Estimating the 3D Layout of Indoor Scenes and its Clutter from Depth Sensors

Jian Zhang
Tsingua University
jizhang@ethz.ch

Chen Kan
Tsingua University
chenkan0007@gmail.com

Alexander G. Schwing
ETH Zurich
aschwing@inf.ethz.ch

Raquel Urtasun
TTI Chicago
rurtasun@ttic.edu

Abstract

In this paper we propose an approach to jointly estimate the layout of rooms as well as the clutter present in the scene using RGB-D data. Towards this goal, we propose an effective model that is able to exploit both depth and appearance features, which are complementary. Furthermore, our approach is efficient as we exploit the inherent decomposition of additive potentials. We demonstrate the effectiveness of our approach on the challenging NYU v2 dataset and show that employing depth reduces the layout error by 6% and the clutter estimation by 13%.

1. Introduction

Finding the 3D structures composing the world is key for developing autonomous systems that can navigate the environment, and importantly, recognize and interact with it. While finding such structures from monocular imagery is extremely difficult, depth sensors can be employed to reduce the inherent ambiguities of still images. In the outdoor setting, high-end depth sensors such as the Velodyne laser scanner are a must for autonomous navigation. Notable examples are the Google car as well as the participants of the DARPA Urban Challenge, which rely heavily on these sensors as well as prior knowledge in the form of detailed annotated maps.

In the past few years, a wide variety of approaches have exploited cheap depth sensors (*e.g*., Microsoft Kinect) to improve the accuracy and robustness of computer vision tasks. A notable example is the kinect pose estimation system [24], probably one of the most successful commercial products to come out of the computer vision community. Additionally, the superiority of RGB-D sensors when compared to more traditional imagery has been demonstrated for the tasks of semantic segmentation [25, 26, 8], inferring support relations [26], 3D detection [14] or estimating physical properties of images [2].

Semantic parsing approaches that utilize RGB-D imagery try to estimate the basic components of a room (*e.g*., walls, floor, furniture). While effective, they formulate the problem as a semantic segmentation task, failing to exploit the structure of the problem: rooms mostly satisfy the Manhattan world assumption, and the walls, floor and ceiling are typically aligned with three dominant orientations which are orthonormal. While these assumptions are widely used in the monocular setting in order to estimate the layout of rooms [29, 10, 11, 22, 23, 18], to our knowledge, they are not commonly exploited in the presence of RGB-D imagery.

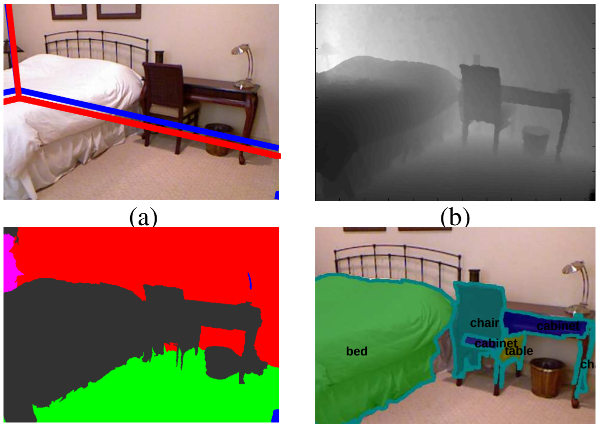
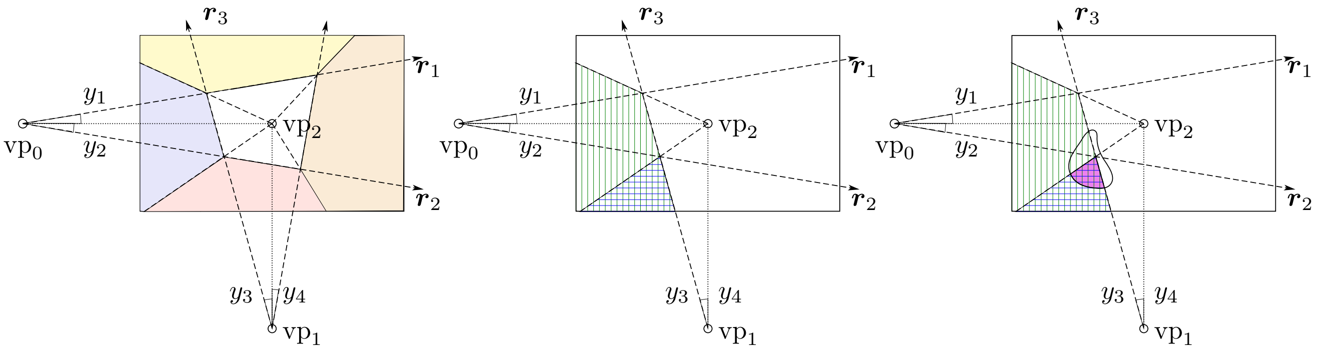

(a)
(b)
(c)
(d)

Figure 1. **Inferring layout, clutter and semantic classes:** The input image with layout prediction result (blue) and ground truth (red) overlayed is shown in (a), the depth map employed is shown in (b). (c) and (d) show our inferred labeling and segmentation.

