Geometry Driven Semantic Labeling of Indoor Scenes

Salman Hameed Khan[1], Mohammed Bennamoun[1], Ferdous Sohel[1], and Roberto Togneri[2]

[1] School of CSSE, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia
[2] School of EECE, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia

Abstract. We present a discriminative graphical model which integrates geometrical information from RGBD images in its unary, pairwise and higher order components. We propose an improved geometry estimation scheme which is robust to erroneous sensor inputs. At the unary level, we combine appearance based beliefs defined on pixels and planes using a hybrid decision fusion scheme. Our proposed location potential gives an improved representation of the planar classes. At the pairwise level, we learn a balanced combination of various boundaries to consider the spatial discontinuity. Finally, we treat planar regions as higher order cliques and use graphcuts to make efficient inference. In our model based formulation, we use structured learning to fine tune the model parameters. We test our approach on two RGBD datasets and demonstrate significant improvements over the state-of-the-art scene labeling techniques.

1 Introduction

The task of indoor scene labeling is a relatively difficult problem compared to its outdoor counterpart. Indoor scenes have a large number of categories that are significantly different from each other (e.g., corridors, bookstores and kitchens). They also contain illumination variations, clutter, significant appearance variations and imbalanced representation of object categories [27]. Recently, inexpensive structured light sensors (e.g., Microsoft Kinect) are proving to be a rich source of information for indoor scenes. They provide co-registered color (RGB) and depth (D) images in real-time. Efficient use of this information for indoor scene labeling problems is a critical opportunity.

Several recent works focus on the use of RGBD images for scene labeling of indoor scenes. Koppula *et al.* [20] used Kinect fusion to create a 3D point cloud and then densely labeled it using a Markov Random Field (MRF) model. Silberman and Fergus [34] achieved a reasonable semantic labeling performance using a Conditional Random Field (CRF) with SIFT features and 3D location priors. Couprie *et al.* [3] used ConvNets to learn feature representations from RGBD data to label the images while Ren *et.al* [31] employed kernel descriptors to

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 679–694, 2014.

capture the distinctive features. These works are focused on extracting discriminative features from RGBD data and have shown that the depth information can certainly improve the scene labeling performance. However, the question of how to adequately incorporate depth information to model local, pairwise and higher order interactions has not been fully addressed.

In this work, we propose a novel depth-based geometrical CRF model to more efficiently utilize the depth information along side the RGB data. *First*, we incorporate the geometrical information in the most important potential of our CRF model, namely the appearance potential. At the appearance level, we encode both the intensity and depth based characteristics in the feature space. These features are used to predict the unary potentials in a discriminative fashion. Likewise, planes, which are the fundamental geometric units of indoor scenes, are extracted using a new smoothness constraint based *region growing algorithm* (see Sec. 5). Compared to other plane detection methods (e.g.,[29, 35]), our method is robust to outer-boundary holes present in Kinect's depth maps. The geometric as well as the appearance based characteristics of these planar patches are learned and used to provide unary estimates. We propose a novel *hierarchical fusion scheme* to combine the pixel and planar based unary potentials. This hierarchical scheme first uses a number of contrasting opinion pools and finally combines them using a Bayesian framework (see Sec. 3.1).

Next, we turn our attention towards the *location potential*, which encodes the possible spatial locations of all classes. In contrast to the conventional 2D location prior (e.g., in [33, 34]), we propose to integrate the rough geometry of planar regions along with their location in each scene (see Sec. 3.1, 4.1). We also propose a novel *spatial discontinuity potential* (SDP) in the pairwise smoothness model. It combines a number of different boundaries (such as depth edges, contrast based edges and super-pixel edges) and learns a balanced combination of these using a quadratic cost function minimization procedure based on the manually segmented images of the training set (see Sec. 4.2). *Finally*, we add a higher order potential (HOP) in our CRF model which is defined on cliques that encompass planar patches. The proposed HOP increases the expressivity of the random field model by assimilating the geometric context. This encourages all pixels inside a planar patch to take the same class label (see Sec. 3.3).

In short, we have proposed a new random field formulation which elegantly combines the geometric information with the appearance information at various levels of the model hierarchy (Fig. 1).

2 Related Work

The use of depth sensors for scene analysis and understanding is increasing. Recent works employ depth information for various purposes e.g., object detection [8], semantic segmentation [11, 20], object grasping [30], door-opening [28] and object placement [14] tasks. For the case of semantic labeling, works such as [3, 31, 34, 35] demonstrate that depth information reasonably helps in achieving better performance. They however do not explore possible ways, other than the depth based features, to incorporate depth information. In this paper, we define

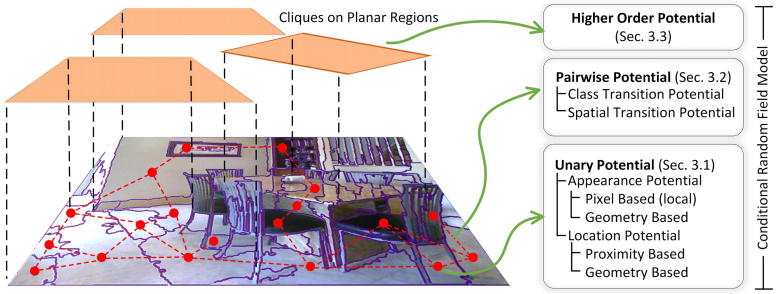

Fig. 1. Our approach combines geometrical information with low-level cues with in a CRF model. Only limited graph nodes are shown for the purpose of clear illustration.

various levels where depth information can be incorporated in a random field model and then explore how each level contributes to enhance the performance of semantic labeling. Our framework is particularly inspired by the works on semantic labeling of RGBD data [34, 35], considering long range interactions [19], parametric learning [36, 37] and geometric reconstruction [29].

The **scene parsing** problem has been studied extensively in recent years. Graphical models e.g., MRF and CRF have found success in modeling context and providing a consistent labeling [9, 12, 13, 23, 26]. Hierarchical MRFs are employed in [21] to make inference jointly on pixels and super-pixels. Huang *et al.* [13] trained the CRF on separate clusters of similar scenes and used them with standard CRF to label street images. Several research works (such as [3, 34, 41]) have shown that the depth based information enhances segmentation performance. They however remain limited to the use of depth based features and do not exploit the geometry of the regions and high level interactions.

