Int J Comput Vis (2016) 117:1–20
DOI 10.1007/s11263-015-0843-8

Integrating Geometrical Context for Semantic Labeling of Indoor Scenes using RGBD Images

Salman H. Khan[1] · Mohammed Bennamoun[1] · Ferdous Sohel[1] · Roberto Togneri[2] · Imran Naseem[3]

Received: 23 November 2014 / Accepted: 24 June 2015 / Published online: 3 July 2015

Abstract Inexpensive structured light sensors can capture rich information from indoor scenes, and scene labeling problems provide a compelling opportunity to make use of this information. In this paper we present a novel conditional random field (CRF) model to effectively utilize depth information for semantic labeling of indoor scenes. At the core of the model, we propose a novel and efficient plane detection algorithm which is robust to erroneous depth maps. Our CRF formulation defines local, pairwise and higher order interactions between image pixels. At the local level, we propose a novel scheme to combine energies derived from appearance, depth and geometry-based cues. The proposed local energy also encodes the location of each object class by considering the approximate geometry of a scene. For the pairwise interactions, we learn a boundary measure which defines the spatial discontinuity of object classes across an image. To model higher-order interactions, the proposed energy treats smooth surfaces as cliques and encourages all the pixels on a surface to take the same label. We show that the proposed higher-order energies can be decomposed into pairwise submodular energies and efficient inference can be made using the graph-cuts algorithm. We follow a systematic approach which uses structured learning to fine-tune the model parameters. We rigorously test our approach on SUN3D and both versions of the NYU-Depth database. Experimental results show that our work achieves superior performance to state-of-the-art scene labeling techniques.

Keywords Scene parsing · Graphical models · Geometric reasoning · Structured learning

1 Introduction

The main goal of scene understanding is to equip machines with human-like visual interpretation and comprehension capabilities. A fundamental task in this process is that of *scene labeling*, which is also well-known as *scene parsing*. In this task, each of the smallest discrete elements in an image (*pixels* or *voxels*) is assigned a semantically-meaningful class label. In this manner, the scene labeling problem unifies the conventional tasks of object recognition, image segmentation, and multi-label classification (Farabet et al. 2013). A high-performance scene labeling framework is useful for the design and development of context-aware personal assistant systems, content-based image search engines and domestic robots, among several other applications.

From a scene-labeling viewpoint, scenes can broadly be classified into two groups: indoor and outdoor. The task of indoor scene labeling is relatively difficult in comparison to

Communicated by Derek Hoiem.

✉ Salman H. Khan
salman.khan@research.uwa.edu.au;
salman.khan@uwa.edu.au

Mohammed Bennamoun
mohammed.bennamoun@uwa.edu.au

Ferdous Sohel
ferdous.sohel@uwa.edu.au

Roberto Togneri
roberto.togneri@uwa.edu.au

Imran Naseem
imrannaseem@pafkiet.edu.pk

[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

[3] Department of Engineering, Karachi Institute of Economics and Technology, Karachi 75190, Pakistan

its outdoor counterpart (Quattoni and Torralba 2009). There are many different types of indoor scenes (e.g. consider a corridor, a bookstore or a kitchen), and it is non-trivial to handle them all in a unified way. Moreover, in contrast to common outdoor scenes, indoor scenes more often contain illumination variations, clutter and a variety of objects with imbalanced representations. In many outdoor scenes, common classes (e.g. ground, sky and vegetation) do not exhibit much variability, whereas objects in indoor scenes can change their appearance significantly between different images (e.g. a bed may change appearance due to different bedsheets). Such difficulties can prove challenging when performing scene labeling purely from color (RGB) images. However, with the advent of consumer-grade sensors such as the Microsoft Kinect that capture co-registered color (RGB) and depth (D) images of indoor scenes, a much richer source of information has become available (Hayat et al. 2015). A number of popular and relevant databases e.g., NYU-Depth (Silberman and Fergus 2011), RGBD Kinect (Lai et al. 2011) and SUN3D (Xiao et al. 2013) have been acquired using the Kinect sensor. These notable efforts have opened the door to the development of improved schemes for labeling indoor scenes from RGBD images.

Various recent works have focused on the use of RGBD images for labeling indoor scenes. Koppula et al. (2011) used KinectFusion (Izadi et al. 2011) to create a 3D point cloud and then densely labeled it using a Markov Random Field (MRF) model. Silberman and Fergus (2011) provided a Kinect-based dataset for indoor scene labeling and achieved decent semantic labeling performance using a Conditional Random Field (CRF) model with SIFT features and 3D location priors. Although they showed that depth information has significant potential to improve scene labeling performance, their own work was limited to depth-based features and priors, and did not explore the possibilities of effectively utilising the scene geometry or exploiting long-range interactions between pixels. In this work, we develop a novel depth-based geometrical CRF model to efficiently and effectively incorporate depth information in the context of scene labeling. We propose that depth information can be used to explore the geometric structure of the scene, which in turn will help colorblue with the scene labeling task. We propose to incorporate depth information in all the components of our hierarchical probabilistic model (unary, pairwise and higher-order). Our model uses both intensity and depth information for efficient segmentation.

For the purpose of integrating depth information, we begin with the modification of unary potentials. First, we incorporate geometric information in the most important energy of our CRF model, namely the appearance energy. In this local energy, we encode both appearance and depth-based characteristics in the feature space. These features are used to predict the local energies in a discriminative fashion. Note that in general, man-made environments contain a lot of flat structures, because they are easier to manufacture than curved ones. Therefore we extract planes, which are the fundamental geometric units of indoor scenes, using a new smoothness constraint based *'region growing algorithm'* (see Sect. 5). Compared to other plane detection methods (e.g., Rabbani et al. (2006); Silberman et al. (2012)), our method is robust to large holes which can potentially appear in the Kinect's depth maps (Sect. 5). The geometric as well as the appearance based characteristics of these planar patches are used to provide unary estimates. We propose a novel *'decision fusion scheme'* to combine the pixel and planar based unary energies. This scheme first uses a number of contrasting opinion pools and finally combines them using a Bayesian framework (see Sect. 3.1.1). Next, we consider the location based local energy that encodes the possible spatial locations of all classes. Along with the conventional 2D location prior, we propose to use the planar regions in each image to channelize the location energy (see Sect. 3.1.2).

Our approach also incorporates depth information in the pairwise and higher-order clique potentials. We propose a novel *'spatial discontinuation energy'* in the pairwise smoothness model. This energy combines evidence from several edge detectors (such as depth edges, contrast based edges and different 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 Sect. 4.1). Finally, we propose a higher-order term in our CRF model which is defined on cliques that encompass planar surfaces. The proposed Higher-Order Energy (HOE) increases the expressivity of the random field model by assimilating the geometrical context. This encourages all pixels inside a planar surface to take a consistent labeling. We also propose a logarithmic penalty function (see Sect. 3.3) and prove that the HOE can be decomposed into sub-modular energy functions (see Appendix).

To efficiently learn the parameters of our proposed CRF model, we use a max-margin learning algorithm which is based on a one-slack formulation (Sect. 4.1).

The rest of the paper is organized as follows. We discuss related work in the next section and propose a random field model in Sect. 3. We then outline our parameter learning procedure in Sect. 4. In Sect. 5, the details of our proposed geometric modeling approach are presented. We evaluate and compare our proposed approach with related methods in Sect. 6 and the paper finally concludes in Sect. 7.

2 Related Work

The use of range or depth sensors for scene analysis and understanding is increasing. Recent works employ depth information for various purposes e.g., semantic segmentation

(Koppula et al. 2011), object grasping (Rao et al. 2010; Khan et al. 2015), door-opening (Quigley et al. 2009) and object placement (Jiang et al. 2012). For the case of semantic labeling, works such as Silberman and Fergus (2011); Silberman et al. (2012) and Silberman et al. (2012) demonstrate the potential depth information has to help with vision-related tasks. However, they do not go beyond the depth-based features or priors. In this paper, we show how to incorporate depth information into the various components of a random field model and then evaluate the contribution made by each component in enhancing semantic labeling performance (Khan et al. 2014b). Our framework is particularly inspired by the works on semantic labeling of RGBD data (Silberman and Fergus 2011; Silberman et al. 2012), considering long-range interactions (Kohli et al. 2009), parametric learning (Szummer et al. 2008; Tsochantaridis et al. 2004) and geometric reconstruction (Rabbani et al. 2006).

The *scene parsing* problem has been studied extensively in recent years. Probabilistic graphical models, e.g. MRFs and CRFs, have been successfully applied to model context and provide a consistent labeling (He et al. 2004; Gould et al. 2009; Lempitsky et al. 2011; Huang et al. 2011). Some of these methods, e.g. Gould et al. (2009), work on a pixel grid, whilst others perform inference at the super-pixel level (Huang et al. 2011). He et al. (2004) combined local, regional and global cues to formulate multi-scale CRFs to address the image labeling problem. Hierarchical MRFs are employed in Ladicky et al. (2009) to perform joint inference on pixels and super-pixels. Huang et al. (2011) trained their CRF on separate clusters of similar scenes and used the clusters with standard CRF to label street images. Silberman and Fergus (2011) showed that when segmenting RGBD data, it is possible to achieve better results by making use of all the available channels (including depth) than by relying on RGB alone. They used features extracted from the depth channel and a 3D location prior to incorporate depth information. However, the question of how to incorporate depth information in an optimal manner remains unanswered and warrants further investigation. Moreover, although works such as Silberman and Fergus (2011); Xiong and Huber (2010) and Xiong and Huber (2010) use depth-based features to enhance segmentation performance, they do not incorporate depth information into the higher-order components of the CRF.

Another important challenge in scene labeling is to take account of *long-range context* in the scene when making local labeling decisions. Farabet et al. (2013) extracted dense features at a number of scales and thereby encoded multiple regions of increasing size and decreasing resolution at each pixel location. Other works have incorporated long-range context by generating a number of segmentations at various scales (often arranged as trees) to propose many possible labelings [e.g., Ladicky et al. (2009); Carreira and Sminchisescu (2012)]. HOEs have been employed to model long-range smoothness (Kohli et al. 2009), shape-based information (Li et al. 2013; Gulshan et al. 2010), cardinality-based potential (Woodford et al. 2009) and label co-occurrences (Ladickỳ et al. 2013). While densely-connected pairwise models such as Krähenbühl and Koltun (2011) are suitable for fine-grained segmentation, indoor scenes rarely require such full connectivity because most of the candidate classes exhibit definite boundaries unlike e.g. trees or cat fur. In contrast to previously-proposed HOEs, we propose using the geometrical structure of the scenes to model high-level interactions.

Currently popular *parameter estimation* methods include partition function approximations (Shotton et al. 2009), cross validation (Shotton et al. 2009) or simply hand picked parameters (Silberman and Fergus 2011). We used a one-slack formulation (Joachims et al. 2009) of the parameter learning technique of Szummer et al. (2008), which gives a more efficient optimization of the cost function compared to the n-slack formulation employed in Tsochantaridis et al. (2004); Szummer et al. (2008) and Szummer et al. (2008). Further, we extend the parameter estimation problem to consider multiple edge-based energies and learn parameters using a quadratic program.

Our *geometric reconstruction* scheme is close to the one used by Xiong and Huber (2010) to create semantic 3D models of indoor scenes and the smoothness constraint-based segmentation technique of Rabbani et al. (2006). Whilst both these schemes use data from accurate laser scanners, we improved their algorithm to make it suitable to tackle the less accurate depth data acquired by a low-cost Microsoft Kinect sensor that operates in real time. Our proposed algorithm relaxes the smoothness constraint in missing depth regions and considers more reliable appearance cues to define planar surfaces.

3 Proposed Conditional Random Field Model

As a prelude to the development of a hierarchical appearance model and a HOE defined over planes (Fig. 1), we first outline briefly the conditional random field model and its components. We use a CRF to capture the conditional distribution of output classes given an input image. The CRF model takes into consideration the color, location, texture, boundaries and layout of pixels to reason about a set of semantically-meaningful classes. The CRF model is defined on a graph composed of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. We want the model to capture not only the interactions between direct neighbours in the graph, but also long-range interactions between nodes that form part of the same planar regions (Fig. 2). To achieve this, we treat our problem as a graphical probabilistic segmentation process in which a graph $\mathcal{G}(\mathcal{I}) = \langle \mathcal{V}, \mathcal{E} \rangle$ is defined over an image $\mathcal{I}$ (Blake et al. 2011).