In addition to estimation of the room layout, we should be able to retrieve the objects that compose the scene in order to develop autonomous systems. In this paper we propose an approach to semantic parsing that estimates both the layout of rooms as well as the clutter present in the scene. We make use of both appearance and depth features, which, as we show in our experimental evaluation are complementary, and frame a joint optimization problem which exploits the dependencies between these two tasks. We derive an effective iterative scheme and show how decomposition methods (*i.e*., integral geometry) can be employed to decompose our additive energies into Markov random fields (MRFs) with potentials containing at most two random variables. This results in an efficient algorithm, which performs very well in practice. Furthermore, we employ appearance and depth features in order to estimate a set of furniture



(a) Parameterization of the layout (b) Integral geometry for the left wall (c) Integral geometry and compatibility potential

Figure 2. **Layout parameterization and integral geometry:** While the parameterization of the layout task is illustrated in (a), (b) depicts that the area of the left wall is decomposable into a difference of green and blue highlighted ranges both depending on two angles only. (c) Usage of integral geometry for computation of compatibility potentials where we count how much a superpixel overlaps with a hypothesized wall.

classes from the super-pixels labeled by our approach as clutter. The output of our method is illustrated in Fig. 1, where RGB-D imagery is utilized in order to estimate the room layout, the clutter as well as semantic segmentation.

We demonstrate the effectiveness of our approach using the challenging NYU v2 dataset [26] and show that by employing depth we boost performance of the layout estimation task by 6% while clutter estimation improves by 13%. This is to be expected since "clutter" occupies the foreground which is easily apparent in depth images. Additionaly we show how a wide variety of semantic classes can be obtained by utilizing depth and appearance features.

2. Related Work

Early approaches to semantic scene understanding in the outdoor setting focused on producing qualitative 3D parses [19, 13, 6], ground plane estimates [30] or parsing facades [27, 17]. More recently, accurate estimations of the road topologies at intersections [4] as well as the 3D vehicles present in the scene [5] have been estimated from stereo and monocular video respectively. Depth sensors in the form of high-end laser scanners have become a standard in the context of autonomous driving (*e.g.*, the Google car).

Indoor scene understanding approaches have taken advantage of the Manhattan world properties of rooms and frame the layout estimation task as the prediction of a 3D cuboid aligned with the three main dominant orientations [10, 11, 18, 22, 23, 16, 29]. Assuming vanishing points to be given, Hedau *et al*. [10] and Wang *et al*. [29] showed that the problem has only four degrees of freedom. Inference, however, remains difficult as a priori the involved potentials, counting features in each of the faces defined by the layout, are high-order. As a consequence, only a few candidates were utilized, resulting in suboptimal solutions. A few years later, Schwing *et al*. [22] showed that the a priori high-order potentials, are decomposable into sums of pairwise potentials by extending the concept of integral images to accumulators oriented with the dominant orientations. As a consequence denser parameterizations were possible, resulting in much better performance. In [23], a branch and bound approach was developed to retrieve a global optimum of the layout problem. More general layouts than 3D cuboids were predicted in [3]. Among other applications, room layouts have been used in [7] to predict affordances and in [12, 20] to estimate the free space.

While a wide variety of approaches have been proposed in the monocular setting, to our knowledge no approach has taken advantage of RGB-D imagery. Perhaps one of the reasons is the absence of a dataset with depth and layout labels. In this paper we investigate how cheap depth sensors can be used to help the layout problem and show that significant improvements are obtained. Towards this goal, we labeled a subset of the NYU-RBGD v2 dataset with layout labels.

In the monocular setting, Wang *et al*. [29] reason jointly about the layout as well as the clutter present in the scene. They propose to make use of an iterated conditional modes (ICM) algorithm, to tractably deal with the complex potentials resulting from the interaction of the clutter and the layout. However, this algorithm gets easily trapped in local optima. As a result, their layout estimation results are more than 5% lower than the state-of-the-art. In contrast, in this paper we propose an effective approach to the joint layout and clutter estimation problem, which is able to exploit appearance, depth, as well as compatibility potentials linking the two estimation problems. We propose an effective inference scheme, which alternates between computing the layout and solving for the image labeling problem. As we take advantage of the inherent decomposition of the potentials, our approach is efficient and results in impressive performance improving 6% over the state-of-the-art in the layout task and 13% in estimating clutter.

3. Joint Layout and Clutter Labeling

In this section we describe our novel approach, which jointly estimates the layout of rooms as well as the clutter present in the scene. Towards this goal, we propose a

1	a can't be up above b
2	a can't be below to b
3	a can't be right to b
4	a can't be left to b
5	a can't be in front of b
6	a can't be behind b

a – b	ceiling	floor	left wall	front wall	right wall	clutter
ceiling		2	2, 4	2, 5	2, 3	2
floor	1		1, 4	1, 5	1, 3	1
left wall	1, 3	2, 3		3, 5	3	3
front wall	1, 6	2, 6	4, 6		3, 6	6
right wall	1, 4	2, 4	4	4, 5		4
clutter	1	2	4	5	3	

Table 1. **3D Physical Constrains**: (left) set of physical constraints (right) encoding in terms of pairwise potentials.

holistic approach that is able to exploit appearance as well as depth features. In the remainder of the section, we first show how to obtain better superpixels by exploiting depth. We then introduce our joint model which operates on superpixels and random variables representing the layout, and discuss our learning and inference procedures.

