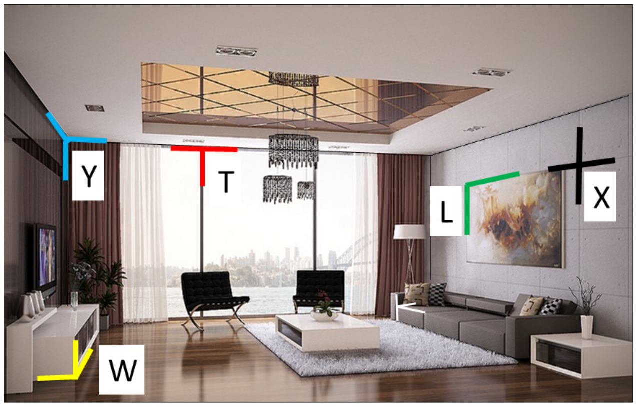


Figure 1. *A living room with several junctions of types* **L***,* **T***,* **Y***,* **X** *and* **W***. We present a novel method to detect these junctions and demonstrate that they are discriminative features in recovering the spatial layout of an indoor scene.*

### 1.1. Related Work

3D reconstruction from a single image is probably the oldest problem in computer vision [24, 15, 28]. As Sugihara [28] pointed out, human beings invented a noble class of pictures called *line drawings* to represent 3D shapes of objects. The problem of interpreting line drawings was considered as a means to reach the final goal of single view 3D reconstruction. Such approaches took it for granted that the line drawings are available through some means. While it is almost straightforward to detect junctions in a given line drawing, detecting junctions in real images is hard and ambiguous even for humans [19]. While the earlier approaches were purely geometrical with rigid constraints, the single view reconstruction problem was revisited recently with a newer set of ideas and algorithms. Rather than reconstructing a 3D scene using geometric primitives, Hoiem et al. [11, 12] represented a 3D scene as a popup model normally used to build stages for children's book. Using several image and geometrical features, their approach auto-

[*] now at A9, a subsidiary of Amazon.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.394

matically classifies the regions into ground, buildings and sky. In this paper, we use their terminology and refer to such coarse modeling as layout estimation. Saxena et al. [25] took a different approach to infer the absolute depth directly using both image features and weak assumptions based on coplanarity and connectivity. While it is relatively easier to extract depth from multiple images, this self-imposed single view constraint has driven vision researchers to look for every possible depth cue hidden in an image. We have begun revisiting some of the forgotten, yet powerful approaches in the early vision literature [24, 28]. Along this line of research, Gupta et al. [7] used physics-based constraints to model 3D scenes based on stability and mechanical properties. By categorizing scenes into 15 different scene geometries or stages with unique depth profiles, we can reduce the space of solutions in the 3D reconstruction problem [21].

For modeling indoor scenes, Hedau et al. [9] used a cuboid model to approximate the geometry of the room. Under this model, the pixels in a given image are classified into left wall, middle wall, right wall, floor and ceiling. In order to estimate the layouts, hundreds of cuboids are sampled and each one is given a score based on several image and geometric features. While orthogonal lines and vanishing points were useful in outdoor modeling, it is almost impossible to recover the indoor layouts without these features. This is because most indoor scenes satisfy Manhattan world assumptions. There has been several results in detecting rectangular structures [8, 20] in Manhattan worlds and even ordering them [34]. The success of these geometry estimation algorithms heavily depend on the success of several low-level image processing operations such as edge and line detection. Recently, Tretyak et al. [29] used a joint framework to robustly detect several useful geometric features such as line segments, groups of parallel lines, vanishing points and horizon. While clutter was initially considered as noise, recent approaches model them explicitly [22, 10, 32] as cuboids for better understanding of indoor scenes. Other useful single view cues include orientation maps [16] and human activities [5]. While many approaches do not explicitly use the boundary information, Del Pero et al. [22] emphasized the importance of using edge and corner features for accurately sampling the room layouts.

Without using novel features, we can still improve layout estimation by using better optimization algorithms. Schwing et al. [26] showed significant improvement in the performance of the layout estimation algorithm on standard benchmark using two existing features: geometric context [12, 9] and orientation maps [16]. This improvement was achieved by dense sampling of the image space and blowing up the solution space, followed by the use of a unified inference algorithm that combines the advantages of conditional random fields (CRFs) and structured support vector machines (SVMs) to find the optimum solution efficiently. It was also recently shown that such an inference can be performed exactly [27]. Natural statistics priors typically involve long range interactions and such cues can also be incorporated as higher order potentials in CRF-based layout estimation techniques [23, 6].

Our work is motivated by the use of interesting constraints on corners [17, 4]. In their work, corners are classified into concave, convex and occluding to build Manhattan worlds from single and multiple images. We refer to them as junctions and classify them into different categories based on the number and orientations of the line segments. This classification already captures some of the cues (concave and convex) explored in [17, 4] by using them in conjunction with the position of the vanishing points. While our work focuses on Manhattan junctions, other researchers have studied the detection of junctions [18] and boundaries [13] formed by contours in natural images. Example-based approaches have been used to reconstruct 3D scenes from line drawings [33, 2]. This problem addressed in [2] is highly combinatorial and the authors use novel shape patch representation and carefully design their pairwise compatibility terms to make the inference problem tractable. However, as we explain in Section 2.1, a more important challenge is to generate such line drawings from real images.