The set of vertices $\mathcal{V}$ represents individual pixels in a graph defined on $\mathcal{I}$. If the set cardinality ($\#\mathcal{V}$) is T then the vertex

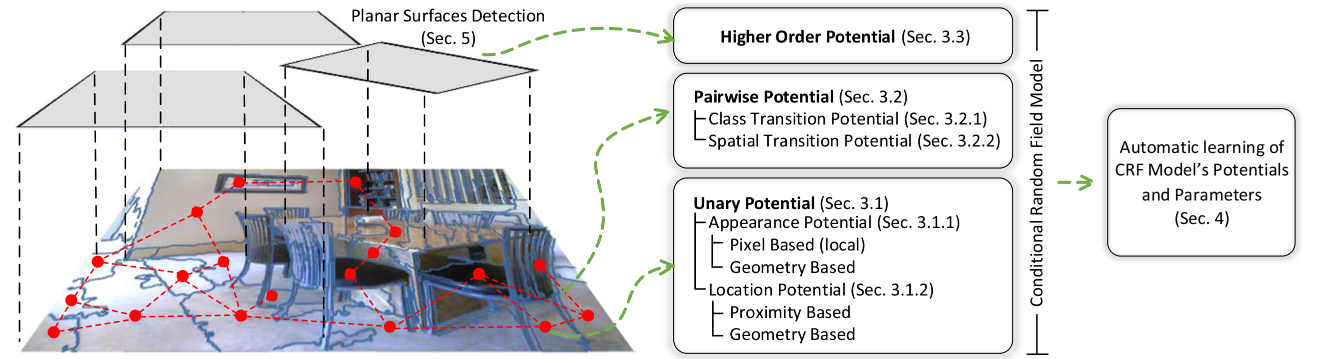
Planar Surfaces Detection
(Sec. 5)
Higher Order Potential (Sec. 3.3)
Pairwise Potential (Sec. 3.2)
Class Transition Potential (Sec. 3.2.1)
Spatial Transition Potential (Sec. 3.2.2)
Unary Potential (Sec. 3.1)
Appearance Potential (Sec. 3.1.1)
Pixel Based (local)
Geometry Based
Location Potential (Sec. 3.1.2)
Proximity Based
Geometry Based
Conditional Random Field Model
Automatic learning of CRF Model's Potentials and Parameters (Sec. 4)

Fig. 1 The figure summarizes our proposed approach to combine global geometric information with low-level cues. Only limited graph nodes are shown for the purpose of a clear illustration

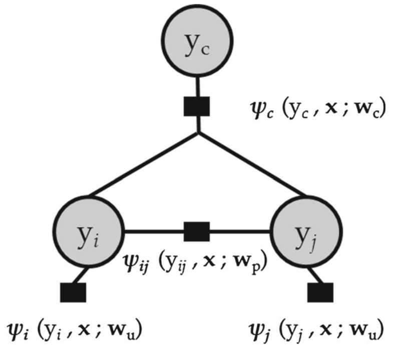

Fig. 2 A factor graph representation for our CRF model. The bottom layer represents pixels and the top layer represents planar regions. Each circle represents a latent class variable while black boxes represent terms in the CRF model (Eq. 2)

set represents all the pixels: $\mathcal{V} = \{p_i : i \in [1, \mathrm{T}]\}$. Similarly, $\mathcal{E}$ represents a set of edges which connect adjacent vertices in $\mathcal{G}(\mathcal{I})$. These edges are undirected based on the assumption of conditional independence between the nodes. 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 $\ell_i \in \mathcal{L}$. The set of all possible classes is given by $\mathcal{L} = \{1, \dots, L\}$ and the total number of classes is $\#\mathcal{L} = L$.

If the estimated labeling of an image $\mathcal{I}$ is represented by a vector $\mathbf{y}$, where $\mathbf{y} = (y_i : i \in [1, \mathrm{T}]) \in \mathcal{L}^{\mathrm{T}}$ is composed of discrete random variables associated with each vertex in $\mathcal{G}(\mathcal{I})$, we have the likelihood of labeling $\mathbf{y}$ decomposed into node and maximal clique potentials as follows:

$$\mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) = \frac{1}{Z(\mathbf{w})}\prod_{i\in\mathcal{V}}\theta_u^{\mathbf{w}_u}(y_i,\mathbf{x}) \prod_{\{i,j\}\in\mathcal{E}}\theta_p^{\mathbf{w}_p}(y_{ij},\mathbf{x})\prod_{c\in\mathcal{C}}\theta_c^{\mathbf{w}_c}(y_c,\mathbf{x}) \qquad (1)$$

where, $\mathbf{x}$ denotes the observations made from an image $\mathcal{I}$, $Z(\mathbf{w})$ is a normalizing constant known as the partition function, $\mathbf{w}$ represents a vector which parametrizes the model and $\mathbf{w}_u$, $\mathbf{w}_p$ and $\mathbf{w}_c$ are the components of $\mathbf{w}$ which parametrize the unary, pairwise and higher-order potential functions. The variables y_i, y_{ij} and y_c represent the labeling over node i, pairwise clique $\{i, j\}$ and the higher-order clique c respectively. The potential functions associated with y_i, y_{ij} and y_c are denoted by θ_u, θ_p and θ_c, respectively. The conditional distribution in Eq. 1 for each possible labeling $\mathbf{y} \in \mathcal{L}^{\mathrm{T}}$ can be represented by an exponential formulation in terms 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 can be defined in terms of log-likelihoods:

$$\begin{aligned}\mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w}) &= -\log(\mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w})\; Z(\mathbf{w}))\\ &= \sum_{i\in\mathcal{V}}\psi_u(y_i,\mathbf{x};\mathbf{w}_u) + \sum_{\{i,j\}\in\mathcal{E}}\psi_p(y_{ij},\mathbf{x};\mathbf{w}_p)\\ &\quad+\sum_{c\in\mathcal{C}}\psi_c(y_c,\mathbf{x};\mathbf{w}_c). \end{aligned} \qquad (2)$$

These three terms in Eq. 2, in which the Gibbs energy has been decomposed (using Eq. 1) are called the unary, pairwise and higher order energies respectively (Fig. 2). These energies are related to the potential functions defined in Eq. 1 by: $\theta_k^{\mathbf{w}_k}(y_k,\mathbf{x}) = \exp(-\psi(y_k,\mathbf{x};\mathbf{w}_k))$ with $k \in \{u, p, c\}$. We will describe the unary, pairwise and higher order energies in Sects. 3.1, 3.2 and 3.3, respectively.

In the inference stage, the most likely labeling is chosen using Maximum a Posteriori (MAP) estimation over possible labelings $\mathbf{y} \in \mathcal{L}^{\mathrm{T}}$, and denoted $\mathbf{y}^*$:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{L}^{\mathrm{T}}}{\operatorname{argmax}}\; \mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) \qquad (3)$$

Since the partition function $Z(\mathbf{w})$ does not depend on $\mathbf{y}$, Eq. 3 can be reformulated as an energy minimization problem, as follows:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{L}^{\mathrm{T}}}{\operatorname{argmin}}\; \mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w}) \qquad (4)$$

The parameter vector $\mathbf{w}$, introduced in Eq. 4, is learnt using a max-margin criterion (see Sect. 4.1 for details).

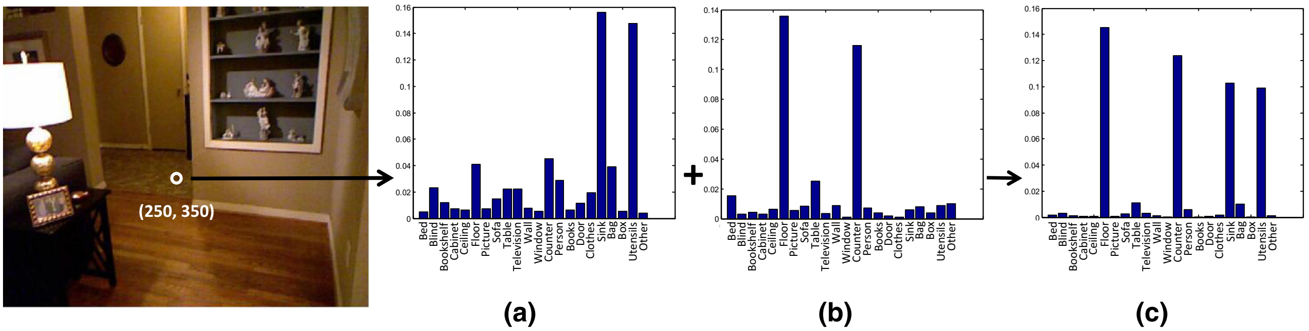

Fig. 3 Effect of the Ensemble Learning Scheme: At the pixel location, shown in the figure, the posterior predicted by the local appearance model favors the class *Sink*. On the other hand, the planar regions based appearance model takes care of the geometrical properties of the region and favors the class *Floor*. The right most bar plot shows how our proposed ensemble learning scheme picks the correct class decision. (*Best viewed in color*)

3.1 Unary Energies

The unary energy in Eq. 2 is further decomposed into two components, appearance energy and location energy (Fig. 1):

$$\sum_{i\in\mathcal{V}} \psi_u(y_i, \mathbf{x}; \mathbf{w}_u) = \sum_{i\in\mathcal{V}} \overbrace{\phi_i(y_i, \mathbf{x}; \mathbf{w}_u^{app})}^{\text{appearance}} + \sum_{i\in\mathcal{V}} \overbrace{\phi_i(y_i, i; \mathbf{w}_u^{loc})}^{\text{location}} \tag{5}$$

We describe both terms in the following sections.

3.1.1 Proposed Appearance Energy

The proposed appearance energy (first term) in Eq. 5 is defined over the pixels and the planar regions (Fig. 1). We use the class predictions defined over the planar regions to improve the posterior defined over the pixels. In other words, planar features are used to reinforce beliefs for some dominant planar classes (e.g., walls, blinds, floor and ceiling). To combine the local appearance and the geometric information, we use a hierarchical ensemble learning method (Fig. 3). Our technique combines two axiomatic ensemble learning approaches; linear opinion pooling (LOP) and the Bayesian approach. Note that we have outputs from a pixel based classifier which operates on pixels, and a planar regions based classifier which works on planar regions. With these outputs, we *first* fuse them using a simple LOP which produces a weighted combination of both classifier outputs,

$$\mathcal{P}(y_i|\mathbf{x}_1, \ldots, \mathbf{x}_m) = \sum_{j=1}^{m} \kappa_j \mathcal{P}_j(y_i|\mathbf{x}_j), \tag{6}$$

where $\mathbf{x}_j$ denotes the representation of an image in different feature spaces, $\mathcal{P}_j$ denotes probability of a class y_i given a feature vector $\mathbf{x}_j$, $\kappa_j : j \in [1, m]$ denotes the weights and $m = 2$. Note that instead of using a single set of weights, we use multiple configurations of weights, each with a small component of random noise, to obtain several contrasting opinions. After unifying beliefs based on contrasting opinions, the Bayesian rule is used to combine them in the subsequent stage. To try a number of weighting options (r configurations of weights κ) to generate contrasting opinions $\mathbf{o} = [\mathcal{P}(y_i|\mathbf{x})\kappa^{\mathrm{T}}]_r$, 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 given y_i, 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,

$$\mathcal{P}(y_i|\mathbf{o}_1, \ldots, \mathbf{o}_r) = \rho \frac{\mathcal{P}(y_i|\mathbf{o}_1)\ldots\mathcal{P}(y_i|\mathbf{o}_r)}{\mathcal{P}(y_i)^{r-1}}. \tag{7}$$

Here, $\mathcal{P}(y_i)$ is the prior and ρ is a constant which depends on the data and is given by $\rho = \frac{\mathcal{P}(\mathbf{o}_1)\ldots\mathcal{P}(\mathbf{o}_r)}{\mathcal{P}(\mathbf{o}_1,\ldots,\mathbf{o}_r)}$ (Edwards et al. 2007). The appearance energy is therefore defined by:

$$\phi_i(y_i, \mathbf{x}; \mathbf{w}_u^{app}) = \mathbf{w}_u^{app} \log \mathcal{P}(y_i|\mathbf{o}_1, \ldots, \mathbf{o}_r), \tag{8}$$

where, $\mathbf{w}_u^{app}$ is the parameter of the appearance energy. This energy is dependent on the output of two Randomized Decision Forest (RDF) classifiers which give the posterior probabilities $\mathcal{P}(y_i|\mathbf{x}_i)$. These classifiers capture the important characteristics of an image using a set of features, which encode information about the shape, the texture, the context and the geometry. The appearance energy proves to be the most important one for the scene labeling problem as shown in the results section (Sect. 6).