3.1. Superpixel estimation

We are interested in partitioning the image into superpixels such that each one represents a planar surface which is part of a single object. Following [32], we extend the SLIC [1] algorithm to utilize both appearance and depth information. We formulate the segmentation problem as minimizing the sum of three energies, encoding shape, appearance and depth. In particular, E_{loc} is a location energy encoding the fact that superpixels should have regular shape, E_{depth} encourages the depth of the superpixels to be piecewise planar and E_{app} encodes the fact that we would like to have similar appearance for all pixels that are subsumed by a superpixel. More formally, let $s_p \in \{1, \cdots, K\}$ be a random variable encoding the assignment of pixel $\mathbf{p}$ to a superpixel. We define the energy of a pixel $\mathbf{p}$ to be

$$\begin{aligned} E(\mathbf{p}, s_p, \mu_{s_p}, c_{s_p}, D_{s_p}) &= E_{loc}(\mathbf{p}, s_p) + \lambda_a E_{app}(\mathbf{p}, s_p, \mathbf{c}_{s_p}) \\ &\quad + \lambda_d E_{depth}(\mathbf{p}, s_p, g_{s_p}), \end{aligned}$$

with two scalars λ_a and λ_d encoding the importance of the appearance and depth terms. We encode the appearance of each superpixel in terms of a mean descriptor in Lab space, and set

$$\begin{aligned} E_{loc}(\mathbf{p}, s_p) &= \|\mathbf{p} - \mu_{s_p}\|_2^2, \\ E_{app}(\mathbf{p}, s_p, \mathbf{c}_{s_p}) &= \|I(\mathbf{p}) - \mathbf{c}_{s_p}\|_2^2, \\ E_{depth}(\mathbf{p}, s_p, g_{s_p}(\mathbf{p})) &= \|\nabla d(\mathbf{p}) - g_{s_p}\|_2^2, \end{aligned}$$

with μ_{s_p} the mean position of superpixel s_p, $\mathbf{c}_{s_p}$ the mean descriptor in Lab space, $I(\mathbf{p})$ the Lab image at pixel $\mathbf{p}$, $g_{s_p}(\mathbf{p})$ the mean depth gradient descriptor, and $\nabla d(\mathbf{p})$ the depth gradient computed from the RGB-D imagery.

We thus define the superpixel labeling problem as the following minimization

$$\min_{\mathbf{s},\mathbf{D},\mu,\mathbf{c}} \sum_{\mathbf{p}=1}^{N} E(\mathbf{p}, s_p, \mu_{s_p}, \mathbf{c}_{s_p}, g_{s_p}),$$

with $\mathbf{s} = \{s_1, \cdots, s_N\}$, $\mu = \{\mu_1, \cdots, \mu_K\}$, $\mathbf{c} = \{\mathbf{c}_1, \cdots, \mathbf{c}_K\}$ and $\mathbf{g} = \{g_1, \cdots, g_K\}$ the set of all superpixel assignments, mean positions, mean appearance and depth descriptors. We solve this optimization problem by alternating between solving for assignments $\mathbf{s}$ given all other variables, and solving for $\mathbf{g}, \mu, \mathbf{c}$ given fixed assignments. Note that this is done very efficiently: The first step decomposes over pixels, while the latter decomposes over superpixels and even admits an update in closed form [1].

3.2. Joint model

Now that we have partitioned the image into superpixels by incorporating the depth cue, we define our energy considering both layout and labeling variables. Our joint model is a conditional random field (CRF) over both the labeling and layout task. For each superpixel, the labeling task assigns one of the six labels, *i.e.*, $x_i \in \{clutter, left, right, front, ceiling, floor\} = \mathcal{L}$. Following recent monocular approaches [29, 22], we represent the layout task in terms of rays originating from two different vanishing points (VPs). In particular, we utilize the vanishing points of [26], which take advantage of the depth channel to produce better estimates. Given these VPs, we represent the layout problem with four parameters $\mathbf{y} = \{y_1, y_2, y_3, y_4\} \in \mathcal{Y}$. We refer the reader to Fig. 2(a) for an illustration of this parameterization.

We define the energy of the system to be the sum of three energies representing the layout and labeling tasks as well as a compatibility term which encodes the relationship between the two tasks. We thus have

$$E(\mathbf{x}, \mathbf{y}) = E_{layout}(\mathbf{y}) + E_{labeling}(\mathbf{x}) + E_{comp}(\mathbf{x}, \mathbf{y}),$$

where we did not explicitly provide the dependency on the RGB-D imagery for notational convenience. Note that E_{comp} couples the inference problems, making the estimation computationally difficult. In the following we describe each of these terms.

Layout Energy: Following monocular approaches [15, 22], we define the energy of the layout as a sum of energies over the faces α of the cuboid. For each of the five faces, we obtain the energy as a weighted sum of counts

$$E_{layout}(\mathbf{x}, \mathbf{y}) = \sum_{\alpha=1}^{5} w_{lay,\alpha}^{\top} \phi_{lay,\alpha}(\mathbf{y}). \quad (1)$$

We employ Orientation maps (OM) [16] and Geometric Context (GC) [13, 10] as image evidence. Given edges detected in the image, OMs estimate a normal orientation for

Algorithm 1 MAP Inference

```
$\mathbf{y}^{(0)} = \min_{\mathbf{y}} E_{layout}(\mathbf{y})$
for all $i = 1 : M$ do
    $\mathbf{x}^{(i)} = \min_{\mathbf{x}} E_{labeling}(\mathbf{x}) + E_{comp}(\mathbf{x}, \mathbf{y}^{(i-1)})$
    $\mathbf{y}^{(i)} = \min_{\mathbf{y}} E_{layout}(\mathbf{y}) + E_{comp}(\mathbf{x}^{(i)}, \mathbf{y})$
end for
Return $\mathbf{x}^{(M)}, \mathbf{y}^{(M)}$
```

each pixel. Using the VP configuration, we convert these normals into wall estimates, resulting in a five-dimensional feature for each pixel. GCs are six-dimensional features that utilize classifiers to predict the probability of each wall as well as cutter. As a consequence, $\phi_{lay,\alpha}(\mathbf{y}) \in \Re^{11}$. Note that although a priori these potentials are high-order (*i.e.*, up to order 4 as specifying the front wall requires four variables), we utilize integral geometry to decompose the potentials into sums of terms of order at most two [22]. This is illustrated in Fig. 2(b) for the left wall.