An important challenge in scene labeling is to incorporate **long-range interactions** between graph nodes while making local decisions. Farabet *et al.* extracted dense features at a number of scales at each pixel location [5]. Other works incorporate wide context by generating a number of varying scale segmentations (often arranged as trees) to propose many possible labelings (e.g., [2, 21]). HOPs have been employed to model long range smoothness [19], shape based information [24], cardinality based potential [39] and label co-occurrences [22]. In contrast to previously proposed HOPs [18, 19], we propose to consider the geometrical structure of the scenes to model high level interactions.

Currently popular **parameter estimation** methods include partition function approximations [33], cross validation [33] or simply hand picked parameters [34]. We used a one-slack formulation [15] of the parameter learning technique of [36], which gives a more efficient optimization compared to [36, 37]. Further, we extend the parameter estimation problem to consider various different boundary potentials in the SDP and learn them using a tractable quadratic program.

Our **geometric reconstruction** scheme is close to those proposed in [29, 41]. Both these schemes use data from accurate laser scanners and can not handle the less accurate depth data acquired by a real time operating Kinect sensor. Our

proposed algorithm relaxes the smoothness constraint in the erroneous depth map regions and considers more reliable cues to segment the planar patches.

3 Proposed Conditional Random Field Model

The CRF model considers the appearance, location, boundaries and layout of pixels to reason about a set of semantically meaningful classes. We want the model to capture not only the neighboring interactions in a standard grid graph structure, but to also consider the long range interactions defined on planar regions (Fig. 1). The CRF model is defined on a graph $\mathcal{G}(\mathcal{I}) = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ composed of a set of vertices $\mathcal{V}$, edges $\mathcal{E}$ and cliques $\mathcal{C}$. The goal of multi-class image labeling is to segment an image $\mathcal{I}$ by labeling each pixel p_i with its correct class label $\ell_i \in \mathcal{L} = \{1..L\}$. The conditional distribution of output classes ($\mathbf{y}$) given an input image ($\mathbf{x}$) and parameters ($\mathbf{w}$) can be defined as a function of Gibbs energy: $\mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) = \frac{1}{Z(\mathbf{w})}\exp(-\mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w}))$. This energy is defined in terms of negative log-likelihoods as:

$$\mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w}) = \sum_{i\in\mathcal{V}} \psi_i(y_i,\mathbf{x};\mathbf{w}_u) + \sum_{(i,j)\in\mathcal{E}} \psi_{ij}(y_{ij},\mathbf{x};\mathbf{w}_p) + \sum_{c\in\mathcal{C}} \psi_c(y_c,\mathbf{x};\mathbf{w}_c). \quad (1)$$

The three terms in Eq. 1 are the unary, pairwise and higher order energies respectively. The parameters introduced in Eq. 1 are learnt using a max-margin criterion, details of which are given in Sec. 4.2. At the inference stage, the most likely labeling is found by making a MAP estimate $\mathbf{y}^*$ upon a set of random variables $\mathbf{y} \in \mathcal{L}^N$: $\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{L}^N}{\operatorname{argmax}}\, \mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w})$.

3.1 Unary Potentials

The unary potential in Eq. 1 is further divided into two components, appearance potential and location potential (Fig. 1):

$$\sum_{i\in\mathcal{V}} \psi_i(y_i,\mathbf{x};\mathbf{w}_u) = \sum_{i\in\mathcal{V}} \overbrace{\phi_i(y_i,\mathbf{x};\mathbf{w}_u^{app})}^{\text{appearance}} + \sum_{i\in\mathcal{V}} \overbrace{\phi_i(y_i,i;\mathbf{w}_u^{loc})}^{\text{location}} \quad (2)$$

We treat both terms separately in the following sections.

Appearance Potential: The proposed appearance potential in Eq. 2 is defined over both pixels and planar regions (Fig. 1). We used a hierarchical ensemble learning method to combine local appearance and geometric information (Fig. 2). We use the class predictions defined over planar regions to help in improving the posterior defined over pixels. In other words, planar features are used to aid in reinforcing beliefs on some dominant planar classes (e.g., walls, blinds, floor and ceiling). At the first level, m contrasting opinions ($\kappa_j : j \in [1,m]$) are used to combine the classifier outputs using linear opinion pooling (LOP) [4], $\mathcal{P}(y_i|\mathbf{x}_1,\ldots,\mathbf{x}_m) = \sum_{j=1}^{m} \kappa_j \mathcal{P}_j(y_i|\mathbf{x}_j)$, where $\mathbf{x}_j$'s denote the representation of an image in different feature spaces. Since we want to combine two classifiers: the pixel based classifier

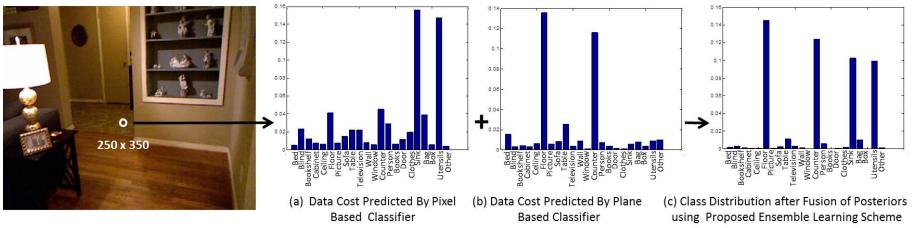

Fig. 2. Effect of Ensemble Learning Scheme: At the pixel location shown in *left most* image, the pixel based appearance model favors class *Sink*. On the other hand, planar regions based appearance model takes care of geometrical properties of region and favors class *Floor*. The right most bar plot shows how our proposed ensemble learning scheme picks the correct class decision.