[1] In this work we set $r = 3$ and $\boldsymbol{\kappa}$ 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 Sect. 6.2).

Features for Local Appearance Energy: The local appearance energy is modeled in a discriminative fashion using a trained classifier (RDF in our case). We extract features densely at each point and then aggregate them at the super-pixel level using a simple averaging operation. It must be noted that the feature aggregation is done on the super-pixels in order to reduce the computational load and to ensure that similar pixels are modeled by a unified representation in the feature space. The super-pixels are obtained using the Felzenszwalb graph-based segmentation method (Felzenszwalb and Huttenlocher 2004). We use a scale of ten with a minimum region size of 200 pixels. This parameter selection is based on prior tests which were performed on a validation set (Sect. 6.2).

A rich feature set is extracted which includes local binary patterns (LBP) (Ojala et al. 2002), texton features (Shotton et al. 2009), SPIN images (Johnson and Hebert 1999), scale invariant feature transform (SIFT) (Lowe 2004), color SIFT, depth SIFT and histogram of gradients (HOG) (Dalal and Triggs 2005). These low-level features help in differentiating between the distinct classes commonly found in indoor scenes. LBP is a strong texture classification feature which captures the relation between a pixel and its neighbors in the form of an encoded binary word. LBP is extracted from a 10×10 region around a pixel and the normalized histogram is converted to a 59 dimensional vector. For the calculation of texton features, we first convolve the image with a filter bank of even and odd symmetric oriented energy kernels at four different scales (0.5, 0.6, 0.72, 0.86) with four different orientations (0, 0.79, 1.57 and 2.35 radians). The Gaussian second derivative and the Hilbert transform of the Gaussian second derivative are used as the even and odd symmetric filters respectively. This creates a filter-bank consisting of a total of 32 filters of varying sizes (11×11, 13×13, 15×15 and 17×17). Next, image pixels are grouped into $k = 32$ textons by clustering the filter-bank responses into 32 groups. This gives a 96 dimensional vector which is composed of filter responses.

SPIN images are extracted by considering a radius of $r = 8$ around a pixel with eight bins. This gives us a 64 dimensional vector. SIFT descriptors of length 128 are extracted on a 40×40 patch both for the case of simple SIFT and depth SIFT. We followed the same procedure as detailed in Silberman and Fergus (2011) to calculate the depth SIFT. To incorporate the color information into the local SIFT, we use the opponent angle, hue and spherical angle method of Van De Weijer and Schmid (2006). The parameters are set in a way similar to Van De Weijer and Schmid (2006) and this gives a 111 dimensional vector. We extract a 36 dimension HOG feature vector on a 4×4 region quantized into nine orientation bins. Trilinear interpolation is used to place each gradient in the appropriate spatial and orientation bin.

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 the genetic search algorithm from the Weka attribute selector tool (Hall et al. 2009) to find the most useful set of 256 features on the validation set (Sect. 6.2). This feature subset selection effectively reduces the classifier training time to one third of what it was originally. Also, the performance of the lower-dimensional feature vector is comparable to that of the original feature set, e.g., on the validation set from NYU v1, we noted only 0.03 % decrease in accuracy.

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 characterise its extent and position. Since these measures may vary significantly and a relative measure is needed, we normalize each value with respect to the largest instance in the scene. Color histograms in the HSV and CIE LAB color spaces are also included. The responses to various filters are calculated and aggregated at the planar level (in the same manner as *textons*). The RDF classifier is trained using these features and used to predict the posterior on planar regions.

Unary Classifiers: Separate RDF classifiers are trained, one for the extracted local features on super-pixels and the other for the planar regions. The RDF classifier creates an ensemble of trees during the training phase and combines their outputs for predictions (Breiman 2001). For our purpose, we directly obtain the class probabilities $\mathcal{P}(y_i|\mathbf{x})$ by averaging the decisions over all tress. We use the RDF classifiers to predict the unary cost (Eq. 8) in the CRF model (Fig. 2) because of their efficiency and inherent multi-class classification ability. We trained both RDFs with 100 trees and 500 randomly-sampled variables as candidates at each split. This configuration was set empirically taking into account the trade-off between reasonable performance and efficient training of the RDFs.

3.1.2 Proposed Location Energy

The unary location prior (second term) in Eq. 5 models the class label distribution based on the location of the pixels in an image. This energy is useful during the segmentation process since it encodes the probability of the spatial presence of a class. The location energy is defined for each class and every pixel location in the image plane:

$$\phi(y_i, i; \mathbf{w}_u^{loc}) = \mathbf{w}_u^{loc} \log \mathcal{F}_{loc}(y_i, i), \tag{9}$$

where, $\mathbf{w}_u^{loc}$ parameterises the location energy and the function $\mathcal{F}_{loc}(y_i, i)$ is dependent on both the location and the geometry of a pixel (Fig. 1).

Our formulation of $\mathcal{F}_{loc}(y_i, i)$ 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 highly unlikely to have a *bed* or *floor* at some locations 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 observations made from the scene. First, we average the class occurrences over the ground truth of the training set for each class (y_i) (Silberman and Fergus 2011; Shotton et al. 2009). This can be represented by the ratio of the class occurrences at the i^{th} location to the total number of occurrences:

$$\mathcal{F}_{loc}(y_i, i) = \frac{N_{\{y_i, i\}} + \alpha}{N_i + \alpha}, \tag{10}$$

where α is a constant which corresponds to the weak Dirichlet prior on the location energy (Shotton et al. 2009). Next, we incorporate the geometric information into the location prior. For this, we extract the planar regions, which occur in an indoor scene, and divide them into two distinct geometrical classes: *horizontal* and *vertical* regions. Since the Kinect sensor gives the pitch and roll for each image, the intensity and depth images in the NYU-Depth dataset are rotated appropriately to remove any affine transformations. This positions the horizon (estimated using the accelerometer) horizontally at the center of each image. We use this horizon to split the horizontal geometric class into two subclasses, the *'above-horizon'* and *'below-horizon'* regions. 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 decrease its value by a constant factor in the 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 geometrical context. It must be noted that this elimination procedure is only carried out for planar classes e,g., roof, floor, bed and blinds. After that, the location prior is smoothed using a Gaussian filter and the actual prior distribution is normalized in such a way that a uniform distribution across different classes is obtained. The prior distribution is normalized to give $\sum_i \mathcal{F}_{loc}(y_i, i) = 1/L$, where L is the total number of classes. Examples of the resulting location priors are shown in Fig. 4.

3.2 Pairwise Energies

The pairwise energy in Eq. 2 is defined on the edges $\mathcal{E}$ (Fig. 2). This energy is defined in terms of an edge-sensitive Potts model (Boykov et al. 2001),

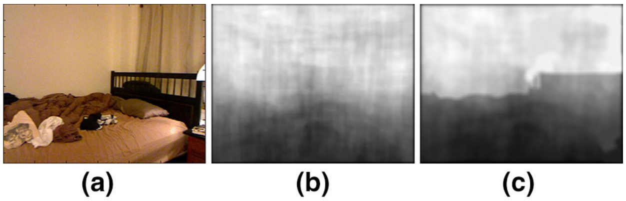

Fig. 4 Learning Location Prior using Geometrical Context: **a** original image. **b** The normal location prior for *wall* is shown. **c** It shows how the prior (**b**) is combined with the planar information to channelize the general location information of a class by considering the scene geometry. Note that white color in **b** and **c** shows high probability

$$\psi_p(y_{ij}, \mathbf{x}; \mathbf{w}_p) = \mathbf{w}_p^{\mathrm{T}} \phi_{p_1}(y_i, y_j) \phi_{p_2}(\mathbf{x}). \tag{11}$$

The first function (ϕ_{p_1}) is a *class transition energy* and the second one (ϕ_{p_2}) is the *spatial discontinuation energy*. These functions are defined in the following subsections (Sects. 3.2.1 and 3.2.2 respectively).

3.2.1 Class Transition colorblueEnergy

The class transition energy in Eq. 11 is a simple zero-one indicator function which enforces a consistent labeling. The function is defined as:

$$\phi_{p_1}(y_i, y_j) = a\mathbf{1}_{y_i \neq y_j} = \begin{cases} 0 & \text{if } y_i = y_j \\ a & \text{otherwise} \end{cases}$$

For this work we used $a = 10$. This parameter selection was based on the validation set (Sect. 6.2).

3.2.2 Proposed Spatial Discontinuation Energy

The spatial discontinuation energy in Eq. 11 encourages label transitions at natural boundaries in the image (Shotton et al. 2009; Rother et al. 2004). It is defined as a combination of edges from the intensity image, depth image and the super-pixel edges extracted using Mean-shift (Fukunaga and Hostetler 1975) and Felzenswalb (Felzenszwalb and Huttenlocher 2004) segmentation: $\phi_{p_2}(\mathbf{x}) = \mathbf{w}_{p2}^{\mathrm{T}} \phi_{edges}(\mathbf{x})$. Weights assigned to each edge-based energy are learned using a quadratic program (see Sect. 4.1). In simple terms, edges which match with the manual annotations to a large extent contribute more in the energy ϕ_{p_2}. The edge-based energy is given by:

$$\phi_{edges}(\mathbf{x}) = [\beta_x \exp\left(-\frac{\sigma_{ij}}{\langle \sigma_{ij} \rangle}\right), \beta_d \exp\left(-\frac{\sigma_{ij}^d}{\langle \sigma_{ij}^d \rangle}\right),$$
$$\beta_{\text{sp-fw}} \mathcal{F}_{\text{sp-fw}}(\mathbf{x}), \beta_{\text{sp-ms}} \mathcal{F}_{\text{sp-ms}}(\mathbf{x}), \alpha]^{\mathrm{T}}, \tag{12}$$

where, $\sigma_{ij} = \|x_i - x_j\|^2$, $\sigma^d_{ij} = \|x^d_i - x^d_j\|^2$ and $\langle . \rangle$ denotes the average contrast in an image. x_i and x^d_i 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 (Fukunaga and Hostetler 1975) or Felzenswalb (Felzenszwalb and Huttenlocher 2004) super-pixels respectively. The output is a binary image containing ones at the super-pixel boundaries. The inclusion of a constant $\alpha = 1$ allows a bias to be learned to remove small isolated parts during the segmentation process. For our case, we set $\beta_x = \beta_d = 150$ and $\beta_{\text{sp-ms}} = \beta_{\text{sp-fw}} = 5$ based on the validation set (see Sect. 6.2).

3.3 Proposed Higher-Order Energies

A useful strategy to enhance the representational power of a CRF model is to introduce high-order energies (Eq. 1). These energies are dependent on a relatively large number of dimensions of the output labeling vector $\mathbf{y}$ and therefore incorporate long-range interactions (Fig. 2). HOEs try to eliminate inconsistent variables in a clique. On the other hand, these energies try to encourage all the variables in a clique to take the dominant label. The robust P^n model (Kohli et al. 2009) poses this encouragement in a soft manner while the P^n Potts model (Kohli et al. 2007) presents this requirement in a hard fashion. In the robust P^n model 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. We define our proposed HOE which works in a similar manner as the robust HOE (Kohli et al. 2009):

$$\psi_c(y_c, \mathbf{x}; \mathbf{w}_c) = \mathbf{w}_c \min_{\ell \in \mathcal{L}} \mathcal{F}_c(\tau_c), \tag{13}$$

where, $\mathcal{F}_c(.)$ is a function which takes the number of inconsistent pixels $\tau_c = \#c - n_\ell(y_c)$ as its argument. Here, n_ℓ is a function which computes the number of pixels in clique c taking the label ℓ. The non-decreasing concave function $\mathcal{F}_c$ is defined as: $\mathcal{F}_c(\tau_c) = \lambda_{max} - (\lambda_{max} - \lambda_\ell)\exp(-\eta\tau_c)$, where $\eta = \eta_0/Q_\ell$ and $\eta_0 = 5$ (Fig. 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. The higher-order cliques are formed using the depth-based segmentation method (Sect. 5). Details about the disintegration of the HOE (Eq. 13) are given in Appendix to describe how the graph cuts algorithm can be applied.