Labeling Energy: We define the labeling energy to be composed of unary and pairwise potentials as follows

$$E_{labeling}(\mathbf{x}) = \sum_{i=1}^{K} w_{lab}^{\top} \phi_{lab}(x_i) + \sum_{i,j \in \mathcal{N}(i)} w_{pair}^{\top} \phi_{pair}(x_i, x_j),$$

with $\mathcal{N}(i)$ the set of neighbors adjacent to the i-th superpixel. The unary term employs OM, GCs and depth-based features. In the case of GCs we learn a different weight for each entry, thus resulting in 36 features. We utilize two depth features: The first one encodes the idea that the superpixel normals which do not belong to the clutter class should be in accordance with the main dominant directions. To capture this, we generate a six dimensional feature, where each entry is the cosine of the angle between the surface normal of the superpixel and each dominant orientation. The second feature, encodes our intuition that the superpixels labeled as bounding surfaces (*i.e.*, left, right, front, ceiling, floor) should be close to the boundary of the scene. We thus define a six dimensional feature, which computes the distance between the mean 3D position of the superpixel being referred to as its centroid and the centroid of the superpixel furthest away in each dominant direction. As pairwise features, we encode the 3D physical relations that exist between the different labels, *e.g.*, we know that the floor cannot be above the left wall. Table 1 summarizes the relationships we employ. Note that we learn a different weight for every entry in the table which results in $36 + 12 = 48$ unary features and 36 pairwise features.

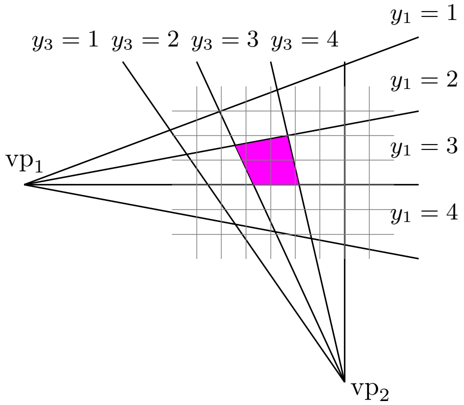


Figure 3. Dynamic programming computation of accumulators.

Compatibility Energy: This energy encodes the fact that the layout and the labeling problems should agree. In particular, we define

$$E_{comp}(\mathbf{x}, \mathbf{y}) = \sum_{\alpha=1}^{5} \sum_{\gamma=1}^{6} w_{comp,\alpha,\gamma} \phi_{comp,\alpha,\gamma}(\mathbf{x}, \mathbf{y}),$$

where we compute compatibility for all classes

$$\phi_{comp,\alpha,\gamma}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{p} \in \beta(\mathbf{y},\alpha)} \delta(x_i(\mathbf{p}) = \gamma),$$

with $\beta(\mathbf{y}, \alpha)$, a function that returns the set of pixels $\mathbf{p}$ which compose the wall α defined by the layout $\mathbf{y}$. For a specific choice of α and γ the feature measures the area that is predicted to be α by the layout task while being assigned γ by the labeling prediction. This enforces both tasks to produce consistent labels. Note that we can extend the concept of integral geometry introduced by [22] to this case. In a naïve way we could iterate over all superpixels, set their corresponding area to one while leaving everything else to be zero, thus performing integral geometry computations for each superpixel independently. Since we know which pixels belong to a specific superpixel, we derive a method that allows computation of those compatibility potentials with a single linear pass over the image, *i.e.*, with complexity quadratic in the dimension of the image. The details are provided in Sec. 3.5. Fig. 2(c) provides an illustration of both the potentials as well as integral geometry.

3.3. Inference

During inference we are interested in computing the maximum a-posteriori (MAP) estimate, or equivalently the minimum energy configuration which can be computed by solving the following optimization problem

$$\min_{\mathbf{x},\mathbf{y}} E_{layout}(\mathbf{y}) + E_{labeling}(\mathbf{x}) + E_{comp}(\mathbf{x}, \mathbf{y}).$$

Due to the high-order terms imposed by the compatibility potential, we propose to solve this in an iterative fashion, alternating between solving for the labeling $\mathbf{x}$, and the layout $\mathbf{y}$. As a consequence, the potentials are of order at most 2 for each minimization problem. Alg. 1 summarizes the

Algorithm 2 Accumulator computation

```
1: for all $\mathbf{p}$ do
2:   Initialize $y_1, y_3$ from pixel above and to the left
3:   while $\angle(\mathbf{p} - \mathrm{vp}_1, r_1(y_1)) < 0$ do
4:     $y_1 \leftarrow y_1 + 1$
5:   end while
6:   while $\angle(\mathbf{p} - \mathrm{vp}_2, r_3(y_3)) < 0$ do
7:     $y_3 \leftarrow y_3 + 1$
8:   end while
9:   $A(y_1, y_3) \leftarrow A(y_1, y_3) + 1$
10: end for
```

inference process. We use convex belief propagation [9] to solve each one of the minimization tasks. In particular, we employ the parallel implementation of [21], which exploits multiple cores and machines.