and the planar region based classifier, we therefore set $m = 2$. After unifying beliefs based on contrasting opinions, the Bayesian rule is used to combine them at the second stage. To try a number of weighting options (r configurations of weights κ) to generate contrasting opinions $\mathbf{o}$, we can represent our ensemble of probabilities as[1], $\mathcal{P}(y_i|\mathbf{o}_1,\ldots,\mathbf{o}_r) = \frac{\mathcal{P}(\mathbf{o}_1,\ldots,\mathbf{o}_r|y_i)\mathcal{P}(y_i)}{\mathcal{P}(\mathbf{o}_1,\ldots,\mathbf{o}_r)}$. Since $\mathbf{o}_1,\ldots,\mathbf{o}_r$ are independent measurements, we have, $\mathcal{P}(y_i|\mathbf{o}_1,\ldots,\mathbf{o}_r) = \frac{\mathcal{P}(\mathbf{o}_1|y_i)\ldots\mathcal{P}(\mathbf{o}_r|y_i)\mathcal{P}(y_i)}{\mathcal{P}(\mathbf{o}_1,\ldots,\mathbf{o}_r)}$. Again applying the Bayes rule and after simplification we get, $\mathcal{P}(y_i|\mathbf{o}_1,\ldots,\mathbf{o}_r) = \rho\frac{\mathcal{P}(y_i|\mathbf{o}_1)\ldots\mathcal{P}(y_i|\mathbf{o}_r)}{\mathcal{P}(y_i)^{r-1}}$. Here,
$\mathcal{P}(y_i)$ is the prior and ρ is a constant which depends on the data and is given by $\rho = \frac{\mathcal{P}(\mathbf{o}_1)\ldots\mathcal{P}(\mathbf{o}_r)}{\mathcal{P}(\mathbf{o}_1,\ldots,\mathbf{o}_r)}$ [4]. The appearance potential is therefore defined by:

$$\phi_i(y_i, \mathbf{x}; \mathbf{w}_u^{app}) = \mathbf{w}_u^{app} \log \mathcal{P}(y_i|\mathbf{o}_1,\ldots,\mathbf{o}_r). \tag{3}$$

The posterior probabilities $\mathcal{P}(y_i|\mathbf{x}_i)$ are estimated using the random forest (RF) classifier. It captures the discriminative features of an image which encode information about shape, texture, context and geometry. We trained the RF with 100 trees and 500 randomly sampled variables as candidates at each split.

Location Potential: The proposed location prior in Eq. 2 models the class distribution based on the orientation and spatial location:

$$\phi(y_i, i; \mathbf{w}_u^{loc}) = \mathbf{w}_u^{loc} \log \mathcal{F}_{loc}(y_i, i), \tag{4}$$

where, $\mathcal{F}_{loc}(y_i, i)$ is defined in Sec. 4.1 and $\mathbf{w}_u^{loc}$ is the parameter. The function $\mathcal{F}_{loc}(y_i, i)$ is dependent on both the location and the orientation of a pixel (Fig. 1, see Sec. 4.1).

[1] In this work we set $r = 3$ and κ is set to $[0.25, 0.75]$, $[0.5, 0.5]$ and $[0.75, 0.25]$ respectively in each case. This choice is based on the validation set (see Sec. 6.2).

3.2 Pairwise Potentials

The pairwise potential in Eq. 1 is defined on the edges $\mathcal{E}$ and takes the form of a boundary aware Potts model:

$$\psi_{ij}(y_{ij}, \mathbf{x}; \mathbf{w}_p) = \mathbf{w}_p^{\mathrm{T}} \phi_{p_1}(y_i, y_j) \phi_{p_2}(\mathbf{x}). \tag{5}$$

The sub-potentials in Eq. 5 are defined as follows.

Class Transition Potential: The CTP in Eq. 5 is a simple zero-one indicator function which enforces a consistent labeling. It is defined as: $\phi_{p_1}(y_i, y_j) = a\mathbf{1}_{y_i \neq y_j}$. For this work we used $a = 10$ based on the validation set (Sec. 6.2).

Spatial Discontinuity Potential: The SDP in Eq. 5 encourages the label transition at the boundaries [32, 33]. It is defined as a combination of edges from the intensity image, depth image and the super-pixel edges extracted using Mean-shift [7] and Felzenswalb [6] segmentation: $\phi_{p_2}(\mathbf{x}) = \mathbf{w}_{p2}^{\mathrm{T}} \phi_{edges}(\mathbf{x})$. Weights assigned to each edge potential are learned using a quadratic program (see Sec. 4.2). In simple terms, edges which match with the manual annotations to a large extent contribute more in the SDP. The edge potential is given by:

$$\phi_{edges}(\mathbf{x}) = [\beta_x \exp(-\tfrac{\sigma_{ij}}{\langle \sigma_{ij} \rangle}), \beta_d \exp(-\tfrac{\sigma_{ij}^d}{\langle \sigma_{ij}^d \rangle}), \beta_{\text{sp-fw}} \mathcal{F}_{\text{sp-fw}}(\mathbf{x}), \beta_{\text{sp-ms}} \mathcal{F}_{\text{sp-ms}}(\mathbf{x}), \alpha]^{\mathrm{T}} \tag{6}$$

where, $\sigma_{ij} = \|x_i - x_j\|^2$, $\sigma_{ij}^d = \|x_i^d - x_j^d\|^2$ and $\langle . \rangle$ denotes the average contrast in an image. x_i and x_i^d shows the color and depth image pixels respectively. $\mathcal{F}_{\text{sp-ms}}$ and $\mathcal{F}_{\text{sp-fw}}$ are indicator functions which give all zeros except at the boundaries of the Mean-shift [7] or Felzenswalb [6] super-pixels respectively. For our case, we set $\alpha = 1$, $\beta_x = \beta_d = 150$ and $\beta_{\text{sp-ms}} = \beta_{\text{sp-fw}} = 5$ based on the validation set (see Sec. 6.2).