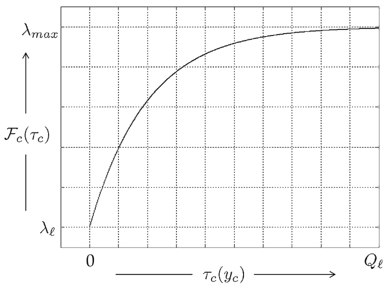

Fig. 5 Robust Higher-Order Energy: When the number of inconsistent nodes in a clique increases, the penalty term defined over the clique increases in a logarithmic fashion

4 Structured Learning and Inference

The task of indoor scene labeling involves making joint predictions over many complex yet correlated and structured outputs. The CRF model defined in the previous section (Sect. 3) explicitly models the correlations over the output space and performs approximate inference at test time. However, the CRF model contains a number of energies, parametrized by weights which we learn using a S-SVM formulation. The learning procedure is outlined as follows.

4.1 Learning Parameters

Unary, pairwise and high order terms (Eq. 2 and Figs. 1, 2) in the CRF model introduce many parameters which need a more principled tuning procedure rather than simple hand-picked values, cross validation learning or a piecewise training mechanism. In this work, we use a structured large-margin learning method (S-SVM) to efficiently adjust the probabilistic model parameters. Instead of using an n-slack formulation of the cost function, we use a single slack formulation, which results in more efficient learning (Joachims et al. 2009). Given N training images, the training set can be represented in the form of ordered pairs of image data $\mathbf{x}$ and labelings $\mathbf{y}$: $\mathcal{T} = \{(\mathbf{x}_n, \mathbf{y}_n), n \in [1, \ldots, N]\}$. If $\xi \in \mathbb{R}_+$ is a single slack variable, the following margin re-scaled cost function is solved to compute the parameter vector $\mathbf{w}^*$:

$$(\mathbf{w}^*, \xi^*) = \operatorname*{argmin}_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \tag{14}$$

subject to;

$$\begin{aligned} &\frac{1}{N}\sum_{n=1}^{N}[E(\mathbf{y}, \mathbf{x}^n; \mathbf{w}) - E(\mathbf{y}^n, \mathbf{x}^n; \mathbf{w})] \geq \frac{1}{N}\sum_{n=1}^{N}\Delta(\mathbf{y}, \mathbf{y}^n) - \xi \\ &\forall n \in [1..N], \forall \mathbf{y} \in \mathcal{L} : \mathbf{y} \neq \mathbf{y}^n, C > 0, \\ &w_i \geq 0 : \forall w_i \in \{\mathbf{w}\}_{\setminus w_u}, \end{aligned} \tag{15}$$

where, C is the regularization constant, $\Delta(\mathbf{y}, \mathbf{y}^n)$ is the Hamming loss function and the parameter vector $\mathbf{w}$ consists of the appearance energy weight ($\mathbf{w}_u^{app}$), the location energy weight ($\mathbf{w}_u^{loc}$), the pairwise energy weight ($\mathbf{w}_p$) and the weight for HOE ($\mathbf{w}_c$). Due to the large number of constraints in Eq. 15, a cutting plane algorithm (Joachims et al. (2009), Algorithm 4) is used for training which only considers the most violated constraints to solve our optimization problem. It can be proved that the algorithm converges after O$(1/\epsilon)$ steps with the guarantee that the objective value (once the final solution is reached) differs by at most ϵ from the global minimum (Tsochantaridis et al. 2004). The two major steps in this algorithm are the quadratic optimization step, which is solvable by off-the-shelf convex optimization problem solvers and the loss-augmented prediction step, which can be solved by graph cuts.

Once suitable parameters for the CRF are learned, the parameters for the edge-based energies are learned which results in a balanced representation of each edge in the pairwise energy. In our approach, instead of a simple contrast-based energy, we define a weighted combination of various possible edge-based energies (such as based on depth edges, contrast-based edges, super-pixels edges) to accommodate information from all these sources (see Sect. 3.2.2 and Eq. 12). We start with a heuristic-based initialization and iterate over the training samples to learn a more balanced representation between the different edge-based energies. The weights for edges are restrained to be non-negative so that the energy remains sub-modular. This condition is necessary because the graph cuts based exact inference methods can be applied only to sub-modular energy minimization problems.

We use structured learning to learn weights for the spatial discontinuation energy (Sect. 3.2.2). The corresponding quadratic program is given as follows:

$$\underset{\|\mathbf{w}_{p2}\|=1}{\text{argmax}}\ \gamma \tag{16}$$

s.t.; $\{E_{\text{con}}, E_{\text{dep}}, E_{\text{fel-sp}}, E_{\text{ms-sp}}\} - E_{\text{grd}} \geq \gamma, \{\mathbf{w}_{p2}\} \geq 0$,
where, E_{grd} is the energy when the spatial discontinuation energy is based on the manually identified edges from the training images. Energies for the case when the spatial discontinuation energy 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. 16 is optimized in a similar way to that described in Joachims et al. (2009), Algorithm 4). After learning, it turns out that the contrast and depth-based edge energies are more reliable and therefore play a dominant role in the spatial discontinuation energy.

4.2 Inference in CRF

Once the CRF energies have been learned along with their parameters, the next step is to find the most probable labeling. As discussed earlier in Sect. 3, this turns out to be an energy minimization problem (Eq. 4). Since our energy function is sub-modular, this energy minimization problem can be solved via the expansion move algorithms (alpha-expansion or alpha-beta swap graph cuts algorithm) of Boykov and Funka-Lea (2006). The main idea is to decompose the energy minimization problem into a series of binary minimization problems which can themselves be solved efficiently. The algorithm starts with an arbitrary initial labeling and at each step the move is only made if it results in an overall minimization of the cost function (Boykov et al. 2001; Boykov and Funka-Lea 2006).

5 Planar Surface Detection

Indoor environments are predominantly composed of structures which can be decomposed into planar regions, such as walls, ceilings, cupboards and blinds. These flat surfaces are easier to manufacture and thus appear frequently in man-made environments (Sect. 6.2.2). 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 the depth images produced by a Kinect contain many missing values e.g., along the outer boundaries of an image or when the scene contains a black or a specular surface (Fig. 6). Traditional plane detection algorithms (e.g. Silberman et al. (2012); Rabbani et al. (2006)) 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 values (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 (see Sect. 6.2.1).

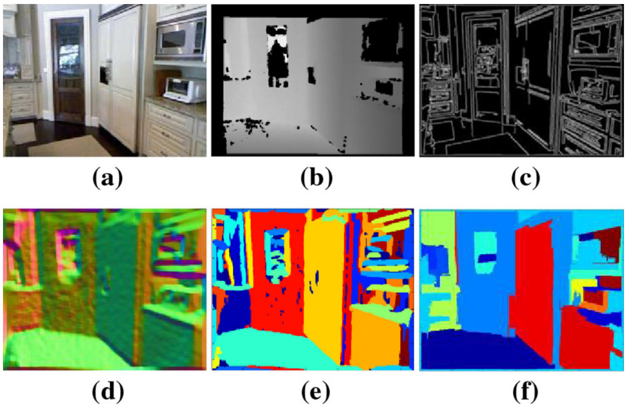

Fig. 6 An illustrative example showing the results of the planar surface detection algorithm. An original image (**a**) and its depth map (**b**) are used as inputs to the algorithm which uses appearance (**c**) and depth-based cues (**d**) to provide an initial (**e**) and a final segmentation map (**f**)

Our method[2] first aligns the 3D points with the principal directions of the room. Next, surface normals are computed at each point. Contiguous points in space are then clustered by a region growing algorithm (Algorithm 1) 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. 7). The basic idea is to make use of 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) (Von Gioi et al. 2010) is used to extract the major line segments. These line segments are grouped according to their 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. However, we found empirically that the use of any simple edge detection method (e.g., Canny edge detector) in our algorithm gives nearly identical 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 approximated by planes rather than missed out during the region-growing process.

Once the regions have been grown to their full extent, small regions are dropped, and only 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. Least-square plane fitting is a non-linear problem, but it 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 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).

Algorithm 1 Region Growing Algorithm for Depth-Based Segmentation

```
Input: Point cloud = {P}, Depth map = {D}, RGB image = {I}, Edge matching threshold $e_{th}$, Normalized boundary matching threshold $b_{th}$
Output: Labeled planar regions = {R}
1: Calculate point normals: $\{\mathbf{N}\} \leftarrow \mathcal{F}_{normal}(\mathbf{D})$
2: Remove inconsistencies by low-pass filtering: $\{\mathbf{N}_{\text{sm}}\} \leftarrow \mathbf{N} * k_{\text{sm}}$   // $k_{\text{sm}}$ is the smoothing kernel
3: Cluster 3D points with similar normal orientations: $\{\mathbf{N}_{\text{clu}}\} \leftarrow \mathcal{F}_{k-means}(\mathbf{N}_{\text{sm}})$
4: Initialize: $\mathbf{R} \leftarrow \mathbf{N}_{\text{clu}}$
5: Line segment detector: $\{\mathbf{L}\} \leftarrow \mathcal{F}_{\text{LSD}}(\mathbf{I})$
6: Diffused line map: $\{\mathbf{L}_{\text{sm}}\} \leftarrow \mathbf{L} * k'_{\text{sm}}$
7: Identify planar regions with missing depth values: $\{\mathbf{M}\} \leftarrow \mathcal{F}_{holes}(\mathbf{N}_{\text{clu}}, \mathbf{D})$
8: Find adjacency links for each cluster in $\mathbf{N}_{\text{clu}}$: $\mathbf{A}_{\text{clu}}$
9: Identify all unique neighbors of clusters in $\mathbf{M}$: $\mathbf{U}_{\text{nb}}$
10: From $\mathbf{U}_{\text{nb}}$, separate correct and faulty clusters into $\mathbf{N}_{\text{cor}}$ and $\mathbf{N}_{\text{inc}}$ respectively
11: Initialize available cluster list: $\mathbf{L}_{avl} \leftarrow \mathbf{N}_{\text{cor}}$
12: Initialize label propagation list: $\mathbf{L}_{prp} \leftarrow \emptyset$
13: while list $\mathbf{L}_{avl}$ is not empty do
14:     Randomly draw a cluster from available $\mathbf{N}_{\text{cor}}$: $\mathbf{r}_{idx}$
15:     Identify $\mathbf{r}_{idx}$ neighbors ($\mathbf{N}_{r-idx}$) with faulty depth values using $\mathbf{A}_{\text{clu}}$ and $\mathbf{M}$
16:     for each neighbor $\mathbf{n}_{r-idx}$ in $\mathbf{N}_{r-idx}$ do
17:         Find mutual boundary ($\mathbf{b}_m$) of $\mathbf{r}_{idx}$ and $\mathbf{n}_{r-idx}$
18:         Calculate edge strength at $\mathbf{b}_m$ using $\mathbf{L}_{\text{sm}}$: $e_{str}$
19:         Calculate normalized boundary matching cost: $b_{str} = \mathbf{b}_m /$ Area of $\mathbf{n}_{r-idx}$
20:         if $e_{str} < e_{th} \wedge b_{str} > b_{th}$ then
21:             $\mathbf{n}_{r-idx} \xrightarrow{\text{add}} \mathbf{N}_{\text{cor}}$, $\mathbf{n}_{r-idx} \xrightarrow{\text{add}} \mathbf{L}_{\text{avl}}$
22:             $\mathbf{r}_{idx} \xrightarrow{\text{rem}} \mathbf{L}_{\text{avl}}$, $\mathbf{n}_{r-idx} \xrightarrow{\text{rem}} \mathbf{N}_{\text{inc}}$
23:             Update $\mathbf{L}_{prp}$ with $\mathbf{r}_{idx}$ and $\mathbf{n}_{r-idx}$. If $\mathbf{n}_{r-idx}$ was previously replaced, use the updated value.
24:     $\mathbf{r}_{idx} \xrightarrow{\text{rem}} \mathbf{L}_{\text{avl}}$
25: for any leftover clusters in $\mathbf{N}_{\text{inc}}$ do
26:     Randomly draw a cluster from available $\mathbf{N}_{\text{inc}}$: $\mathbf{r}'_{idx}$
27:     Execute similar steps (from line 15 to 24) for $\mathbf{r}'_{idx}$
28: Update $\mathbf{R}$ according to $\mathbf{L}_{\text{prp}}$
29: return {R}
```

We show a qualitative comparison of our approach with other plane detection techniques in Fig. 7. Note that our approach provides a depth-based segmentation and then fits planes to the approximate geometry of the region (3^{rd} row, Fig. 7). This makes it possible to identify better planar region candidates compared to Silberman et al. (2012) (2^{nd} row, Fig. 7). We show a quantitative performance and efficiency comparison in Table 1. For the performance evaluation, we report the achieved accuracy when a valid planar region was identified for a strictly planar semantic class (EPC, Table 1). To quantify the validity of a detected planar region, we check its alignment with the three dominant and perpendicular room directions. We also report the accuracy with which a valid

[2] Plane detection code is available at author's webpage: http://www.csse.uwa.edu.au/~salman.