3.4. Parameter Learning

Let $\mathbf{w}$ be the vector concatenating all the weights, and let ϕ be the vector that concatenates all the potentials. Denote $\{(I_i, \hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)\}$ to be a set of N training examples fully annotated with a labeling $\mathbf{x}$ and a layout $\mathbf{y}$ for an RGB-D image I. We employ structured SVM [28] to learn the parameters of the model by minimizing

$$\begin{aligned} &\min_{\mathbf{w},\varepsilon} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{i=1}^{N}\varepsilon_i \\ &s.t. \ \forall i, \ \forall(\mathbf{x}_i, \mathbf{y}_i) \in (\mathcal{X}_i, \mathcal{Y}_i) \backslash (\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) \\ &\mathbf{w}^\top(\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)) \geq \Delta(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) - \varepsilon_i. \end{aligned} \tag{2}$$

For clarity of notation we did not provide the dependency on the image I by means other than the index of the example. Let $\mathcal{X}_i = \mathcal{L}^K$ be the labeling product space of the K superpixels. We define the loss as the sum of the losses over each individual task as follows

$$\Delta(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) = \Delta_{labeling}(\mathbf{x}_i, \hat{\mathbf{x}}_i) + \Delta_{layout}(\mathbf{y}_i, \hat{\mathbf{y}}_i).$$

For both losses we employ the pixel-wise loss as it decomposes into unary potentials for the labeling task and pairwise potentials for the layout task (via integral geometry). We employ the cutting plane algorithm of [28] to solve this optimization problem. As for computing the MAP, we utilize dcBP [21] within an alternate minimization scheme to perform loss-augmented inference.

3.5. Speeding up computations

One of the most expensive tasks in inferring the layout is computation of the accumulators required to perform integral geometry. Note that due to the compatibility potentials, this has to be computed at each iteration in Alg. 1. In this section we develop an algorithm that reduces the complexity by more than one order of magnitude with respect to [22], by utilizing a dynamic programming scheme.

	layout	labeling
GC [13]	–	26.38%
Schwing et al. [23]	13.66%	–
Ours rgb	13.94%	23.68%
Ours rgb depth	**8.04%**	**13.65%**

Table 2. Comparison to the state-of-the-art with different features.

When computing the accumulator A, we accumulate the value of every pixel into a 2D matrix indexed by two angles. This is illustrated for $A(y_1, y_3)$ in Fig. 3, where the pixel grid is given by the gray lines while the black rays illustrate a coarse discretization of the state space with the accumulator matrix entry $A(3, 3)$ being highlighted in magenta color. In [22], the accumulator entry for every pixel is found independently. This is suboptimal, as unnecessary computations are performed. Instead, here we utilize information from the pixel's neighbors to the left and to the top while assuming that an image is parsed from top left to bottom right.

Assume we know that the neighboring pixel to the top was sorted into a row with $y_1 = a$. It is easy to see that the current pixel lies in a row with $y_1 \geq a$. Furthermore, it is typically either the same row or the neighboring one. Analogously, assume that the neighboring pixel to the left was sorted into a column with $y_3 = b$. It is again easy to see that the current pixel lies in a column with $y_3 \geq b$, most likely very close to the b-th column.

We propose to exploit this locality to reduce computation. Alg. 2 summarizes our approach. To find the exact accumulator cell we proceed by initializing y_1 and y_3 with the row and column of the neighbor to the top and to the left respectively as stated in line 2. In the next step we find the first angle y_1 having a ray $r_1(y_1)$ that has to be rotated in counter-clockwise direction onto the ray connecting the current pixel $\mathbf{p}$ with the VP vp_1, *i.e.*, we increase y_1 by one as long as the angle $\angle(\mathbf{p} - \mathrm{vp}_1, r_1(y_1))$ between the two rays is negative and hence the vector denoted by $r_1(y_1)$ has to be rotated in clockwise direction onto the vector $\mathbf{p} - \mathrm{vp}_1$. Proceeding similarly for y_3 provides the respective cell which is – depending on the level of discretization – frequently found after only one iteration within the loop. A similar strategy is utilized for each accumulator. The computational complexity of computing all accumulators is $O(N^2)$, while the complexity in [22] is $O(N^2|\mathcal{Y}|^2)$. As show in our experiments this results in significant savings of computation.

4. Experimental Evaluation

We took a randomly chosen subset of the NYU-RGBD-v2 dataset [26] and manually labeled it to have ground truth layouts, as only pixel-wise labelings in terms of semantic categories are provided. We split the data into 202 images for training and 101 images for testing.

Comparison to state-of-the-art: We first compare our approach to the state-of-the-art, which only employs monocular imagery. We choose [22] as it has been recently shown to be the best performing approach in the benchmarks that exist for this problem (*i.e.*, the layout and bedroom datasets of [10, 11]). As shown in Table 2, in the layout task we outperform [22] by more than 5%. For the labeling task, the results are even better since usage of a depth cue improves GC by 13%.