3.3 Higher Order Potentials

HOPs incorporate long range interactions and enhance the representational power of the CRF model (Eq. 1). We treat planar patches as n-order cliques and define HOPs on them to eliminate inconsistent variables by encouraging all variables in a clique to take the dominant label. The robust P^n model [19] poses this encouragement in a soft manner and some pixels in a clique may retain different labelings. Hence, it is a linear truncated function of the number of inconsistent variables in a clique. Our proposed HOP enforces consistency by applying a logarithmic penalty:

$$\psi_c(y_c, \mathbf{x}; \mathbf{w}_c) = \mathbf{w}_c \min_{\ell \in \mathcal{L}} \mathcal{F}_c(\tau_c), \tag{7}$$

where, $\mathcal{F}_c(.)$ is a function which takes the number of inconsistent pixels $\tau_c = \#c - n_\ell(\mathbf{y}_c)$ as its argument. $\mathcal{F}_c$ is a non-decreasing concave function of the form $\mathcal{F}_c(\tau_c) = \lambda_{max} - (\lambda_{max} - \lambda_\ell)\exp(-\eta\tau_c)$, where $\eta = \eta_0/Q_\ell$ and $\eta_0 = 5$. Here η_0 is the slope parameter which decides the rate of increase of the penalty, with the increase in the number of pixels disagreeing with the dominant label. The parameters λ_{max} and λ_ℓ define the penalty range which is typically set to 1.5

and 0.15 respectively. Q_ℓ is the truncation parameter which provides the bound for the maximum number of disagreements in a clique. To apply the graph cuts algorithm, details regarding the disintegration of the HOP (Eq. 7) are given in the supplementary material.

4 Learning CRF Model

4.1 Learning Potentials

For a robust semantic labeling, all the characteristics of a class (including its texture, shape, context, geometry and spatial location) need to be taken into account. The procedure of learning this information is outlined as follows.

Features for Local Appearance Potential: The local appearance potential is modeled in a discriminative fashion using a trained classifier (RF in our case). We extract features densely at each point and then aggregate them at the super-pixel[2] level to reduce the computational load and to ensure that similar pixels get a unified representation in the feature space. A rich feature set is extracted which includes local binary patterns (LBP), texton features, SPIN images, scale invariant feature transform (SIFT), color SIFT, depth SIFT and histogram of gradients (HOG). Overall, these features form a high dimensional space (~640 dimensions) and it becomes computationally intensive to train the classifier with all these features. Moreover, some of these features are redundant while some others have a lower accuracy. We therefore employ a genetic search algorithm[3] to find the most useful set of features on the validation dataset (Sec. 6.2).

Features for Appearance Model on Planes: One of the most important features is the plane orientation which is characterized by the direction of its normal. We include the area and height (maximum z-axis value) of the planar region in the feature set to consider its extent and position. Since these measures may vary significantly and a relative measure is needed, we normalize each value with the largest instance in the scene. Moreover, color histograms in the HSV and CIE LAB color spaces are also included. The responses to various filters (in the same manner as *textons*) are calculated and aggregated at the planar level.

Learning Location Potential: Our formulation is based on the idea that the location of a class which has a characteristic geometric orientation can further be made specific, if any geometric information about the scene is available. For example, it is very unlikely to have a *bed* or *floor* at some location in an image, where we know a vertical plane exists. Therefore, we seek to minimize the location prior on the regions where the geometric properties of an object class do not match with the observation made from a scene. First, we average class occurrences over the ground truth for each class (y_i) at each i^{th} location [33, 34]: $\mathcal{F}_{loc}(y_i, i) = \frac{N_{\{y_i,i\}}}{N_i}$. Next, we incorporate geometric information into the location prior. For this, we extract the planar patches (see Sec. 5) and divide them

[2] The super-pixels are obtained using a graph based segmentation method [6].

[3] We use the standard implementation of genetic search algorithm in Weka attribute selector tool [10] to choose the 256 best features.

into three distinct geometrical classes: *below-horizon horizontal regions*, *above-horizon horizontal regions* and *vertical* regions. Since the Kinect sensor gives the pitch and roll for each image, the RGBD images are rotated appropriately to remove any affine transformations. This makes the horizon (estimated using the accelerometer) to lie horizontally at the center of each image. We use this horizon to split the horizontal regions into *above-horizon* and *below-horizon* subclasses. For each planar object class, we retain the 2D location prior in the regions where the geometric properties of the class match with those of the planar region and reduce its value in regions where that class cannot be located. For example, the roof cannot lie on a horizontal plane in the below-horizon region or a vertical region. This effectively reduces the class location prior to only those regions which are consistent with the geometric context. It must be noted that this elimination procedure is only carried out for planar classes e,g., roof, floor, bed and blinds. Finally, the location prior is smoothed and the prior distribution is normalized to give $\sum_i \mathcal{F}_{loc}(y_i, i) = 1/L$ [34].

4.2 Learning Parameters

We used a structured large-margin learning method (S-SVM [36]) to efficiently adjust the probabilistic model parameters. Whilst Szummer *et al.* [36] used the n-slack formulation of cost function, we use a single slack formulation which results in a more efficient learning without any performance degradation[4] [15]. Algorithm 1 shows the learning procedure where the training set $\mathcal{T}$ consists of N training images, $\xi \in \mathbb{R}_+$ is a single slack variable, C is the regularization constant and $\Delta(\mathbf{y}, \mathbf{y}^n)$ is the hamming loss function [36]. It can be proved that the algorithm converges after $\mathrm{O}(1/\epsilon)$ steps [15, 37]. The two major steps in this algorithm are the quadratic optimization step (line 8), which is solvable by off-the-shelf convex optimization problem solvers and the loss augmented prediction step (line 4), which can be solved by graph cuts. Although graph cuts move making algorithm gives an approximate solution, but it is efficient and well suited for the task [16, 36]. To further minimize any chance of getting sub-optimal solution, we initialize the parameters using validation set. With these good initial estimates, S-SVM training converged mostly with in 40 iterations.

We also learn the parameters of the boundary potentials to get a balanced representation of each edge in the SDP potential. In our approach, we define a weighted combination of various possible edge potentials (such as depth edges, contrast based edges, Felzenswalb and mean-shift super-pixels edges) to accommodate information from all these sources (see Sec. 3.2 and Eq. 6). We start with a heuristic based initialization (given by parameters such as β_x and α in Eq. 6) and iterate over the training samples to learn a more balanced representation. Note that here we use double parameterization to minimize the chances of getting into a local minimum. The weights for edges are restrained to be non-negative ($\mathbf{w}_{p2} > 0$) so that the energy remains sub-modular and the graph cuts

[4] Interested readers are referred to [15] for more details and efficiency comparisons between n-slack and 1-slack formulations.