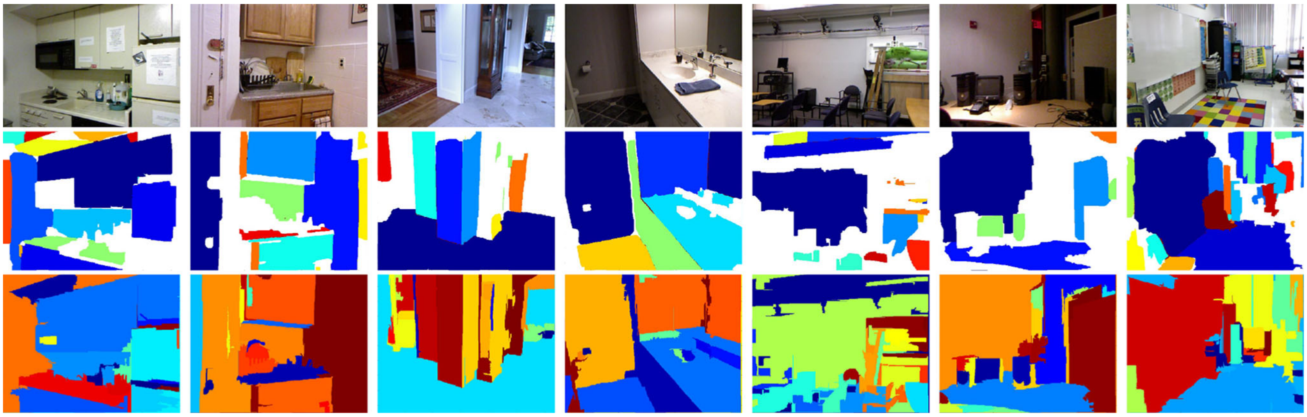

Fig. 7 Comparison of our algorithm (*last* row) with Silberman et al. (2012) (*middle* row) is shown. Note that the *white* color in middle row shows *non-planar* regions. The *last* row shows detected planes averaged over super-pixels. Results show that our algorithm is more accurate especially near the outer boundaries of the scene. (*Best viewed in color*)

Table 1 Comparison of plane detection results on the NYU-Depth v2 dataset

Performance Evaluation		
Method	EPC Acc.	E+NPC Acc.
Silberman et al. (2012)	0.69 ± 0.09	0.67 ± 0.10
Rabbani et al. (2006)	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 et al. (2012)	Rabbani et al. (2006)	This paper
41 s	73 s	3.1 s

We report detection accuracies for ‘exactly planar classes’ (EPC) and ‘exact and nearly planar classes’ (E+NPC). Efficiency of the proposed method is also compared with related approaches
Performances reported in bold denote the best performance (in each column)

planar region was identified for the exactly (e.g., walls, ceilings) and nearly planar (e.g., blinds, beds) semantic classes (E+NPC, Table 1). The results demonstrate that our algorithm is superior to other region growing algorithms (e.g., Rabbani et al. (2006)) which are suitable for the segmentation of dense point clouds and fail to deal with erroneous depth measurements from the Kinect sensor (Table 1).

6 Experiments and Analysis

6.1 Datasets

We evaluated our framework on the NYU-Depth datasets (v1 and v2) and the SUN3D dataset. All these are recent RGBD datasets for indoor scenes acquired using the Microsoft Kinect structured light sensor. The NYU-Depth dataset is the only one of its kind and comes with manual annotations acquired via Amazon Mechanical Turk. The dataset comes in two releases. The first version (v1) of NYU-Depth (Silberman and Fergus 2011) consists of 64 different indoor scenes categorized into seven major scene types and contains 2284 labeled frames. The second version (v2) of NYU-Depth (Silberman et al. 2012) consists of 464 different indoor scenes classified into 26 major scene types and contains 1449 labeled frames. SUN3D is a large-scale indoor RGBD video dataset (Xiao et al. 2013); however, it is still under development and only a small portion has been labeled. We extracted labeled key-frames from the SUN3D database which amounted to 83 images. We evaluated our method on the labeled portions of the NYU v1, v2 and SUN3D datasets.

6.2 Results

In the NYU-Depth v1 dataset, around 1400 different object classes are present in all indoor scenes. Since not all object classes have a sufficient representation, we follow the procedure in Silberman and Fergus (2011) to cluster the existing annotations into the 13 most frequently occurring classes. This clustering is performed using the Wordnet Natural Language Toolkit (NLTK). In the NYU-Depth v2 dataset, around 900 different object classes are present overall. We used a similar procedure to cluster existing annotations into the 22 most frequently occurring classes. Moreover, we report results on 40 classes to show how our performance compares when the number of semantic classes is increased. For the SUN3D dataset, 32 classes are present in the labeled images we acquired. We clustered them into 13 major classes using Wordnet. In all three datasets, a supplementary class labeled ‘*other*’ is also included to model rarely-occurring objects. In our evaluations, we exclude all unlabeled regions. For all the three datasets, roughly a train/test split of 60/40 % was used. A relatively small validation set consisting of 50 random images was extracted from each dataset (except for SUN3D where we used the parameters of NYU-Depth v1). This validation set was used with the genetic search algorithm (Sect. 3.1.1) for the selection of useful features and for the choice of the initial estimates of the parameters which give the best performance. Afterwards, these parameters were optimized during the learning process as described in Sect. 4.1.

We use two popular evaluation metrics to assess our results, '*global accuracy*' and '*class accuracy*' (see Table 2). Global accuracy measures the average number of superpixels 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 along the diagonal of the confusion matrix. We extensively evaluated our approach on both versions of the NYU-Depth dataset and on the SUN3D dataset. Our experimental results are reported in Tables 2, 3, 4 and 5. Comparisons with state-of-the-art techniques are reported in Tables 6, 7, 8 , 9 and 10. Sample labelings for NYU-Depth v1 and v2 and SUN3D are presented in Figs. 8, 9 and 11 respectively. 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 Figs. 8 and 9).

6.2.1 Ablation Study

We report our results in terms of average pixel and class accuracies in Table 2. The first row shows the performance when a simple unary energy defined on pixels using an ensemble of features is used. We achieve pixel and class accuracies of 52.8 and 53.4 % respectively on NYU-Depth v1. The corresponding accuracies for NYU-Depth v2 and SUN3D are 44.4, 39.2 and 41.9, 40.0 % respectively. Starting from this baseline, we were able to obtain significant improvements. Upon the introduction of the planar appearance model, the pixel and class accuracies increased by 10.5 and 9.3 % from their previous values for NYU-Depth v1 (row 3, Table 2). Similarly for NYU-Depth v2, an increase of 8.1 and 3.2 % is noted for pixel and class accuracies respectively. Finally for the SUN3D database, we achieve an increase of 6.4 and 2.6 % in pixel and class accuracies respectively. Note that a sim-

Table 2 Results on the NYU-Depth v1, v2 and the SUN3D Datasets: we report the results of our proposed framework when only the unary energy was used (top 3 rows) and report the improvements observed when more sophisticated priors and HOEs (last row) were added

Variants of our method	NYU-Depth v1		NYU-Depth v2		SUN3D	
	Global accuracy (%)	Class acc. (%)	Global accuracy (%)	Class Acc. (%)	Global accuracy (%)	Class acc. (%)
Feature Ensemble (FE)	52.8 ± 13.3	53.4	44.4 ± 15.8	39.2	41.9 ± 11.1	40.0
FE + PAM (single opinion)	60.9 ± 13.3	60.2	51.1 ± 15.6	41.5	47.6 ± 11.3	41.8
FE + Planar Appearance Model (PAM)	63.3 ± 13.1	62.7	52.5 ± 15.5	42.4	48.3 ± 11.5	42.6
FE + PAM + Location Prior (2D)	65.2 ± 13.4	63.5	53.6 ± 15.6	42.8	48.9 ± 11.7	42.8
FE + PAM + Planar Location Prior (PLP)	68.6 ± 13.8	65.0	55.3 ± 15.8	43.1	51.5 ± 11.9	43.3
FE + PAM + PLP + CRF	70.5 ± 13.8	66.5	58.0 ± 16.0	44.9	53.7 ± 12.1	44.4
FE + PAM + PLP + CRF (HOE)	**70.6 ± 13.8**	**66.5**	**58.3 ± 15.9**	**45.1**	**54.2 ± 12.2**	**44.7**

Accuracies are reported for 13, 22 and 13 class semantic labelings for NYU v1, v2 and SUN3D datasets, respectively. The best performance is achieved by combining unary, pairwise and HOEs in the CRF framework

Performances reported in bold denote the best performance (in each column)

Table 3 Class-wise Accuracies on NYU-Depth v1: mean class and global accuracies are also reported

Class	Bed	Blind	Bookshelf	Cabinet	Ceiling	Floor	Picture	Sofa	Table	Television	Wall	Window	Other	Unlabeled	Mean class accuracy	Mean pixel accuracy
Class freq.	1.3	3.7	13.4	7.7	3.7	11.3	4.7	2.5	4.6	0.6	26	2.1	0.24	18.1	-	-
This paper	66.8	67.7	47.5	72.6	79.2	67.8	53.4	75.1	69.3	78.6	86.2	62.0	38.1	-	**66.5**	**70.6**

Our proposed framework performs very well on the planar classes (e.g., '*wall*', '*television*', '*ceiling*')

Performances reported in bold denote the best performance (in each column)

Table 4 Class-wise accuracies on NYU-Depth v2 (22 classes): mean class and global accuracies are also reported

Class	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
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	-	-
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**

Our proposed framework performs very well on the planar classes (e.g., ‘*wall*’, ‘*door*’, ‘*floor*’)
Performances reported in bold denote the best performance (in each column)

Table 5 Class-wise Accuracies on the NYU-Depth v2 (40 classes): Mean class and global accuracies are also reported. Our proposed framework performs very well on the planar classes (e.g., ‘*wall*’, ‘*ceiling*’, ‘*whiteboard*’)

Class	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Bookshelf	Picture	Counter	Blinds	Desk	Shelves	Curtain	Dresser	Pillow	Mirror	Floor mat	Clothes	Ceiling
Class Freq.	21.4	9.1	6.2	3.8	3.3	2.7	2.1	2.2	2.1	1.9	2.1	1.4	1.7	1.1	1.0	1.1	0.9	0.8	1.0	0.7	0.7	1.4
This paper	65.7	62.5	40.1	32.1	44.5	50.8	43.5	51.6	49.2	36.3	41.4	39.2	55.8	48.0	45.2	53.1	55.3	50.5	46.1	54.1	35.4	50.6

Class	Books	Refrigerator	Television	Paper	Towel	Shower curtain	Box	Whiteboard	Person	Nightstand	Toilet	Sink	Lamp	Bathtub	Bag	Other structure	Other furniture	Other props	Unlabeled	Mean Class Accuracy	Mean Pixel Accuracy
Class Freq.	0.6	0.6	0.5	0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2	3.8	2.5	2.2	17.4	-	-
This paper	39.1	53.6	50.1	35.4	39.9	41.8	36.3	60.6	35.6	32.5	31.8	22.5	26.3	38.5	37.3	45.7	24.9	29.1	-	**43.9**	**50.7**

Performances reported in bold denote the best performance (in each column)

ple averaging operation on the pixel and planar appearance energies (equivalently an LOP with weights $[\frac{1}{2}, \frac{1}{2}]$) gives less accurate results (row 2, Table 2). The addition of the CRF and the proposed location energy enforce a better label consistency which results in an improvement of 7.2 and 3.8 % for NYU-Depth v1, 5.5 and 2.5 % for NYU-Depth v2, 5.4 and 2.1 % for SUN3D datasets. The introduction of HOEs gives a slight boost in accuracy. This is logical since the introduction of cardinality-based HOEs improves segmentation accuracies for porous and fine structures such as trees and cat fur, respectively. The classes which are considered in this work usually have solid structures with definite and well-defined boundaries. However, when we consider the segmentation performance around the boundary regions, the HOEs give a significant increase in accuracy (Fig. 10).