Importance of depth features: As shown in Table 2, by employing depth, our approach improves accuracy by 10% in labeling and by 6% in layout estimation. Depth features improve the labeling error on many of the images quite significantly. This is visualized in Fig. 4(a) where every circle denotes a test sample. For samples above the red line, the approach utilizing only RGB features has an error larger than the method employing the depth cue.

Unsupervised segmentation: We now compare the importance of using depth in our unsupervised segmentation algorithm. For all images, we utilize 200 superpixels per image. This is a good compromise between inference speed and accuracy. We evaluate the segmentation in terms of three metrics. The first one is the mean error (ME), which is define as the percentage of incorrectly labeled pixels if we had an oracle labeling each super-pixel with the semantic labels. We use as oracle the semantic labels provided with the dataset. The second measure represents the segmentation covering in percentage. This metric computes the overlap measure multiplied by the area covered by the region, over all labels present in the region. Thus, it is defined as

$$\mathcal{C}(S' \to S) = \frac{1}{N}\sum_{l_i} |R_{l_i}| \,.\, \max_{l'_i} \mathcal{O}(R_{l_i}, R'_{l'_i}),$$

with S' a segmentation, S the ground truth, and R_{l_i} the region in S covered by label l_i. The last metric is the variation of information, which is a measure of the difference of information between the ground truth and our segmentation. It is defined as

$$VI(S, S') = H(S) + H(S') - 2I(S, S'),$$

with $H(S)$ and $H(S')$ measuring respective information (*i.e.*, entropy) $H(S) = -\sum_{l_i} \mathrm{p}(l_i) \log \mathrm{p}(l_i)$ where $\mathrm{p}(l_i)$ defined as the empirical distribution. Further, $I(S, S')$ represents the mutual information shared between S and S' and is defined as

$$I(S, S') = \sum_{l_i}\sum_{l'_i} \mathrm{p}(l_i, l'_i) \log \frac{\mathrm{p}(l_i, l'_i)}{\mathrm{p}(l_i)\mathrm{p}(l'_i)}.$$

I (λ_a)	D (λ_d)	ME	Seg. Cover	Var of Inf
1	0	2.51%	88.76%	0.63
5	1	2.28%	89.17%	0.60
2	1	2.19%	89.32%	0.59
1	1	**2.15%**	**89.36%**	**0.58**
1	2	2.16%	89.35%	**0.58**
1	5	2.26%	89.15%	0.59
0	1	2.81%	88.18%	0.66

Table 3. **Super-pixel Estimation:** Unsupervised segmentation results as a function of the importance of the appearance and depth terms. A good compromise is equal weighting for both appearance and depth. The original SLIC corresponds to $\lambda_d = 0$. Its performance is clearly inferior to using both sources of information.

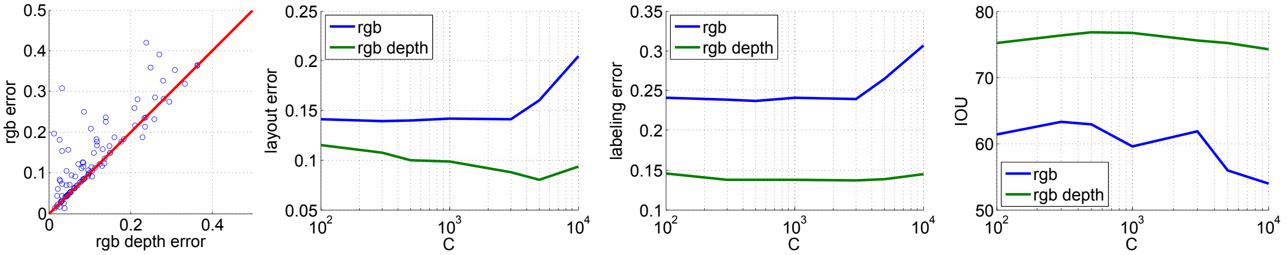
Table 3 depicts segmentation performance for different parameters. A good compromise is attained when equal weighting is used for both appearance and depth. Note that the original SLIC algorithm corresponds to $\lambda_d = 0$. Its performance is clearly inferior.

Fast accumulator computation: We now compare our proposed computation of compatibility accumulators to the implementation proposed in [22]. While the naïve implementation needs about 7 seconds to process an image, our dynamic programming improvement retrieves an identical result in 0.32 seconds, *i.e.*, we observe a significant improvement of more than one order of magnitude.

Robustness to the parameters: We now show the robustness of our approach to the main parameter, which is C, the strength of regularization in Eq. 2. As shown in Fig. 4, our approach is fairly robust and good results can be obtained for a wide range of the parameter C. This is illustrated in Fig. 4(b) – (d) for the pixel-wise layout estimation error, the superpixel labeling error and the average intersection over union measure for the different walls as well as clutter. The intersection over union measure is detailed for $C = 1000$ in Tab. 4 for the different walls and clutter class as well as the resulting average.

Semantic labels: We employ [31] to compute pixel-wise classifiers in terms of 6 furniture classes. In particular, we used six RGB-D kernel descriptors *gradient*, *color*, *local binary pattern*, *depth gradient*, *spin/surface normal*, and *KPCA/self-similarity*. Table 5 depict the performance of our semantic segmentation. Note that the class 'table' is particularly difficult to detect as it is typically surrounded by clutter.