Algorithm 1. S-SVM Training with Rescaled Margin Cutting Plane Algorithm

Input: Training set ($\mathcal{T}$), ϵ tolerance (or convergence threshold), initial parameters $\mathbf{w}_0$
Output: Learned parameters $\mathbf{w}^*$

1: $\mathbf{S} \leftarrow \emptyset$ (working set of low energy labelings that are used as active constraints)
2: **while** $U(y_n, x_n; \mathbf{w}) \geq \epsilon - \xi$ **do**
3: **for** n = 1 ... N **do**
4: $y^* = argmin_{y \in \mathcal{Y}} E(y, x^n; w) - \Delta(y, y^n)$
5: $S = S \cup \{y^*\}$
6: **end for**
7: $(\mathbf{w}, \xi) \leftarrow \underset{\mathbf{w},\xi}{\text{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\xi$
8: s.t. $\frac{1}{N}\sum_{n=1}^{N}[E(\mathbf{y}, \mathbf{x}^n; \mathbf{w}) - E(\mathbf{y}^n, \mathbf{x}^n; \mathbf{w})] \geq \frac{1}{N}\sum_{n=1}^{N}\Delta(\mathbf{y}, \mathbf{y}^n) - \xi \quad ; \quad C > 0, w_i \geq 0.$
9: **end while**
10: where, $U(y_n, x_n; \mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}[E(\mathbf{y}, \mathbf{x}^n; \mathbf{w}) - E(\mathbf{y}^n, \mathbf{x}^n; \mathbf{w})] - \frac{1}{N}\sum_{n=1}^{N}\Delta(\mathbf{y}, \mathbf{y}^n)$

inference can be applied. We use structured learning to learn SDP weights (Sec. 3.2) and the resulting quadratic program is given as follows:

$$\underset{\|\mathbf{w}_{p2}\|=1}{\text{argmax}}\ \gamma \qquad s.t. \quad \{E_{\text{con}}, E_{\text{dep}}, E_{\text{fel-sp}}, E_{\text{ms-sp}}\} - E_{\text{grd}} \geq \gamma, \{\mathbf{w}_{p2}\} \geq 0, \qquad (8)$$

where, E_{grd} is the energy when the SDP is based on the manually identified edges from the training images. Energies for the case when the SDP is based on image contrast, image depth, Felzenswalb or mean-shift super-pixels are represented as $E_{\text{con}}, E_{\text{dep}}, E_{\text{fel-sp}}$ or $E_{\text{ms-sp}}$ respectively. The cost function given in Eq. 8 is optimized in a similar fashion as in Algorithm 1.

5 Plane Detection and Geometric Modeling Scheme

Indoor environments are predominantly composed of structures which can be decomposed into planar regions such as walls, ceilings, cupboards and blinds. We extract the dominant planes which best fit the sparse point clouds of indoor images (obtained from RGBD data) and use them in our model based representation (Fig. 1). It must be noted that depth map from Kinect contains many missing values e.g., along the outer boundaries of an image or when the scene contains a black or a specular surface. Traditional plane detection algorithms (e.g. [29, 35]) either make use of dense 3D point clouds or simply ignore the missing depth regions. In contrast, we propose an efficient plane detection algorithm which is robust to missing depth vlaues (often termed as *holes*) in the Kinect depth map. We expect that the inference made on the improved planar regions will help us achieve a better semantic labeling performance.

Our method[5] first aligns the 3D points with the principal directions of the room. Next, surface normals are computed at each point. Contiguous points in

[5] More details can be found in the supplementary material. Plane detection code is available at `http://www.csse.uwa.edu.au/~salman`

Fig. 3. Comparison of our algorithm (*last* row) with [35] (*middle* row) is shown. Note that the *white* color in middle row shows *non-planar* regions. The *last* row shows detected planes averaged over super-pixels.

Table 1. Comparison of plane detection results on the NYU-Depth v2 dataset. We report detection accuracies for 'exactly planar classes' (EPC) and 'exact and nearly planar classes' (E+NPC).

Performance Evaluation

Method	EPC Acc.	E+NPC Acc.
Silberman et al. [35]	0.69 ± 0.09	0.67 ± 0.10
Rabbani et al. [29]	0.60 ± 0.12	0.57 ± 0.14
This paper	$\mathbf{0.76 \pm 0.09}$	$\mathbf{0.81 \pm 0.07}$

Timing Comparison (averaged for NYU v2)
(for Matlab prog. running on single core, thread)

Silberman [35]	Rabbani [29]	This paper
41 sec	73 sec	3.1 sec

space are then clustered by a region growing algorithm which groups the 3D points in a way to maintain their continuity and smoothness. It is robust to erroneous normal orientations caused due to big holes mostly present along the borders of the depth image acquired via Kinect sensor (Fig. 3). The basic idea is to take help from appearance based cues when the depth information is not reliable. The algorithm begins with a seed point and at each step, a region is grown by including the points in the current region with normals pointing in the same direction. Iteratively, the region is extended and the newly included points are treated as seeds in the subsequent iteration. To deal with erroneous sensor measurements along the border and any other regions with missing depth measurements, we relax the smoothness constraint and use major line segments present in the image to decide about the region continuity.

The line segment detector (LSD) [38] is used to extract the major line segments. These line segments are grouped according to the vanishing points. Line segments in the direction of the major vanishing points contribute more in separating regions during the smoothness constraint based plane detection process. We, however, empirically found that the use of any simple edge detection method (e.g., canny edge detector) in our algorithm gives nearly similar performance with much better efficiency. We further increased the efficiency by replacing iterative region growing with k-means clustering for regions having valid depth values. The planar patches are grown from regions with valid depth values towards regions having missing depths. In this process, segmentation boundaries are predominantly defined by the appearance based edges in an image. Since the majority of the pixels have correct orientation, fitting a plane decreases the orientation errors and the approximate orientation of major surfaces is retained. An added benefit of our algorithm is that curved surfaces are not missed out during the region growing process, rather they are approximated by planes.