6.2.2 Comparisons

For NYU-Depth v1, we compare our framework with Silberman and Fergus (2011) (Table 6). With the same set of classes used in Silberman and Fergus (2011), we achieved a 13.2 % improvement in terms of average class accuracy. We also report the average global accuracy which gives a better absolute measurement of performance. The class-wise accuracies for NYU-Depth v1 are shown in Table 3 and the complete confusion matrix is presented in Fig. 12. It can be seen that we perform really well on planar classes such as *wall*, *ceiling*, *blinds* and *table*.

For the case of NYU-Depth v2, we compare our framework with recent multi-scale convolutional network based techniques (Farabet et al. 2013; Couprie et al. 2013). Whereas in Farabet et al. (2013); Couprie et al. (2013) 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 (if we sum up the class occurrences of the *chair* and *sofa* which are reported in Couprie et al. (2013), the combined class frequency supports such a comparison). We compare the *furniture* class in Couprie et al. (2013) with our *cabinet* class based on the details given in Couprie et al. (2013). Overall, we get superior performance compared to Farabet et al. (2013) and Couprie et al. (2013) and also achieve best class accuracies for 19/22 classes.

On the NYU-Depth v2 dataset, Silberman et al. (2012) defined just four semantic classes: *furniture*, *ground*, *structure* and *props*. The choice of these classes was based on the need to infer the support relationships between objects. We evaluate our method on the 4-class segmentation task as well. As shown in Table 8, we achieved the best performance overall. In particular, 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. We also compare our results with Gupta et al. (2013) in terms of the weighted average Jaccard index (WAJI). Our system's performance is lower than that of Gupta et al. (2013), which is based on a very strong but computationally-

Table 6 Comparison of the results on the NYU-Depth v1 Dataset: with the same set of classes used in Silberman and Fergus (2011), we achieve a ∼ 13 % improvement in terms of average class accuracy

Method	NYU-Depth v1		Classes
	Global accuracy (%)	Class accuracy (%)	
Silberman and Fergus (2011)	59.8 ± 11.5	53.7 ± 2.9	13
This paper	**70.6 ± 13.8**	**66.5**	13

Performances reported in bold denote the best performance (in each column)

Table 7 Comparison of results on the NYU-Depth v2 Dataset: With nearly two times the number of classes used in Farabet et al. (2013) and Couprie et al. (2013), we get 6 and 9 % improvement in terms of average class and global accuracies respectively

Method	NYU-Depth v2		Classes
	Global accuracy	Class accuracy	
Farabet et al. (2013)	51.0 ± 15.2	35.8	13
Couprie et al. (2013)	52.4 ± 15.2	36.2	13
This paper	**58.3 ± 15.9**	**45.1**	22

Performances reported in bold denote the best performance (in each column)

Table 8 Comparison of results on the NYU-Depth v2 Dataset (four-class labeling task): our method achieved best performance in terms of average pixel and class accuracies for the 4-class segmentation task

Method	Semantic Classes				Pixel	Class
	Floor	Struct.	Furn.	Prop.	Acc.	Acc.
Silberman et al. (2012)	68	59	**70**	**42**	58.6	59.6
Farabet et al. (2013)	68.1	87.8	51.1	29.9	63	59.2
Couprie et al. (2013)	87.3	86.1	45.3	35.5	64.5	63.5
Cadena and Košecká (2014)	**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**

We also get the best classification performance on structure class
Performances reported in bold denote the best performance (in each column)

Table 9 Comparison of results on the NYU-Depth v2 Dataset (4-class labeling task): our method achieved the second best performance in terms of weighted average Jaccard index (WAJI)

Perf.	SC-Silberman et al. (2012)	LP-Silberman et al. (2012)	Ren et al. (2012)	SVM-Gupta et al. (2013)	This paper
WAJI	56.31	53.4	59.19	**64.81**	62.66

Performances reported in bold denote the best performance (in each column)

Table 10 Comparison of results on the NYU-Depth v2 Dataset (40-class labeling task): our method achieved second best performance in terms of weighted average Jaccard index (WAJI)

Perf.	SC-Silberman et al. (2012)	Ren et al. (2012)	SVM-Gupta et al. (2013)	CNN-Gupta et al. (2014)	This paper
WAJI	38.2	37.6	43.9	**47.0**	42.1

Performances reported in bold denote the best performance (in each column)

expensive contour detection technique called gPb (Arbelaez et al. 2011) (Table 9). Finally, we compare our results on a 40-class semantic labelling task (Table 10). We note that the RGBD version of the R-CNN model proposed in Gupta et al. (2014) performs best. Their approach however, uses external data (Imagenet) for pre-training and uses synthetic

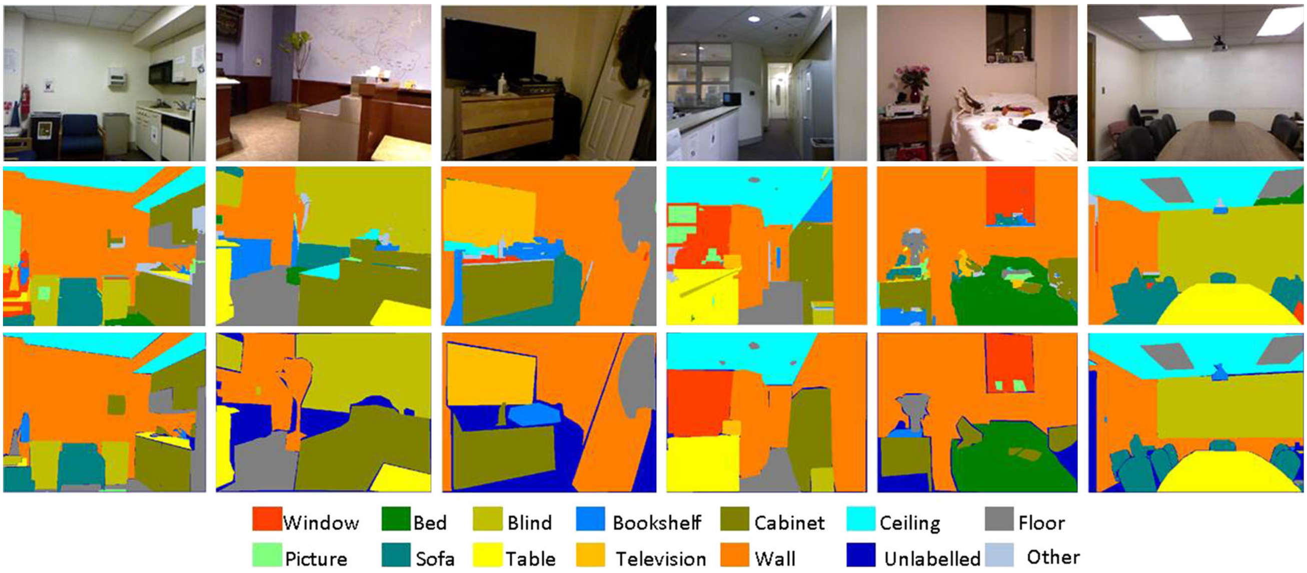

Fig. 8 Examples of the semantic labeling results on the NYU-Depth v1 dataset. The top row shows the intensity images, the bottom row are the ground truths and the middle row are our labeling results. The representative colors are shown in the figure legend at the bottom. Our framework performs well including the case of some unlabeled regions. (*Best viewed in color*)

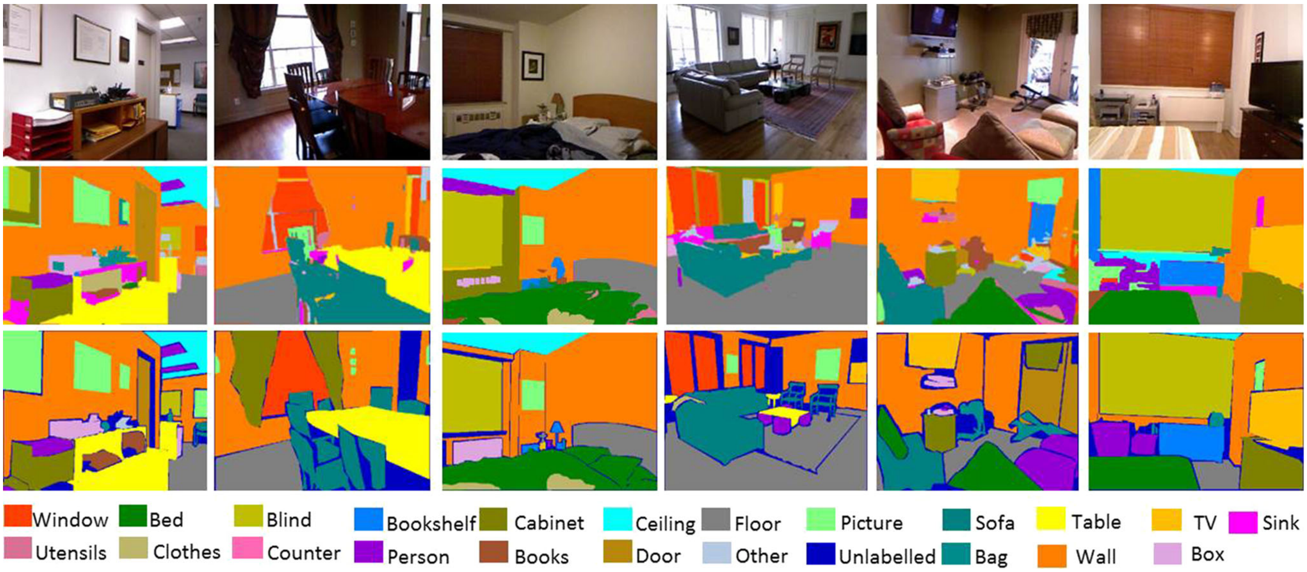

Fig. 9 Examples of semantic labeling results on the NYU-Depth v2 dataset. The *top row* shows the intensity images, the *bottom row* are the ground truths and the *middle row* are our labeling results. The representative colors are shown in the figure legend at the *bottom*. Our framework performs well including the case of some unlabeled regions. (*Best viewed in color*)

3D CAD models from the Internet to generate training data.

One may wonder why the incorporation of geometrical context in the CRF model works and gives such high accuracies? In v1 of the NYU-Depth dataset, there are eight out of 13 classes (cabinet, ceiling, floor, picture, table, wall, bed, blind) which are planar and out of the remaining classes, four (tv, sofa, bookshelf, window) are loosely planar. The planar classes correspond to 77.21 % while the loosely planar classes correspond to 22.79 % of the total labeled data. Second, the floor or wall or other classes may have varying textures across different images. However, with depth information in place, we can determine the correct class of the object. Similarly for 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

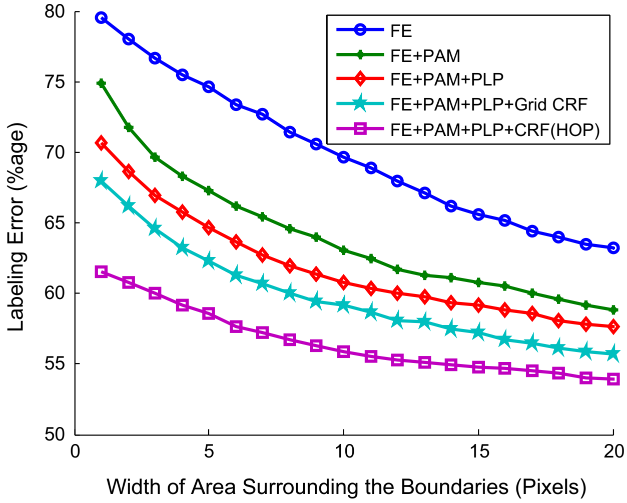

Fig. 10 The error rate decreases as more area surrounding the class boundaries is considered. The introduction of HOE improves the segmentation accuracy around the boundaries

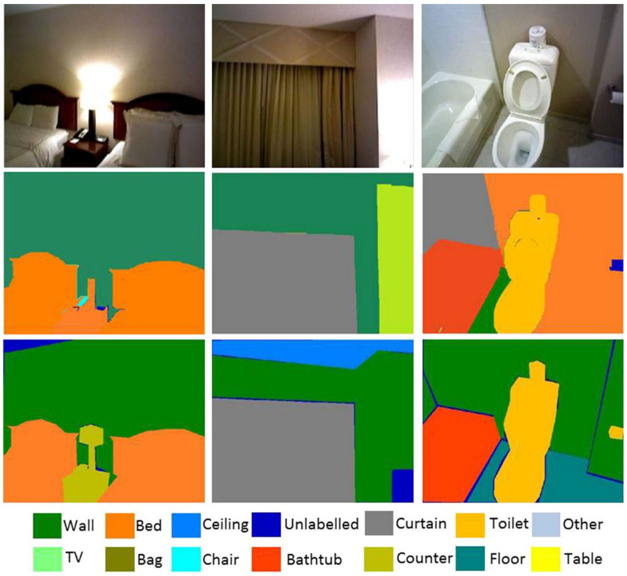

Fig. 11 Examples of the semantic labeling results on the SUN3D dataset. The *top row* shows the intensity images, the *bottom row* are the ground truths and the *middle row* are our labeling results. The representative colors are shown in the figure legend at the *bottom*. (*Best viewed in color*)

the remaining classes 6 are loosely planar (tv, sofa, bookshelf, window, box, sink). The planar classes 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.