Qualitative results: We provide some qualitative results in Fig. 5. In the first three columns we show OM, GC and depth. In the fourth and fifth column we illustrate the ground truth labeling as well as our labeling estimation. The sixth column depicts the layout estimates in (blue) and the



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

Figure 4. Scatter plot of the error when using RGB or RGB-D data in (a). Robustness w.r.t. C for layout, labeling and intersection over union in (b) – (d).

IOU	ceiling	floor	left	front	right	clutter	average
rgb (1000)	53.57	51.48	61.71	69.21	63.37	58.26	59.60
rgb depth (1000)	80.36	85.84	68.93	77.27	75.15	72.47	76.67

Table 4. Intersection over union (IOU) computed as in the PASCAL segmentation challenge. The labeling task consist on 6 classes, the five walls and clutter. Note that by using depth the average IOU measure improves by more than 17%, a very significant result.

	100	300	500	1000	3000	5000	10000
toilet	35.67	35.72	37.19	36.03	39.54	38.30	32.02
bed	43.69	43.59	43.42	43.46	43.00	43.47	42.49
table	16.90	20.90	21.29	21.21	20.60	20.98	21.02
cabinet	20.68	20.61	21.49	21.26	21.43	19.23	19.60
sofa	32.50	32.28	32.90	32.88	32.86	32.64	32.11
chair	35.89	35.90	36.18	36.13	35.96	35.00	35.05

Table 5. IOU for the semantic classes as a function of C

ground truth in red. Finally the last two columns provide the semantic labeling for the ground truth as well as our estimates. Our failure cases are due to wrong vanishing points, as, *e.g*., illustrate in the last row.

5. Conclusion

We have proposed an approach to jointly estimate the layout of rooms as well as the clutter present in the scene using RGB-D data. Towards this goal, we derived and efficient algorithm to perform inference within a joint model and demonstrate its effectiveness on the NYU v2 data set, showing impressive error reductions over the state-of-the-art of 6% for the layout task and 13% in estimating cluttered. We also demonstrated that clutter can be further employed to segment several furniture classes. We plan to further extend our approach to be able to exploit video as well as to incorporate objects in the form of 3D cuboids.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. In *PAMI*, 2012. 3

[2] J. T. Barron and J. Malik. Intrinsic Scene Properties from a Single RGB-D Image. In *Proc. CVPR*, 2013. 1

[3] A. Flint, D. Murray, and I. Reid. Manhatten Scene Understanding Using Monocular, Stereo, and 3D Features. In *Proc. ICCV*, 2011. 2

[4] A. Geiger, M. Lauer, and R. Urtasun. A Generative Model for 3D Urban Scene Understanding from Movable Platforms. In *Proc. CVPR*, 2011. 2

[5] A. Geiger, C. Wojek, and R. Urtasun. Joint 3D Estimation of Objects and Scene Layout. In *Proc. NIPS*, 2011. 2

[6] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Proc. ECCV*, 2010. 2

[7] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3D Scene Geometry to Human Workspace. In *Proc. CVPR*, 2011. 2

[8] S. Gupta, P. Arbelaez, and J. Malik. Perceptual Organization and Recognition of Indoor Scenes from RGBD Images. In *Proc. CVPR*, 2013. 1

[9] T. Hazan and A. Shashua. Norm-Product Belief Propagation: Primal-Dual Message-Passing for LP-Relaxation and Approximate-Inference. *Trans. on Information Theory*, 2010. 5

[10] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the Spatial Layout of Cluttered Rooms. In *Proc. ICCV*, 2009. 1, 2, 3, 6

[11] V. Hedau, D. Hoiem, and D. Forsyth. Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. In *Proc. ECCV*, 2010. 1, 2, 6

[12] V. Hedau, D. Hoiem, and D. Forsyth. Recovering Free Space of Indoor Scenes from a Single Image. In *Proc. CVPR*, 2012. 2