Once the regions are grown to the full extent, the small regions are dropped and only the regions with a significant number of pixels are retained. After that, planes are fitted onto the set of points belonging to each region using TLS (Total Least Square) fitting. The least square plane fitting is a non-linear problem, it

however reduces to an eigenvalue problem in the case of planar patches. This makes the plane fitting process highly efficient. It is important to note that although the indoor surfaces are not strictly limited to planes, we assume that we are dealing with planar regions during the plane fitting process. It turns out that this assumption is not a hard constraint since the majority of the surfaces in an indoor environment are either strictly planar (e.g., walls, ceilings) or nearly planar (e.g., beds, doors). Finally, our algorithm is superior to other region growing algorithms (e.g., [29]) which are suitable for the segmentation of dense point clouds and fail to deal with the erroneous depth measurements from the Kinect sensor (Fig. 3 and Table 1).

6 Experiments and Analysis

6.1 Datasets

We evaluated our framework on the New York University (NYU) Depth dataset (v2) and a recent SUN3D dataset. The NYU dataset [34] consists of 1449 labeled images. SUN3D is a large scale indoor RGBD dataset [40], however it's still under development and only a small portion has been labeled. We extracted keyframes from SUN3D which amounted to 83 labeled images.

6.2 Results

In the NYU-Depth v2, around 900 different object classes are present in all indoor scenes. Since not all object classes have a sufficient representation, we follow the procedure in [34] to cluster the existing annotations into the 22 most frequently occurring classes. This clustering is performed using the Wordnet Natural Language Toolkit (NLTK). For the case of SUN3D dataset, 32 classes are present in the labeled images we acquired. We clustered them into 13 major classes using Wordnet. In both the datasets, a supplementary class labeled '*other*' is also included to model rarely occurring objects. In our evaluations, we exclude all unlabeled regions. For both the datasets, 60%/40% train/test split was used. A relatively small validation set consisting of 50 random images was extracted from NYU-Depth v2. This validation set was used with the genetic search algorithm for the selection of useful features and for the choice of the initial estimates of the parameters which gave the best performance (for SUN3D we used the same parameters). Afterwards, these parameters were optimized during the learning process as described in Sec. 4.2.

We used two popular evaluation metrics to assess our results, '*pixel accuracy*' and '*class accuracy*' (see Table 2). Pixel accuracy accounts for the average number of pixels which are correctly classified in the test set. Class accuracy measures the average of the correct class predictions which is essentially equal to the mean of the values occurring at the diagonal of the confusion matrix. We extensively evaluated our approach on both the NYU-Depth and SUN3D datasets. Our experimental results are shown in Table 2. The comparisons with state-of-the-art

Table 2. Semantic Labeling Performance: We report the results of our proposed framework when only variants of unary potentials were used (top 3 rows), a CRF with regular Potts model was used (second last row) and the improvements observed when more sophisticated priors and HOPs (last row) were added. Accuracies are reported for 22 and 13 class semantic labeling for NYU v2 and SUN3D datasets respectively.

Variants of Our Method	NYU-Depth v2		SUN3D	
	Pixel Accuracy	Class Acc.	Pixel Accuracy	Class Acc.
Feature Ensemble (FE)	$44.4 \pm 15.8\%$	39.2%	$41.9 \pm 11.1\%$	40.0%
FE + Planar Appearance Model (PAM)	$52.5 \pm 15.5\%$	42.4%	$48.3 \pm 11.5\%$	42.6%
FE + PAM + Planar Location Prior (PLP)	$55.3 \pm 15.8\%$	43.1%	$51.5 \pm 11.9\%$	43.3%
FE + PAM + PLP + CRF (Regular Potts Model)	$55.5 \pm 15.8\%$	43.2%	$51.8 \pm 12.0\%$	43.5%
FE + PAM + PLP + CRF (SDP + HOP)	$\mathbf{58.3 \pm 15.9}\%$	**45.1**%	$\mathbf{54.2 \pm 12.2}\%$	**44.7**%

Table 3. Comparison of results on the NYU-Depth v2 (4-class labeling task): Our method achieved best performance in terms of average pixel and class accuracies

Method	Semantic Classes				Pixel Accuracy	Class Accuracy
	Floor	Structure	Furniture	Props		
Supp. Inf. [35]	68	59	**70**	**42**	58.6	59.6
ConvNet [5]	68.1	87.8	51.1	29.9	63	59.2
ConvNet + D [3]	87.3	86.1	45.3	35.5	64.5	63.5
Im ∪ 3D [1]	**87.9**	79.7	63.8	27.1	67.0	64.3
This paper	87.1	**88.2**	54.7	32.6	**69.2**	**65.6**

techniques are shown in Tables 3, 4. Sample labelings for NYU-Depth v2 are presented in Fig. 4. Although the unlabeled portions in the annotated images are not considered during our evaluations, we observed that the labeling scheme mostly predicts accurate class labels (see Fig. 4).

We report our results in terms of average pixel and class accuracies in Table 2. Starting from a simple unary potential defined on pixels using an ensemble of features, we achieve pixel and class accuracies of 44.4% and 39.2% respectively on NYU-Depth v2. The corresponding accuracies for SUN3D are 41.9% and 40.0% respectively. Starting from these moderate accuracies we build up and get significant improvements. Upon the introduction of the planar appearance model, the pixel and class accuracies increased by 8.1% and 3.2% from their previous values for NYU-Depth v2. For the SUN3D database, we get an increase of 6.4% and 2.6% in pixel and class accuracies respectively. The addition of CRF and modified location potential along with the HOP enforced a better label consistency and the results were consequently improved by 5.8% and 2.7% for NYU-Depth v2, 5.9% and 2.4% for SUN3D datasets. By comparing last two rows in Table 2, it can be seen that the proposed SDP performs better much than the regular Potts model.

For the case of NYU-Depth v2, we compare our framework with a recent multi-scale ConvNet based technique [3, 5]. Whereas in [3, 5] evaluations were performed on just 13 classes, we use a broader range of 22 classes to report our results (see Table 4). To compare with the class *sofa*, we report the mean accuracies of the *sofa* and *chair* classes for a fair comparison[6]. We compare the *furniture* class in [3] with our *cabinet* class based on the details given in [3].