6.2.3 Timing Analysis

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 ∼1.6 s for NYU-Depth v1, ∼1.7 s for NYU-Depth v2 and ∼1.4 s for the SUN3D database. For parameter learning on the training set, it took ∼ 17 hrs for NYU-Depth v1, ∼12 h for NYU-Depth v2 and ∼ 45 min for the SUN3D database. The RDF training took ∼4 h, ∼2 h and ∼7 mins on the NYU-Depth v1, v2 and SUN3D databases respectively (Fig. 9).

6.3 Discussion

It may be of interest to know why we used a hierarchical ensemble learning scheme to combine posteriors defined on pixels and planar regions. We prefer to use the proposed scheme because it combines the posteriors on the fly and thus saves a reasonable amount of training time. Alternate ensemble learning methods such as Boosting and Bagging require considerable training data and take much time. It must be noted that we used graph-cuts for making approximate inference during the S-SVM training. This method is not always precisely accurate. Moreover, only a limited set of constraints (the *working set*) from the original infinite number of constraints are used during training. These approximations can sometimes lead to unsatisfactory performance. However, we minimized this behavior by initializing the parameters with values that gave the best performance on the validation set. This heuristic worked well for our case and enhanced the labeling accuracy.

It can be seen that indoor scene labeling is a challenging problem due to the diverse nature of the scenes. The major reason for the low reported scene labeling accuracies (see Table 2) is the presence of a large number of objects with varying textures and layouts across different images. These varied appearances of objects cause many ambiguities. Also there are many bland regions in the scenes, which introduce an additional challenge for a correct segmentation. Many times class errors are due to the confusion between two similar classes e.g., as evident in the confusion matrices (Fig. 12), *door* is usually confused with *wall*, *blind* with *window*, *sink* with *counter* and *sofa* with *bed*. Despite the incorporation of the colorbluegeometrical context, an unusual confusion occurs between *ceiling* and *wall*. The reason is that the depth estimates in the regions close to the upper boundary of the scenes were not accurate and this is the typical location where the ceiling normally occurs in the majority of the scenes. The planes extracted in this region give a horizontal orientation (instead of vertical) which contributes to this misclassification, aided by the fact that the walls and ceilings usually have similar appearances.

The NYU corpus captures natural indoor scene conditions which are common in everyday life scenarios. As an example, the dataset contains large illumination variations (e.g., for scenes of offices, stores) which correctly capture

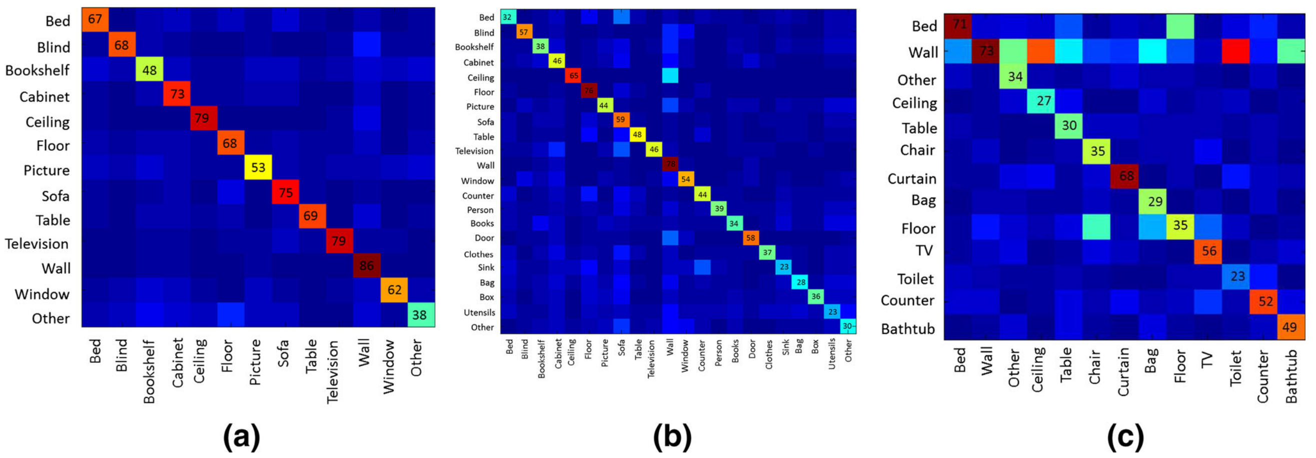

Fig. 12 Confusion Matrices for NYU-Depth dataset: the accuracies in each confusion matrix sum up to 100 % along *each row*. All the class accuracies shown on the diagonal are rounded to the closest integer for clarity. (*Best viewed in color*), **a** NYU-Depth v1, **b** NYU-Depth v2, **c** SUN3D

the indoor conditions. Some misclassifications are possibly due to these illumination variations and specular surfaces e.g., the window or the reflecting mirror was confused with the light source. Another major challenge relates to the *long-tail distribution* of object categories, where a small number of categories appear frequently in indoor scenes while others are rare. For example, the top ten most frequent classes out of a total of 894 classes in the NYU v2 dataset constitutes over 65 % of the total labelled data. This translates into a somewhat unbalanced dataset with an insufficient representation of many semantic classes in the training set (Ren et al. 2012). The labeled portion of the SUN3D database was insufficient for training (because the database is under development). This explains why the achieved accuracies for this database are on the low side (see Table 2; Fig. 12). The availability of more and higher quality training data for each class will certainly improve the performance of scene labeling frameworks. The removal of unwanted artifacts such as illumination variations and shadows can also help in improving the segmentation accuracy (Khan et al. 2014a). In short, the challenging indoor scene classification task is far from being solved and requires further investigation both in terms of new techniques and data for testing and benchmarking.

7 Conclusion

This paper presented a novel CRF model for semantic labeling of indoor scenes. The proposed model uses both appearance and geometry information. The geometry of indoor planar surfaces was approximated using a proposed robust region growing algorithm for segmentation. The approximate geometry was combined with appearance-based information and a location prior in the unary term. A learned combination of boundaries was used to define the spatial discontinuity across an image. The proposed model also captured long-range interactions by defining cliques on the dominant planar surfaces. The parameters of our model were learned using a single slack formulation of the rescaled margin cutting plane algorithm. We extensively evaluated our scheme on both versions of the NYU-Depth and the recent SUN3D database and reported comparisons and improvements over existing works. As a future work, we will extend the proposed model to holistically reason about indoor scenes and to understand the rich interactions between scene elements.

Acknowledgments This research was supported by the IPRS scholarship from The University of Western Australia and the Australian Research Council (ARC) Grants DP110102166, DP150104251 and DE120102960. The authors would especially like to thank the anonymous reviewers and the Associate Editor for their valuable comments and suggestions to improve the quality of the manuscript.

Appendix: Disintegration of Higher-Order Energies

In this appendix, we will show how the higher-order energies can be minimized using graph cuts. Since, graph cuts can efficiently minimize submodular functions, we will transform our higher-order energy function (Eq. 9) to a submodular second-order energy function. For the case of both $\alpha\beta$-swap and α-expansion move making algorithms, we will explain this transformation and the process of optimal moves computation[3]. All of the previously defined notations are used

[3] The development of this section is similar to Kohli et al. (2009). We also used the same notation - wherever possible - to allow the reader to easily sort out differences and commonalities.

in the same context and all of the newly introduced symbols are defined in this section. The function that accounts for the number of disagreeing nodes in a clique is defined as:

$$n_\ell(y_{\mathbf{c}}) = \sum_{i \in \mathbf{c}} w_i^\ell \mathbf{1}_{y_i = \ell}$$

The function $\mathbf{1}_{y_i=\ell}$ is a zero-one indicator function that returns a unit value when $y_i = \ell$. We suppose here that weights are symmetric for all labels $\ell \in \mathcal{L}$ i.e., $w_i^\ell = w_i$. Further, for our implementation we set $w_i = 1 \;\; \forall i \in \mathbf{c}$. This setting satisfies the required constraints for these parameters, i.e.,

$$w_i^\ell \geq 0 \quad \text{and} \quad \sum_{i \in \mathbf{c}} w_i^\ell = \#\mathbf{c} \;\; \forall \ell \in \mathcal{L}.$$

We define a summation function that adds the weights for a subset $\mathbf{s}$ of $\mathbf{c}$,

$$W(\mathbf{s}) = \sum_{i \in \mathbf{s}} w_i^\ell = \#\mathbf{s} \quad \forall \ell \in \mathcal{L}.$$

Disintegration of Higher-Order Energies to Second-Order Sub-modular Energies for Swap Moves

Suppose, in a clique '$\mathbf{c}$', the locations of the active nodes is represented by a set of indices $\mathbf{c}_a$. The nodes which remain inactive during the move making process are termed the *passive* nodes. Their locations are denoted by $\bar{\mathbf{c}}_a = \{\mathbf{c} \backslash \forall c_i \in \mathbf{c}_a\}$. The corresponding set of available moves to the swap move making algorithm are encoded in the form of a vector $\mathbf{t}_{c_a}$. For the sake of a simple demonstration, let us focus on the two class labeling problem i.e., $\ell \in \{0, 1\}$. The induced labeling is the combination of the old labeling for the inactive nodes and the new labeling for the active nodes i.e., $y_c^n = y_{\bar{c}_a}^\circ \cup T_{\alpha\beta}(y_{c_a}^\circ, \mathbf{t}_{c_a})$. If y_c^n denotes the new labeling induced by move $\mathbf{t}_{c_a}$ and y_c° denotes the old labeling, we can define the energy of move for an $\alpha\beta$ swap as:

$$\begin{aligned}
\psi_{\mathbf{c}}^m(\mathbf{t}_{c_a}) &= \psi_{\mathbf{c}}(y_c^n) = \psi_{\mathbf{c}}(y_{\bar{c}_a}^\circ \cup T_{\alpha\beta}(y_{c_a}^\circ, \mathbf{t}_{c_a})) \\
&= \min_{\ell \in \mathcal{L}} \{\lambda_{max} - (\lambda_{max} - \lambda_\ell) \\
&\quad \exp\left(-\frac{W(\mathbf{c}) - n_\ell(y_{\bar{c}_a}^\circ \cup T_{\alpha\beta}(y_{c_a}^\circ, \mathbf{t}_{c_a}))}{Q_\ell}\right)\Bigg\} \\
&= \min_{\ell \in \mathcal{L}} \left\{\lambda_{max} - (\lambda_{max} - \lambda_\alpha)\exp\left(-\frac{W(\mathbf{c}) - n_0^m(\mathbf{t}_{c_a})}{Q_\alpha}\right),\right. \\
&\quad \left.\lambda_{max} - (\lambda_{max} - \lambda_\beta)\exp\left(-\frac{W(\mathbf{c} - \mathbf{c}_a) + n_0^m(\mathbf{t}_{c_a})}{Q_\beta}\right)\right\},
\end{aligned}$$