### 1.2. Contributions

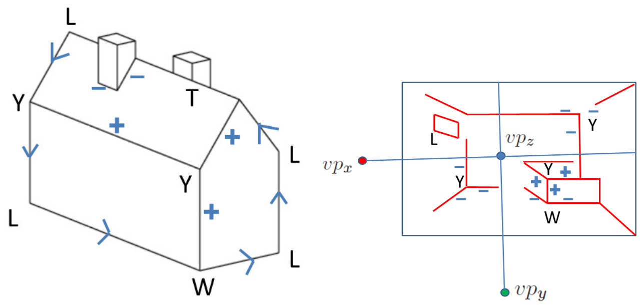
The main contributions of this paper are as follows:

- We exploit Manhattan junctions for spatial understanding of indoor scenes.
- We present an efficient voting-based method to detect the junctions.
- We show a CRF formulation to incorporate junction features for the layout estimation problem.
- We demonstrate state-of-the-art performance for the layout estimation problem on standard datasets.

## 2. Junction Detection

### 2.1. Background

Consider the line drawing shown in Figure 2(a). First, we need to detect the junctions and identify their types, such as **L**, **Y** and **X**. This problem is straightforward in the classical case where the line drawing is already given without any missing or spurious line segments. The second problem is the challenging one where the goal is to label the adjoining line segments of a junction as convex, concave and occluding. Each junction can take labels based on catalogues given by Huffman [14] and Clowes [1]. Several constraint satisfaction algorithms exist in the literature to label line drawings on a graph where the nodes are junctions and the adjacent line segments are edges. After labeling, inflation



(a) (b)

Figure 2. *(a) The classical line labeling problem. Given a line drawing with known junction types, we label the adjoining line segments as convex (+), concave (-) or occluding (←,→). (b) Under real world conditions, we encounter missing and spurious line segments. Our goal is to detect the junctions, classify their types and use them as priors in estimating the geometry of a scene.*

techniques are employed to reconstruct the shapes from the labeled line drawings [28, 31]. These techniques are generally successful and unambiguous on most line drawings, except on a few difficult and pathological cases.

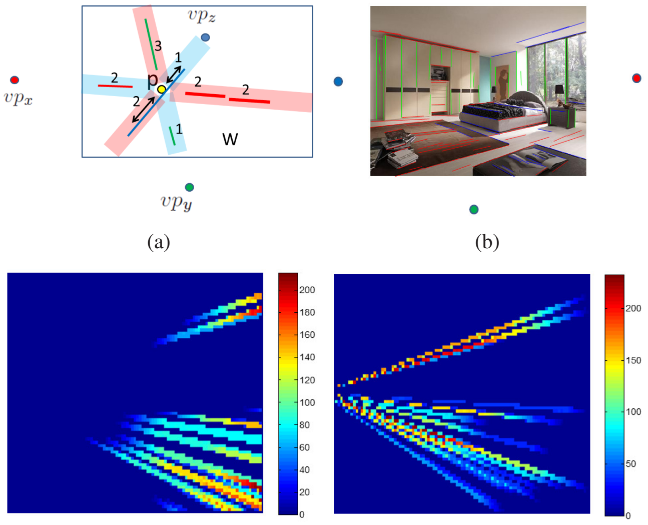
Figure 2(b) shows an illustration of a typical indoor scene with detected line segments. In this paper, we consider a class of junctions that differ from the classical ones as follows:

- We only consider *Manhattan junctions* formed by line segments in three principal orthogonal directions.
- In our problem, we also consider junctions that lie on a single plane. For example, the **L** junction shown in Figure 2(b) lies on a single plane and this can not be called an **L** junction in the classical sense.

Manhattan junctions provide many advantages over the classical ones. We refer to the three vanishing points as $vp_x$, $vp_y$ and $vp_z$. In the case of Manhattan junctions, we can infer whether a junction is convex or concave even without considering the neighboring junctions. For example, in Figure 2(b), the **Y** junction on the top right side of the $vp_z$ is concave. On the other hand, the **Y** junction on the bottom of the image is convex. In real world images, misclassification of junction types is always possible. In this paper, we do not explicitly solve the problem of labeling the adjoining line segments as convex, concave or occluding. However, our formulation implicitly uses this information for the layout estimation problem. While classical approaches use junctions as hard constraints in a constraint satisfaction algorithm, we use the detected junctions as soft priors in a probabilistic inference algorithm.

## 2.2. Voting-Based Detection Method

In a voting scheme, each data point votes for a specific parameter value in accumulators. The scheme is designed such that the peaks in the accumulators correspond to the instances of objects or patterns that are to be detected. In our work, we detect the junctions using a simple two-stage algorithm: (1) we vote for 6 accumulator arrays using line segments along the vanishing points, and (2) we detect different types of junctions by applying a product operation to the contents of the 6 accumulator arrays.