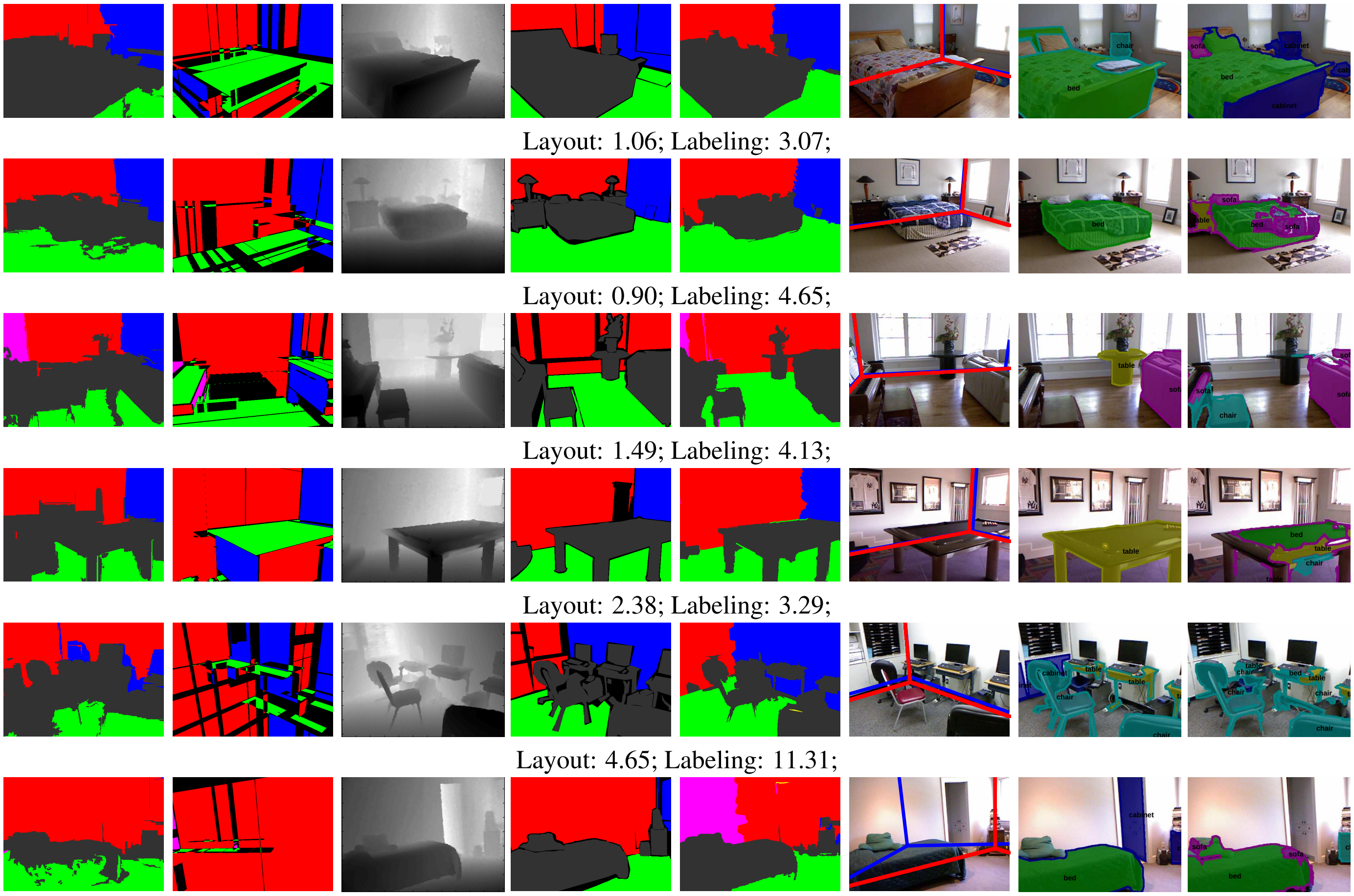
[13] D. Hoiem, A. A. Efros, and M. Hebert. Automatic Photo Pop-up. In *Siggraph*, 2005. 2, 3, 5

[14] H. Jiang and J. Xiao. A Linear Approach to Matching Cuboids in RGBD Images. In *Proc. CVPR*, 2013. 1

[15] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces. In *Proc. NIPS*, 2010. 3

[16] D. C. Lee, M. Hebert, and T. Kanade. Geometric Reasoning for Single Image Structure Recovery. In *Proc. CVPR*, 2009. 2, 3

[17] A. Martinovic, M. Mathias, J. Weissenberg, and L. van Gool. A Three-Layered Approach to Facade Parsing. In *Proc. ECCV*, 2012. 2



Layout: 1.06; Labeling: 3.07;

Layout: 0.90; Labeling: 4.65;

Layout: 1.49; Labeling: 4.13;

Layout: 2.38; Labeling: 3.29;

Layout: 4.65; Labeling: 11.31;

Layout: 44.62; Labeling: 27.39;

Figure 5. **Qualitative results**: In the first three columns we show OM, GC and depth. In the fourth and fifth column we show the ground truth labeling as well as our labeling estimation. The sixth column depicted the layout estimates in (blue) and the ground truth in red. Finally the last two columns depicted the semantic labeling for the ground truth as well as our estimates. A failure mode due to bad vanishing points is illustrated in the last row.

[18] L. Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *Proc. CVPR*, 2012. 1, 2

[19] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. In *PAMI*, 2008. 2

[20] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box In the Box: Joint 3D Layout and Object Reasoning from Single Images. In *Proc. ICCV*, 2013. 2

[21] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Distributed Message Passing for Large Scale Graphical Models. In *Proc. CVPR*, 2011. 5

[22] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction for 3D Indoor Scene Understanding. In *Proc. CVPR*, 2012. 1, 2, 3, 4, 5, 6

[23] A. G. Schwing and R. Urtasun. Efficient Exact Inference for 3D Indoor Scene Understanding. In *ECCV*, 2012. 1, 2, 5

[24] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient Human Pose Estimation from Single Depth Images. In *PAMI*, 2012. 1

[25] N. Silberman and R. Fergus. Indoor Scene Segmentation using a Structured Light Sensor. In *Workshop on 3D Representation and Recognition*, 2011. 1

[26] N. Silberman, P. Kohli, D. Hoiem, and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *Proc. ECCV*, 2012. 1, 2, 3, 5

[27] O. Teboul, I. Kokinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape Grammar Parsing via Reinforcement Learning. In *Proc. CVPR*, 2011. 2

[28] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support Vector Learning for Interdependent and Structured Output Spaces. In *Proc. ICML*, 2004. 5

[29] H. Wang, S. Gould, and D. Koller. Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding. In *Proc. ECCV*, 2010. 1, 2, 3

[30] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes. In *Proc. ECCV*, 2010. 2

[31] L. B. Xiaofeng Ren and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012. 6

[32] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust Monocular Epipolar Flow Estimatio. In *Proc. CVPR*, 2013. 3