where, $W(\mathbf{c}_a) = n_0^m(\mathbf{t}_{c_a}) + n_1^m(\mathbf{t}_{c_a})$. The minimization operation in the above equation can be replaced by defining a piecewise function:

$$\psi_{\mathbf{c}}^m(\mathbf{t}_{c_a}) = \begin{cases}
\lambda_{max} - (\lambda_{max} - \lambda_\alpha)\exp\left(-\frac{W(\mathbf{c}) - n_0^m(\mathbf{t}_{c_a})}{Q_\alpha}\right) \\
\quad \text{if} \;\; n_0^m(\mathbf{t}_{c_a}) > \varrho_{\alpha\beta}\left(\frac{W(\mathbf{c})}{Q_\alpha} - \frac{W(\mathbf{c} - \mathbf{c}_a)}{Q_\beta}\right. \\
\qquad \left. - \log\left(\frac{\lambda_{max} - \lambda_\alpha}{\lambda_{max} - \lambda_\beta}\right)\right), \\
\lambda_{max} - (\lambda_{max} - \lambda_\beta)\exp\left(-\frac{W(\mathbf{c} - \mathbf{c}_a) + n_0^m(\mathbf{t}_{c_a})}{Q_\beta}\right) \\
\quad \text{if} \;\; n_0^m(\mathbf{t}_{c_a}) < \varrho_{\alpha\beta}\left(\frac{W(\mathbf{c})}{Q_\alpha} - \frac{W(\mathbf{c} - \mathbf{c}_a)}{Q_\beta}\right. \\
\qquad \left. - \log\left(\frac{\lambda_{max} - \lambda_\alpha}{\lambda_{max} - \lambda_\beta}\right)\right),
\end{cases}$$

where, $\varrho_{\alpha\beta} = \frac{Q_\alpha Q_\beta}{Q_\alpha + Q_\beta}$. The function $n_\ell^m(\mathbf{t}_{c_a})$ is defined as:

$$n_\ell^m(\mathbf{t}_{c_a}) = \sum_{i \in \mathbf{c}_a} w_i \delta_\ell(\mathbf{t}_i).$$

From Theorem 1 in Kohli et al. (2009), the energy defined above can be transformed to the submodular quadratic pseudo-boolean function with two binary meta variables. In this form the $\alpha\beta$-swap algorithm can be used for minimizing the energy function.

Disintegration of Higher-Order Energies to Second-Order Sub-modular Energies for Expansion Moves

Suppose, in a clique ' c', the location of the nodes with label ℓ is represented by a set of indices $\mathbf{c}_\ell$. The current labeling solution is denoted by $y_{\mathbf{c}}^\circ$.

If the dominant label is denoted by $d \in \mathcal{L}$ in the current labeling $y_{\mathbf{c}}^\circ$ is,

$$\text{s.t} \quad W(\mathbf{c}_d) > W(\mathbf{c}) - Q_d \quad \text{where } d \neq \alpha,$$

there must be one dominant label:

$$\begin{aligned}
&Q_a + Q_b < W(\mathbf{c}) \qquad \forall a \neq b \in \mathcal{L}, \\
&\psi_{\mathbf{c}}^m(t_c) = \psi_{\mathbf{c}}(T_\alpha(y_c^\circ, t_c)) \\
&= \min_{\ell \in \mathcal{L}} \left\{\lambda_{max} - (\lambda_{max} - \lambda_\alpha)\exp\left(-\frac{\sum_{i \in c} w_i t_i}{Q_\alpha}\right),\right. \\
&\left.\lambda_{max} - (\lambda_{max} - \lambda_d)\exp\left(-\frac{W(\mathbf{c}) - \sum_{i \in c} w_i t_i}{Q_d}\right)\right\}.
\end{aligned}$$

The minimization operator in the above function can be replaced by a piecewise function:

$$\psi_{\mathbf{c}}^{m}(\mathbf{t}_c, \mathbf{t}_{c_d}) = \begin{cases} \lambda_{max} - (\lambda_{max} - \lambda_{\alpha})\exp\left(-\frac{n_0^m(\mathbf{t}_c)}{Q_\alpha}\right) \\ \quad \text{if} \quad n_0^m(\mathbf{t}_c) > \varrho_{\alpha d}\left(\frac{W(\mathbf{c})}{Q_\alpha} \right. \\ \qquad\qquad \left. - \log\left(\frac{\lambda_{max}-\lambda_\alpha}{\lambda_{max}-\lambda_d}\right)\right), \\ \lambda_{max} - (\lambda_{max} - \lambda_{d})\exp\left(-\frac{W(\mathbf{c})-n_0^m(\mathbf{t}_{c_d})}{Q_d}\right) \\ \quad \text{if} \quad n_0^m(\mathbf{t}_c) < \varrho_{\alpha d}\left(\frac{W(\mathbf{c})}{Q_\alpha} \right. \\ \qquad\qquad \left. - \log\left(\frac{\lambda_{max}-\lambda_\alpha}{\lambda_{max}-\lambda_d}\right)\right), \end{cases}$$

where, $\varrho_{\alpha d} = \frac{Q_\alpha Q_d}{Q_\alpha + Q_d}$ and function $n_\ell^m(\mathbf{t}_c)$ is defined as:

$$n_\ell^m(\mathbf{t}_c) = \sum_{i \in \mathbf{c}} w_i \delta_\ell(\mathbf{t}_i).$$

From Theorem 2 in Kohli et al. (2009), the energy defined above can be transformed to the submodular quadratic pseudo-boolean function with two binary meta variables. In this form the α-expansion algorithm can be used for minimizing the energy function.

References

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *TPAMI*, *33*(5), 898–916.

Blake, A., Kohli, P., & Rother, C. (2011). *Markov random fields for vision and image processing*. Cambridge: The MIT Press.

Boykov, Y., & Funka-Lea, G. (2006). Graph cuts and efficient nd image segmentation. *IJCV*, *70*(2), 109–131.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *TPAMI*, *23*(11), 1222–1239.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(0885–6125), 5–32.

Cadena, C., & Košecká, J. (2014). Semantic segmentation with heterogeneous sensor coverages.

Carreira, J., & Sminchisescu, C. (2012). Cpmc: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, *34*(7), 1312–1328.

Couprie, C., Farabet, C., Najman, L., & LeCun, Y.(2013). Indoor semantic segmentation using depth information. ICLR.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, vol 1 (pp 886–893).

Edwards, W., Miles, R. F, Jr, & Von Winterfeldt, D. (2007). *Advances in decision analysis: from foundations to applications*. Cambridge: Cambridge University Press.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. *TPAMI*, *35*(8), 1915–1929. doi:10.1109/TPAMI.2012.231.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *IJCV*, *59*(2), 167–181.

Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *TIT*, *21*(1), 32–40.

Gould, S., Fulton, R., & Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. In *IEEE ICCV* (pp 1–8).

Gulshan, V., Rother, C., Criminisi, A., Blake, A., & Zisserman, A. (2010). Geodesic star convexity for interactive image segmentation. In *IEEE CVPR* (pp 3129–3136).

Gupta, S., Arbelaez, P., & Malik, J. (2013), Perceptual organization and recognition of indoor scenes from rgb-d images. In *IEEE CVPR* (pp. 564–571).

Gupta, S., Girshick, R., Arbeláez. P., & Malik, J. (2014). Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV* 2014 (pp. 345–360). Springer.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD*, *11*(1), 10–18.

Hayat, M., Bennamoun, M., & An, S. (2015). Deep reconstruction models for image set classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(4), 713–727. doi:10.1109/TPAMI.2014.2353635.

He, X., Zemel, R. S., & Carreira-Perpinán, M. A. (2004). Multiscale conditional random fields for image labeling. In *IEEE CVPR*, vol 2 (pp II–695).

Huang, Q., Han, M., Wu, B., & Ioffe, S. (2011). A hierarchical conditional random field model for labeling and segmenting images of street scenes. In *IEEE CVPR* (pp. 1953–1960).

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 559–568).

Jiang, Y., Lim, M., Zheng, C., & Saxena, A. (2012). Learning to place new objects in a scene. *IJRR*, *31*(9), 1021–1043.

Joachims, T., Finley, T., & Yu, C. N. J. (2009). Cutting-plane training of structural svms. *JML*, *77*(1), 27–59.

Johnson, A. E., & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(5), 433–449.

Khan, S., Bennamoun, M., Sohel, F., & Togneri, R. (2014a). Automatic feature learning for robust shadow detection. In *IEEE CVPR*.

Khan, S., He, X., Bennamoun, M., Sohel, F., & Togneri, R. (2015). Separating objects and clutter in indoor scenes. In *IEEE CVPR*.

Khan, S. H., Bennamoun, M., Sohel, F., & Togneri, R. (2014b). Geometry driven semantic labeling of indoor scenes. In *Computer Vision–ECCV* 2014 (pp. 679–694). Springer.

Kohli, P., Kumar, M. P., & Torr, P. H. (2007). P3 & beyond: Solving energies with higher order cliques. In *IEEE CVPR* (pp. 1–8).

Kohli, P., Torr, P. H., et al. (2009). Robust higher order potentials for enforcing label consistency. *IJCV*, *82*(3), 302–324.

Koppula, H. S., Anand, A., Joachims, T., & Saxena ,A. (2011). Semantic labeling of 3d point clouds for indoor scenes. In *NIPS* (pp. 244–252).

Krähenbühl, P., & Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS* (pp. 109–117).

Ladicky, L., Russell, C., Kohli, P., & Torr, P. H. (2009). Associative hierarchical crfs for object class image segmentation. In *IEEE ICCV* (pp. 739–746).

Ladickỳ, L., Russell, C., Kohli, P., & Torr, P. H. (2013). Inference methods for crfs with co-occurrence statistics. In *IJCV* (pp. 1–13).

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE ICRA* (pp. 1817–1824).

Lempitsky, V., Vedaldi, A., & Zisserman, A. (2011). Pylon model for semantic segmentation. In *NIPS* (pp. 1485–1493).

Li, Y., Tarlow, D., & Zemel, R. (2013). Exploring compositional high order pattern potentials for structured output learning. In *IEEE CVPR* (pp. 49–56).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Quattoni, A., & Torralba, A. (2009). Recognizing indoor scenes. In *CVPR* (pp. 413–420). doi:10.1109/CVPR.2009.5206537.

Quigley, M., Batra, S., Gould, S., Klingbeil, E., Le, Q., Wellman, A., & Ng, A. Y. (2009). High-accuracy 3d sensing for mobile manipulation: Improving object detection and door opening. In *IEEE ICRA* (pp. 2816–2822).

Rabbani, T., van Den Heuvel, F., & Vosselmann, G. (2006). Segmentation of point clouds using smoothness constraint. *IAPR SSIS*, *36*(5), 248–253.

Rao, D., Le, Q. V., Phoka, T., Quigley, M., Sudsang, A., & Ng, A. Y. (2010). Grasping novel objects with depth segmentation. In *IEEE IROS* (pp. 2578–2585).

Ren, X., Bo, L., & Fox, D. (2012). Rgb-(d) scene labeling: Features and algorithms. In *IEEE CVPR* (pp. 2759–2766).

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG, ACM*, *23*, 309–314.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, *81*(1), 2–23.

Silberman, N., & Fergus, R. (2011). Indoor scene segmentation using a structured light sensor. In *IEEE ICCV Workshops* (pp. 601–608).

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *ECCV* (pp. 746–760). Springer.

Szummer, M., Kohli, P., & Hoiem, D. (2008). Learning crfs using graph cuts. In *ECCV* (pp 582–595). Springer.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *ACM ICML* (p 104).

Van De Weijer, J., & Schmid, C. (2006). Coloring local feature extraction. In *ECCV* (pp 334–348). Springer

Von Gioi, R. G., Jakubowicz, J., Morel, J. M., & Randall, G. (2010). Lsd: A fast line segment detector with a false detection control. *TPAMI*, *32*(4), 722–732.

Woodford, O. J., Rother, C., & Kolmogorov, V. (2009). A global perspective on map inference for low-level vision. In *IEEE ICCV* (pp. 2319–2326).

Xiao, J., Owens, A., & Torralba, A. (2013). Sun3d: A database of big spaces reconstructed using sfm and object labels. In *IEEE ICCV*

Xiong, X., & Huber, D. (2010). Using context to create semantic 3d models of indoor environments. In *BMVC* (pp. 45–1).