(a) (b)

(c) (d)

Figure 3. *The idea behind our voting method. We build 6 accumulator arrays, each of which stores votes from lines along one of the three principal orthogonal directions. (a) In this example, the contents of the accumulators corresponding to the point $p$ are given by $V_{\overrightarrow{x}}(p) = 2$, $V_{\overleftarrow{x}}(p) = 4$, $V_{\overrightarrow{y}}(p) = 1$, $V_{\overleftarrow{y}}(p) = 3$, $V_{\overrightarrow{z}}(p) = 1$ and $V_{\overleftarrow{z}}(p) = 2$. See the text for details on how the votes are computed. (b) An indoor scene along with the detected line segments and vanishing points. (c, d) The contents of the accumulators $\mathsf{V}_{\overrightarrow{z}}$ and $\mathsf{V}_{\overleftarrow{z}}$, respectively, for the image shown in (b).*

**(1) Voting:** We build 6 accumulators that store votes for a subset of pixels uniformly sampled from the original image. We refer to the 6 accumulators as $\mathsf{V}_{\overrightarrow{x}}$, $\mathsf{V}_{\overleftarrow{x}}$, $\mathsf{V}_{\overrightarrow{y}}$, $\mathsf{V}_{\overleftarrow{y}}$, $\mathsf{V}_{\overrightarrow{z}}$ and $\mathsf{V}_{\overleftarrow{z}}$. Let us denote the votes at a specific point $p$ in an accumulator $\mathsf{V}_j$ as $V_j(p)$, where $j \in \{\overrightarrow{x}, \overleftarrow{x}, \overrightarrow{y}, \overleftarrow{y}, \overrightarrow{z}, \overleftarrow{z}\}$. For each point $p$, every line segment that is collinear with the line joining $p$ and $vp_i$ ($i \in \{x, y, z\}$) votes for either $V_{\overrightarrow{i}}(p)$ or $V_{\overleftarrow{i}}(p)$ depending on its location with respect to $p$ and $vp_i$: If the line segment lies in the region between $p$ and $vp_i$, it votes for $V_{\overrightarrow{i}}(p)$; if the line segment lies outside of the region between $p$ and $vp_i$, and not adjacent to $vp_i$, then it votes for $V_{\overleftarrow{i}}(p)$. The vote is weighted by the length of the line segment. The subscript $\overrightarrow{i}$ refers to the line segments towards the vanishing point $vp_i$, while the subscript $\overleftarrow{i}$ refers to the line segments away from the vanishing point $vp_i$. This idea of voting is illustrated with an example in Figure 3(a). The accumulators can be filled efficiently with a complexity of $O(n)$, where $n$

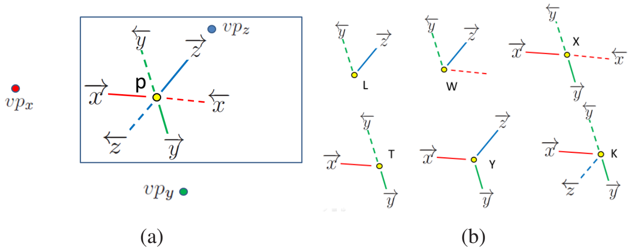


Figure 4. *(a) There are 6 possible directions at a point $p$ based on the three vanishing points. (b) By choosing different combinations of these directions, we can generate junctions of types* **L**, **T**, **Y**, **W**, **X** *and* **K**.

is the number of lines.

**(2) Detection:** Using the 6 accumulators, we can detect junctions using simple product operations. At every point $p$, the corresponding 6 accumulator cells $V_j(p)$ tell us the presence of lines that are incident with this point. As shown in Figure 4(a), the different junctions correspond to different subsets of these 6 elements. To detect a junction, we have to ensure that there are line segments in specific directions, and we also have to ensure that there are no line segments in the rest of the directions. In other words, for a point $p$, some accumulator cells should have non-zero values and the other should have zeros. This is the key insight in our approach that was critical in successfully detecting junctions. Sometimes what we do not see in an image gives more information than what we observe.

Let $S = \{\overrightarrow{x}, \overleftarrow{x}, \overrightarrow{y}, \overleftarrow{y}, \overrightarrow{z}, \overleftarrow{z}\}$. We denote every junction type as $\mathcal{J}_A$ where $A \subseteq S$. For a point $p$ and a junction type $\mathcal{J}_A$, we compute the following function:

$$f(p, A) = \prod_{i \in A} V_i(p) \prod_{j \in S \setminus A} \delta(V_j(p)) \qquad (1)$$

where $\delta(g)$ is a Dirac delta function that is 1 when $g = 0$ and 0 for all other values of $g$. If $f(p, A)$ is non-zero, then we detect a junction at point $p$ of type $\mathcal{J}_A$. Figure 4(b) shows examples of $A \subseteq S$ for different junctions of **L**, **Y**, **T**, **W**, **X** and **K**. There are $2^6$ possible junctions from different subsets of $S$ and some of them are not useful. For example, junctions $\mathcal{J}_{\{\overrightarrow{i}, \overleftarrow{i}\}}$, where $i \in \{x, y, z\}$, are just points lying on lines.