[6] If we sum up the class occurrences of the *chair* and *sofa* which are reported in [3], it supports such comparison.

Table 4. Class wise Accuracies on NYU-Depth v2: Our proposed framework achieves the highest accuracy on 19/22 classes. With nearly double number of classes used in [3, 5], we get $\sim 6\%$ and $\sim 9\%$ improvement in class and pixel accuracies respectively.

Method	Bed	Blind	Bookshelf	Cabinet	Ceiling	Floor	Picture	Sofa	Table	Television	Wall	Window	Counter	Person	Books	Door	Clothes	Sink	Bag	Box	Utensils	Other	Unlabeled	Mean Class Accuracy	Mean Pixel Accuracy	Classes
Class Freq.	4.7	2.0	4.2	10.7	1.4	10.8	2.2	6.2	2.6	0.5	22.8	2.3	2.7	1.7	0.9	2.3	1.7	0.3	1.7	0.8	0.2	0.1	17.4	-	-	-
ConvNet [5]	30.3	-	31.7	28.5	33.2	68.0	-	35.1	18.0	18.8	**89.4**	37.8	-	-	-	-	-	-	-	-	-	-	-	35.8	51.0	13
CNN+D [3]	**38.1**	-	13.7	42.4	62.6	**87.3**	-	29.8	10.2	6.0	86.1	15.9	-	-	-	-	-	-	-	-	-	-	-	36.2	52.4	13
This paper	32.3	**56.9**	**38.3**	**45.6**	**64.7**	75.8	**43.6**	**58.6**	**47.9**	**45.7**	77.5	**54.0**	**43.8**	**38.8**	**34.0**	**58.3**	**37.2**	**23.1**	**28.4**	**35.7**	**22.6**	**29.9**	-	**45.1**	**58.3**	22

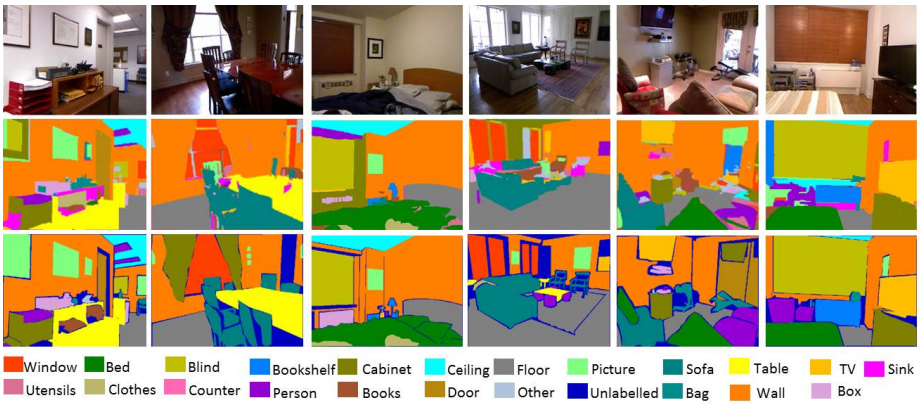

Fig. 4. Examples of semantic labeling results on the NYU-Depth v2 dataset. Figure shows intensity images (*top* row), ground truths (*bottom* row) and our results (*middle* row). Our framework performs well in many cases including some unlabeled regions.

Overall, we get superior performance compared to [3, 5] and also achieve best class accuracies in 19/22 classes.

On NYU-Depth v2, Silberman *et al.* defined just four semantic classes: *furniture*, *ground*, *structure* and *props* [35]. The main goal of [35] was to infer support relationships between objects, for which such a class selection was justified. For our application, such a small number of classes will be meaningless. However, for the sake of comparison we evaluated our method on the 4-class segmentation task as well. As shown in Table 3, we achieved the best performance over all. Particularly we performed well on planar classes such as *floor* and *structures*. In terms of pixel and class accuracies, we noted an improvement of 2.2% and 1.3% respectively. Very recently, Muller and Behnke [25] have reported state-of-the-art labeling performance on NYU-Depth v2. In comparison to [25], which reported results on just 4 classes, our method performs also well on a larger set of 22 classes which demonstrates its scalability.

One may wonder why the incorporation of geometrical context in the CRF model works and gives such high accuracies? In v2 of the NYU-Depth dataset, there are nearly ten out of 22 classes (bed, blind, cabinet, ceiling, floor, picture, table, wall, counter, door) which are planar and out of the remaining classes, 6 are loosely planar (tv, sofa, bookshelf, window, box, sink). The planar classes

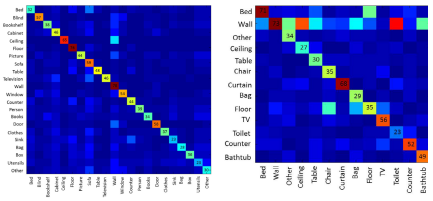

Fig. 5. Confusion Matrices for NYU-Depth v2 (*left*) and SUN3D *right* Databases. All the class accuracies shown on the diagonal are rounded to the closest integer for clarity.

correspond to 62.2% while the loosely planar classes correspond to 14.3% of the total labeled data. There is a similar trend on the SUN3D database. Note that classes such as *floor* or *wall* may have varying textures across different images. However, with depth information in place, we can determine the correct class of the object. Our approach is efficient at test time, since the proposed graph energies are sub-modular and approximate inference can be made using graph-cuts. Empirically, we found average testing time per image to be ~ 1.7 sec for NYU-Depth and ~ 1.4 sec for SUN3D database. For parameter learning on the training set, it took ~ 12 hrs for NYU-Depth and ~ 45 min for SUN3D database.