# 3. Inference

Our inference algorithm follows the approach of Hedau et al. [9] that uses structured SVM [30] for learning the scoring function that is used in evaluating several possible layouts and identifying the best one. We make two contributions in the inference algorithm that play a crucial role in the performance of our approach. First, we use a junction-based sampling for generating possible layouts. Second, we use an energy function based on a CRF model for scoring the layouts.

## 3.1. Junction-Based Sampling

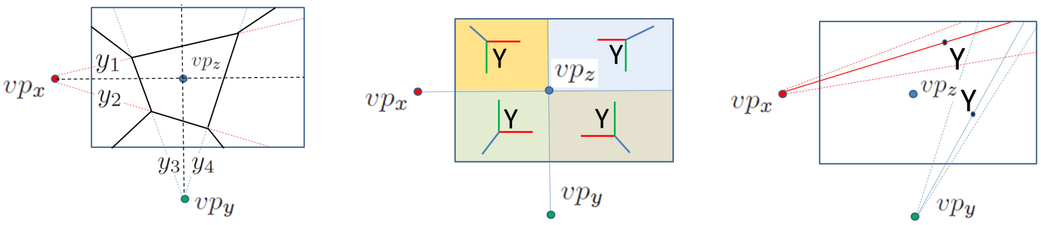
The idea of generating possible layouts for room geometry is shown in Figure 5(a). By sampling two horizontal rays ($y_1$ and $y_2$) passing through $vp_x$ and two vertical rays ($y_3$ and $y_4$) passing through $vp_z$ we can generate different layouts. Hedau et al. [9] used a coarse sampling with about 10 horizontal and 10 vertical rays. Recently, Schwing et al. [26] showed significant improvement in the layout estimation by more densely sampling the image space using 50 rays for each $y_i$ where $i \in \{1, 2, 3, 4\}$. Del Pero et al. [22] also showed that the generated layouts can be close to the true layout if we take into account the edge features and corners. In this work, we use a sampling that respects the detected junctions.

The image can be divided into four quadrants based on the positions of the three vanishing points as shown in Figure 5(a). Note that in some images, one or more of these quadrants may not lie inside the image boundaries. The four corners of the cuboid model must satisfy the property that each corner lies in only one of these quadrants. In addition, this corner must be a subtype of **Y** junctions as shown in Figure 5(b). In each of these quadrants, we detect the corresponding **Y** junctions. We start with a uniform sampling and between every pair of adjacent rays, we identify the **Y** junction with maximum score given by Equation (1). These high scoring junctions are used to get a new set of rays. As we show in the experiments, this data dependent sampling allowed us to identify layouts that are close to the true layouts while still using a coarse sampling.

## 3.2. Scoring Function Using a CRF Model

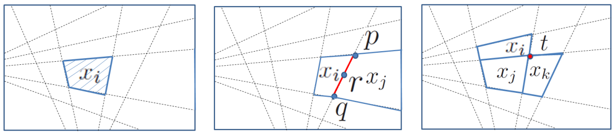
Given a set of training data $\{d_1, d_2, ..., d_n\} \in \mathbb{D}$ and the corresponding true layouts $\{l_1, l_2, ..., l_n\} \in \mathbb{L}$, our goal is to learn a mapping $g : \mathbb{D}, \mathbb{L} \to \mathbb{R}$ to identify the best layout. Here $l_i = \{y_1, y_2, y_3, y_4\}$, where $y_i$ corresponds to the rays used in generating the layout as shown in Figure 5(a). We would like to learn the function in such a manner that for the correct combination of the image $d_i$ and its true layout $l_i$, the function $g(d_i, l_i)$ is high. We also want to ensure that for any layout $l$ that is not the true layout, $g(d_i, l)$ should decrease as the deviation between the layouts $\Delta(l, l_i)$ increases. Usually the function $g$ will be of the form $g(d, l) = w^T \psi(d, l)$, where $w$ is the parameter vector that we would like to learn and $\psi(d, l)$ is a vector of features that can be computed from the image $d$ and the layout $l$. The mapping function can be learned discriminatively using a quadratic program formulated by structure learning techniques [30]. This has become a standard approach for learning structured outputs and it has been used by several layout estimation algorithms [9, 26].

We use the same methodology of [9], generating possible layouts and scoring them using a cost function. The difference in our work is that the cost function we use to evaluate



(a) (b) (c)

Figure 5. *Illustration of junction-based sampling of layouts. (a) By sampling two horizontal rays passing through $vp_x$ and two vertical rays passing through $vp_y$, we can generate a box layout where the five faces correspond to left wall, ceiling, middle wall, floor and right wall. (b) The image is divided into four quadrants based on the positions of the three vanishing points. Each corner of the room, if visible in the image, can only appear in one of the quadrants and it should belong to a specific subtype of* **Y** *junctions as shown. In each of these quadrants we store the detected* **Y** *junctions. (c) Given a regular sampling, we identify top scoring* **Y** *junctions in the cone spanned by two consecutive rays. These high scoring* **Y** *junctions are used to generate a new set of rays to sample layouts.*



(a) (b) (c)

Figure 6. *Our CRF model. (a) Given the rays used for sampling the layouts, we divide the image space into a set of polygons. Each polygon corresponds to a node $x_i$ in the CRF. Each node can take 5 labels $\{Left, Middle, Right, Floor, Ceiling\}$ that correspond to the five faces of the room. (b) Two nodes $x_i$ and $x_j$ are adjacent if they share a line segment $pq$. Our pairwise potentials in the CRF can be computed based on the presence of a line segment in the image that coincides with $pq$ or specific junctions detected at a point $r$ on the line segment $pq$. (c) Corners of a true layout usually coincides with specific* **Y** *junctions. If $t$ is detected as a* **Y** *junction, we would like to incorporate this prior in the form of a triple clique involving the incident nodes $x_i$, $x_j$ and $x_k$.*

a given layout is a CRF-based energy function.

We build our CRF graph using the rays that are used for generating layouts as shown in Figure 6. The rays partition the image space into several polygons. Our vertex set is given by $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, where $x_i$ is a node corresponding to a polygon in the image. The set of edges is given by $\{i, j\} \in \mathcal{E}$ if the nodes $x_i$ and $x_j$ share a line segment. Each possible layout can be parameterized by four rays $\mathbf{y} = \{y_1, y_2, y_3, y_4\}$, where $y_i \in \{1, 2, ..., k\}$ and $k$ corresponds to the number of sampled rays. In our CRF graph, every node $x_i = \{L, M, R, F, C\}$, which corresponds to the five faces given by left wall (L), middle wall (M), right wall (R), floor (F) and ceiling (C). Given $\mathbf{y}$, we can compute the corresponding $\mathbf{x}$ and vice versa. At the boundaries, a node $x_i$ in the CRF graph may correspond to two regions given by the rays $\mathbf{y}$. In such cases the node $x_i$ takes the label corresponding to the face that covers maximum area in the polygon. We could exactly handle the offset if we use few additional pairwise terms at the boundaries.

We use an energy function $E$ of the following form with unary, pairwise and triple clique terms:

$$E(\mathbf{x}, \omega) = \sum_{i=1}^{n} \omega_i \Psi(i, a)\delta_{ia} + \sum_{\{i,j\}\in\mathcal{E}} \omega_{i,j}\Psi(i, j, a, b)\delta_{ia}\delta_{jb} + \sum_{\{i,j,k\}\in\mathcal{T}} \omega_{i,j,k}\Psi(i, j, k, a, b, c)\delta_{ia}\delta_{jb}\delta_{kc}$$

$$i, j, k = \{1, 2, \dots, n\} \quad a, b, c = \{L, M, R, F, C\} \qquad (2)$$

Here, we denote unary potentials by $\Psi(i, a)$, pairwise potentials by $\Psi(i, j, a, b)$ and triple cliques by $\Psi(i, j, k, a, b, c)$. The unary potential corresponds to the cost added to the function when $x_i = a$. The pairwise term denotes the cost when $x_i = a$ and $x_j = b$. In the same manner, the triple clique denotes the cost when three variables take specific labels. $\mathcal{T}$ is a set of triplets $\{i, j, k\}$ representing incident nodes corresponding to high scoring **Y** junctions, as shown in Figure 6(c). The function $\delta_{ia}$ is a Kronecker delta function that is 1 when $x_i = a$ and 0 otherwise. The parameter we learn in this energy function is $\omega = \{\omega_i, \omega_{ij}, \omega_{ijk}\}$, which is computed from training data.

We construct the potentials using simple priors on junctions that are normally observed in indoor scenes as follows.

**Unary Potentials:** We use **L**, **T** and **X** junctions to build the unary terms. For every node $x_i$ we compute the cumulative sum of all the junctions of specific types that are detected inside the polygon. For every face, we take into account the orientation of the face. For example, the middle wall gets the junction scores from only junctions that span on the $Z$ plane. This can be easily computed using simple masks on accumulators that stores the junction scores.

**Pairwise Potentials:** Figure 6(b) shows two adjacent nodes $x_i$ and $x_j$ sharing a line segment $pq$. If there is a point $r$ on the line segment $pq$ corresponding to specific **Y** or **W** junctions, we would like to encourage the nodes $x_i$

and $x_j$ to take different labels. We use the junction scores of $\mathbf{Y}$ and $\mathbf{W}$ junctions on the line segment $pq$ as the pairwise potentials. We also use an additional pairwise potential that comes from the original line segments in the image. We would like to separate $x_i$ and $x_j$ into two faces if there is a line segment in the image that coincides with $pq$. This can be obtained directly from the difference of the accumulator cells we constructed for detecting junctions; in the example shown in Figure 6(b), the difference $V_{\overleftarrow{y}}(p) - V_{\overleftarrow{y}}(q)$ gives the length of the overlap of any line segment in the image with the line segment $pq$.

**Triple Clique Potentials:** Figure 6(c) shows a point $t$ that is incident with three nodes $x_i$, $x_j$ and $x_k$. If this point is detected as a $\mathbf{Y}$ junction that corresponds to a corner as shown in Figure 5(b), we would like to give higher score for a layout that is incident with this point. Thus, if a $\mathbf{Y}$ junction coincides with $t$, we use its junction score as a triple clique potential on the three incident nodes. It is easy to observe that by putting priors on three incident nodes, we can encourage the layout to pass through the incident point.