From the achieved performances (Table 2), it can be seen that indoor scene labeling is a challenging problem due to the diverse nature of the scenes and the presence of a large number of objects. Many times, class errors occurred due to the confusion between two similar classes e.g., *door* is usually confused with *wall* and *blind* with *window* (see Fig. 5). Some misclassifications occurred due to illumination variations, specular surfaces and shadows. In future work, we will explore the use of shadow removal methods like [17] to enhance the labeling accuracy. Lastly, the datasets are somewhat unbalanced and a sufficient representation of all classes is not present in the training set. The labeled portion of SUN3D database is really small (because the database has been released recently) and this is why the achieved accuarcies are on the lower side (see Table 2). The availability of more and higher quality training data for each class will certainly improve the quality of scene labelings.

7 Conclusion

With the availability of depth data for indoor scenes, a pressing issue is to leverage this information in a better way. We extract geometric information from indoor scenes using a novel region growing algorithm which uses dominant lines and surface normals to group the pixels. We use this information at a number of levels in the proposed CRF model. First, we accommodate a posterior defined on planar regions in the appearance based potential to reinforce our beliefs on the dominant planar classes. We also include geometry aware location priors and HOPs defined over n-order cliques to encourage the pixels lying on a planar region to adopt the same labeling. The pairwise potential in our model is defined as a combination of various edges learned using a quadratic program. We extensively evaluated our scheme on the NYU-Depth and the SUN3D databases and report comparisons and improvements over existing works.

Acknowledgments. This research was supported by the Australian Research Council (ARC) grants DP110102166 and DE120102960.

References

[1] Cadena, C., Košecká, J.: Semantic segmentation with heterogeneous sensor coverages (2014)
[2] Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. TPAMI 34(7), 1312–1328 (2012)
[3] Couprie, C., Farabet, C., Najman, L., LeCun, Y.: Indoor semantic segmentation using depth information. In: ICLR (2013)
[4] Edwards, W., Miles Jr., R.F., Von Winterfeldt, D.: Advances in decision analysis: from foundations to applications. Cambridge University Press (2007)
[5] Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. TPAMI 35(8), 1915–1929 (2013)
[6] Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59(2), 167–181 (2004)
[7] Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. TIT 21(1), 32–40 (1975)
[8] Gould, S., Baumstarck, et al.: Integrating visual and range data for robotic object detection. In: Workshop on M2SFA2 (2008)
[9] Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV, pp. 1–8. IEEE (2009)
[10] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
[11] Hayat, M., Bennamoun, M., An, S.: Learning non-linear reconstruction models for image set classification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2014)
[12] He, X., Zemel, R.S., Carreira-Perpinán, M.A.: Multiscale conditional random fields for image labeling. In: CVPR, vol. 2, pp. II–695. IEEE (2004)
[13] Huang, Q., Han, M., Wu, B., Ioffe, S.: A hierarchical conditional random field model for labeling and segmenting images of street scenes. In: CVPR, pp. 1953–1960. IEEE (2011)
[14] Jiang, Y., Lim, M., et al.: Learning to place new objects in a scene. IJRR 31(9), 1021–1043 (2012)
[15] Joachims, T., Finley, T., Yu, C.N.J.: Cutting-plane training of structural svms. JML 77(1), 27–59 (2009)
[16] Kappes, J.H., Andres, B., Hamprecht, F.A., Schnorr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Lellmann, J., Komodakis, N., et al.: A comparative study of modern inference techniques for discrete energy minimization problems. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1328–1335. IEEE (2013)
[17] Khan, S., Bennamoun, M., Sohel, F., Togneri, R.: Automatic feature learning for robust shadow detection. In: CVPR. IEEE (2014)
[18] Kohli, P., Kumar, M.P., Torr, P.H.: P3 & beyond: Solving energies with higher order cliques. In: CVPR, pp. 1–8. IEEE (2007)
[19] Kohli, P., Torr, P.H., et al.: Robust higher order potentials for enforcing label consistency. IJCV 82(3), 302–324 (2009)
[20] Koppula, H.S., Anand, A., et al.: Semantic labeling of 3D point clouds for indoor scenes. In: NIPS, pp. 244–252 (2011)

[21] Ladicky, L., Russell, C., Kohli, P., Torr, P.H.: Associative hierarchical crfs for object class image segmentation. In: ICCV, pp. 739–746. IEEE (2009)
[22] Ladickỳ, L., Russell, C., et al.: Inference methods for crfs with co-occurrence statistics. IJCV, 1–13 (2013)
[23] Lempitsky, V., Vedaldi, A., Zisserman, A.: Pylon model for semantic segmentation. In: NIPS, pp. 1485–1493 (2011)
[24] Li, Y., Tarlow, D., Zemel, R.: Exploring compositional high order pattern potentials for structured output learning (June 2013)
[25] Muller, A., Behnke, S.: Learning depth-sensitive conditional random fields for semantic segmentation of rgb-d images. In: ICRA (2014)
[26] Munoz, D., Bagnell, J.A., Hebert, M.: Stacked hierarchical labeling. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 57–70. Springer, Heidelberg (2010)
[27] Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR, pp. 413–420 (2009)
[28] Quigley, M., Batra, S., et al.: High-accuracy 3D sensing for mobile manipulation: Improving object detection and door opening. In: ICRA, pp. 2816–2822. IEEE (2009)
[29] Rabbani, T., van Den Heuvel, F., Vosselmann, G.: Segmentation of point clouds using smoothness constraint. Intl. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36(5), 248–253 (2006)
[30] Rao, D., Le, Q.V., et al.: Grasping novel objects with depth segmentation. In: IROS, pp. 2578–2585. IEEE (2010)
[31] Ren, X., Bo, L., Fox, D.: Rgb-(d) scene labeling: Features and algorithms. In: CVPR, pp. 2759–2766. IEEE (2012)
[32] Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. TOG 23, 309–314 (2004)
[33] Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV 81(1), 2–23 (2009)
[34] Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: ICCV Workshops, pp. 601–608. IEEE (2011)
[35] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)
[36] Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
[37] Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML, p. 104. ACM (2004)
[38] Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. TPAMI 32(4), 722–732 (2010)
[39] Woodford, O.J., Rother, C., Kolmogorov, V.: A global perspective on map inference for low-level vision. In: ICCV, pp. 2319–2326. IEEE (2009)
[40] Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: ICCV. IEEE (2013)
[41] Xiong, X., Huber, D.: Using context to create semantic 3D models of indoor environments. In: BMVC, pp. 45–41 (2010)