Once we have the unary, pairwise and triple clique potentials, our energy function in Equation (2) can also be seen as $E(\mathbf{x}, \omega)) = \omega^T \Phi(\Psi, \mathbf{x})$, where $\Phi(\Psi, \mathbf{x})$ is a linear vector that can be computed from $\mathbf{x}$ and $\Psi$. We can learn the weight parameter $\omega$ using structured SVM [30].

## 4. Experimental Results

**Junction Statistics:** In order to understand the distribution of different junctions on different faces of a room, we computed the statistics of junctions on 100 living room images from Flickr. The results shown in Table 1 match our intuition on junctions. Overall, the orientations of junctions agree well with the orientations of the room faces, despite the presence of furniture. **Y** junctions occur uniformly across all the faces as they predominantly appear on the layout boundaries. This also reaffirms our choice in using them as pairwise potentials. **W** junctions are common on the floor as they mostly occur on furniture. Qualitatively, our junction detection results are promising as shown in Figure 7, where the top scoring junctions are displayed. Quantitative results are difficult because there is no ground truth database for junctions. It would be a useful effort to generate such a ground truth for various junctions, although indoor scenes have hundreds of junctions and it may be hard to manually label them.

**Layout Estimation:** We tested our algorithm on Hedau et al.'s dataset [9] and the UCB dataset [34]. In [9], 209 images were used for training and 105 images were used for testing. We used the same split for training and testing. The UCB dataset includes 340 images, 228 of which were used for training and the rest for testing. Our CRF model can also incorporate other features such as geometric context (GC) and orientation maps (OM). This can be added as unary potentials in the same manner as we do for **L**, **X** and **T** junctions. To understand the advantages of different features, we learned the weight parameter $\omega$ using different combinations of features. For unary potentials, we compute 15 parameters (5 for junctions, 5 for OM and 5 for GC). Since every face can be adjacent to only two faces except the middle wall, there are only 8 possible pairwise interactions. We thus compute a total of 16 parameters for pairwise potentials, 8 for line-segment-based potentials and 8 for junction-based potentials. In every layout, we can only have a maximum of 4 corners. We therefore use 4 triple clique parameters corresponding to the 4 different **Y** junctions that can be seen at the corners of a room. Overall, we use 35 parameters when we consider all the features. We used structured SVM to compute the parameters and chose the regularization parameter $C$ using cross validation.

Table 2 reports our results using various combinations of features on Hedau et al.'s dataset in terms of the pixel misclassification error. We obtained an error of 18.85% using only junctions and 13.34% in combination with GC and

Table 1. *Percentages of junctions appearing in each of the 5 regions, averaged over 100 images. We show the statistics of junction type on each plane for* **L**, **T** *and* **X** *junctions. For example,* **L** *junctions on the $xy$ plane include* $\mathbf{L}_{\{\overrightarrow{x}\overrightarrow{y}\}}$, $\mathbf{L}_{\{\overrightarrow{x}\overleftarrow{y}\}}$, $\mathbf{L}_{\{\overleftarrow{x}\overrightarrow{y}\}}$ *and* $\mathbf{L}_{\{\overleftarrow{x}\overleftarrow{y}\}}$. *The results match our intuition that the junctions explaining a specific plane orientation appear more frequently on regions having the specific orientation.* **Y** *junctions occur uniformly on all faces, since they mostly appear on boundaries between regions.* **W** *junctions are predominant on the floor as they are common along furniture boundaries.*

| Type | Plane | Left | Middle | Right | Floor | Ceiling |
|---|---|---|---|---|---|---|
| **L** | $xy$ | 11.1 | 49.7 | 9.8 | 20.4 | 9.0 |
| | $yz$ | 34.2 | 8.7 | 30.0 | 19.8 | 7.3 |
| | $zx$ | 11.8 | 13.7 | 16.3 | 46.0 | 12.2 |
| **T** | $xy$ | 8.0 | 53.0 | 8.2 | 15.8 | 15.0 |
| | $yz$ | 41.4 | 11.8 | 36.2 | 10.0 | 0.6 |
| | $zx$ | 10.4 | 9.5 | 15.4 | 54.0 | 10.7 |
| **X** | $xy$ | 5.6 | 64.8 | 8.0 | 16.5 | 5.1 |
| | $yz$ | 53.7 | 0.0 | 28.7 | 17.6 | 0.0 |
| | $zx$ | 4.2 | 5.3 | 10.5 | 74.7 | 5.3 |
| **Y** | – | 21.0 | 19.4 | 20.0 | 19.5 | 20.1 |
| **W** | – | 15.0 | 10.5 | 23.9 | 46.0 | 4.6 |

Table 2. *Pixel misclassification error on Hedau et al.'s dataset [9]. Individually, GC gives an error of 19.5% and OM gives an error of 20.2% [16]. As expected, the combination of the three features produces an error of 13.34%, which is slightly better than the 13.59%, which is the current state-of-the-art without furniture reasoning [27].*

| Features | Error [%] |
|---|---|
| Unary (Junction) + Pairwise + Triple | 18.85 |
| Unary (Junction) + GC + OM | 14.61 |
| Unary (Junction) + GC + Pairwise + Triple | 13.70 |
| Unary (Junction) + GC + OM + Pairwise + Triple | 13.34 |

OM. These numbers indicate that junctions are discriminative and are capable of giving good results on layout estimation even without using any other features. Some of the best and worst results are depicted in Figure 8. Note that some of the best results are very close to the true layout despite a coarse sampling of layouts. We used the same sampling rate as used in [9] (10 rays for horizontal and 10 rays for vertical). In all the images, we scored less than 1000 possible layouts to identify the best layout. Our adaptive sampling is powerful and we also detect locally occluded junctions if there are line segments farther that support the junction. On the UCB dataset, we obtained an error of 16.11% by junctions in combination with GC and OM, but without using any furniture reasoning. The current best result on this dataset is an error of 18.84% reported in [22].

**Processing Time:** Our C++ implementation takes less than 1 second for detecting lines, computing vanishing points and obtaining junctions. GC and OM features are computed using the code provided by Hedau et al. [9] and Lee et al. [16]. The CRF is implemented in Matlab and it takes about 5 seconds to build the graph and potentials. The inference takes less than 10 milliseconds, since we score not more than 1000 layouts to pick the best one.

## 5. Conclusion

In this paper, we have shown an efficient method to detect Manhattan junctions and used them for spatial understanding of indoor scenes. As shown in [32], HOG descriptor [3] can also model junctions, but they do not make Manhattan assumptions and represents only local evidence. We use global evidence based on long line segments that may even be occluded near the junctions. In the future, we plan to use junctions for detecting furniture and explore them in the framework of [27] that can handle denser sampling.

**Acknowledgments:** We thank Jay Thornton, Shotaro Miwa, Makito Seki, Jonathan Yedidia, Matthew Brand, Philip H.S. Torr, and Peter Varley for useful discussions on single view 3D reconstruction and line drawings. We also thank the AC and anonymous reviewers for insightful feedback and future directions. This work was supported by and done at MERL. J. K. Pillai and A. Jain contributed to the work while they were interns at MERL.

## References

[1] M. B. Clowes. On seeing things. *AI*, 1971.

[2] F. Cole, P. Isola, W. T. Freeman, F. Durand, and E. H. Adelson. Shapecollage: Occlusion-aware, example-based shape interpretation. In *ECCV*, 2012.

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

[4] A. Flint, D. Murray, and I. Reid. Manhatten scene understanding using monocular, stereo, and 3D features. In *ICCV*, 2011.

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

[6] A. Gallagher, D. Batra, and D. Parikh. Inference for order reduction in markov random fields. In *CVPR*, 2011.

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

[8] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing with attribute grammar. *PAMI*, 2009.

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

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

[11] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 2005.

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

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

[14] D. A. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 1971.

[15] T. Kanade. A theory of origami world. *AI*, 1980.

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

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

[18] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[19] J. McDermott. Psychophysics with junctions in real images. *Perception*, 2004.

[20] B. Micusik, H. Wildenauer, and J. Kosecka. Detection and matching of rectilinear structures. In *CVPR*, 2008.

[21] V. Nedovic, A. W. M. Smeulders, A. Redert, and J.-M. Geusebroek. Stages as models of scene geometry. *PAMI*, 2010.

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

[23] S. Ramalingam, P. Kohli, K. Alahari, and P. Torr. Exact inference in multi-label CRFs with higher order cliques. In *CVPR*, 2008.

[24] L. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, MIT, 1963.

[25] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D depth reconstruction from a single still image. *IJCV*, 2008.

[26] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3D indoor scene understanding. In *CVPR*, 2012.

[27] A. G. Schwing and R. Urtasun. Efficient exact inference for 3D indoor scene understanding. In *ECCV*, 2012.

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

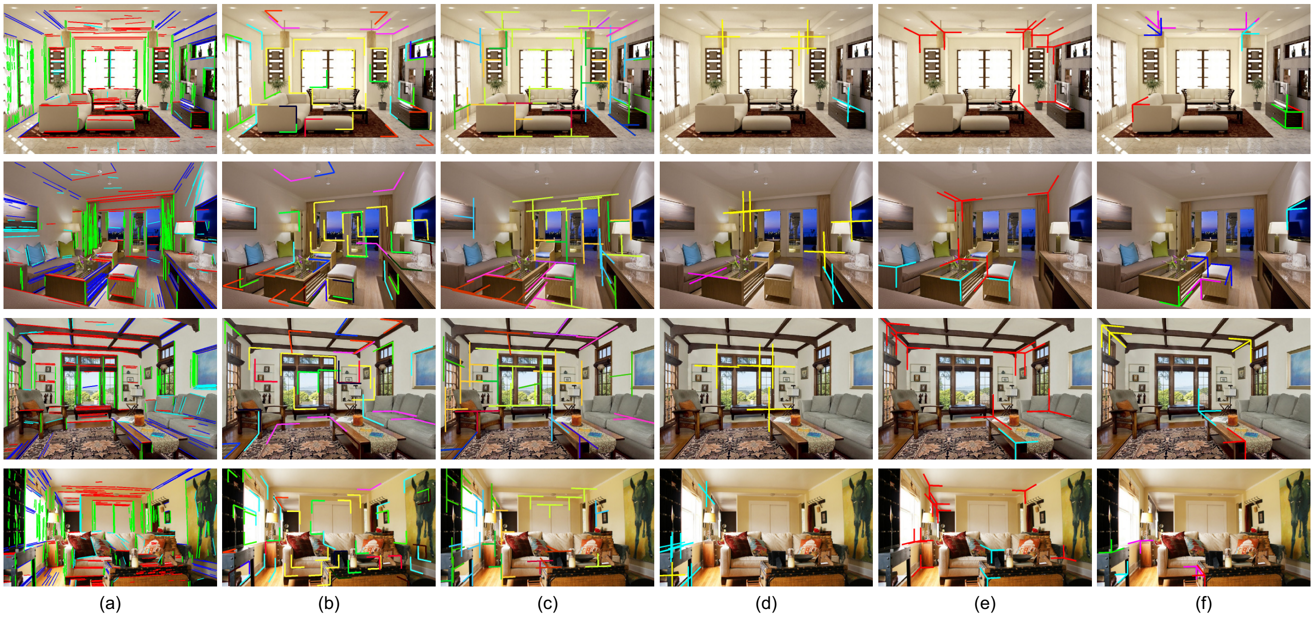


Figure 7. *(a) Input images with detected line segments. (b–f) Detected junctions for each type of* **L**, **T**, **X**, **Y** *and* **W**, *respectively. For visualization purpose, only a subset of junctions that have the highest score in their small neighborhoods for each type are shown.*

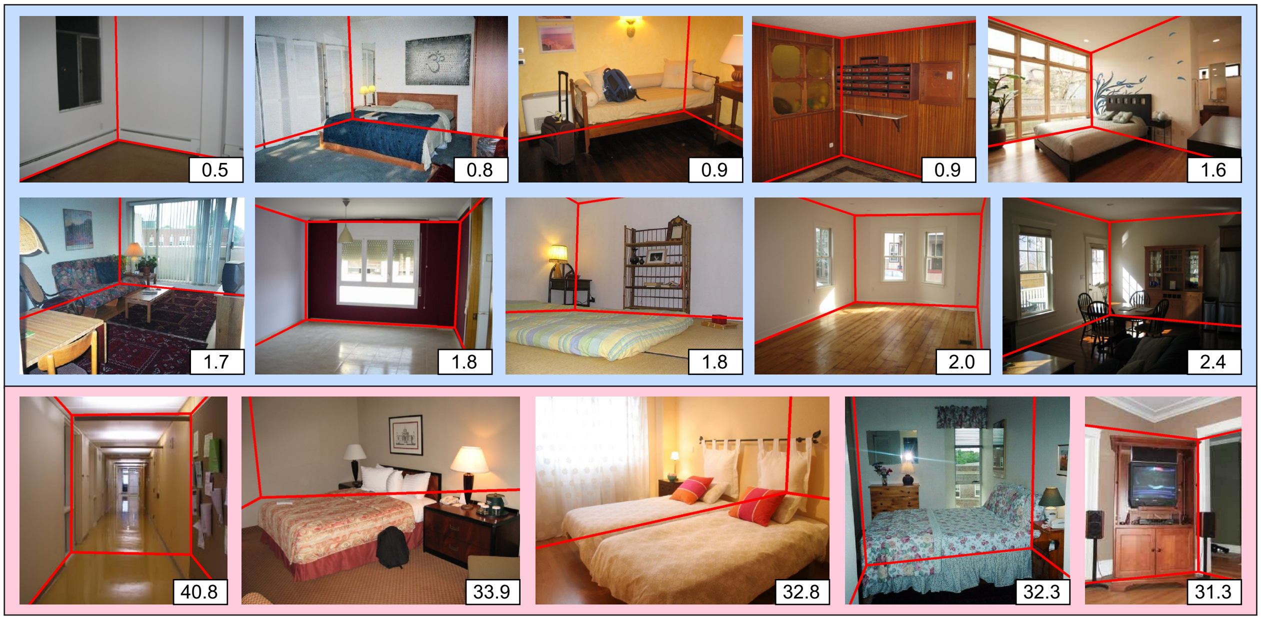


Figure 8. *Layout estimation results on Hedau et al.'s dataset [9]. The 10 best (top) and 5 worst (bottom) results are shown with the pixel misclassification error percentages. Note the accuracy of the layouts despite using a course sampling. We obtain the high boundary accuracy because we detect not only visible junctions, but also occluded ones if there are line segments supporting them.*

[29] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *IJCV*, 2012.

[30] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005.

[31] P. Varley. *Automatic Creation of Boundary-Representation Models from Single Line Drawings*. PhD thesis, Cardiff University, 2003.

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

[33] T. Xue, Y. Li, J. Liu, and X. Tang. Example-based 3D object reconstruction from line drawings. In *CVPR*, 2012.

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