# Unsupervised object region proposals for RGB-D indoor scenes 

Zhuo Deng ${ }^{\mathrm{a}, *}$, Sinisa Todorovic ${ }^{\text {b }}$, Longin Jan Latecki ${ }^{\text {a }}$<br>${ }^{\text {a }}$ CIS Department, Temple University, Philadelphia, 1925 N. 12th St, USA<br>${ }^{\mathrm{b}}$ School of EECS, Oregon State University, Corvallis, 2107 Kelley Engineering Center, USA

## ARTICLE IN F O

## Article history:

Received 29 January 2016
Revised 18 May 2016
Accepted 21 July 2016
Available online $x x x$

## Keywords:

Object segmentation
RGB-D
Sensor fusion


#### Abstract

In this paper, we present a novel unsupervised framework for automatically generating bottom up class independent object candidates for detection and recognition in cluttered indoor environments. Utilizing raw depth map from active sensors such as Kinect, we propose a novel plane segmentation algorithm for dividing an indoor scene into predominant planar regions and non-planar regions. Based on this partition, we are able to effectively predict object locations and their spatial extensions. Our approach automatically generates object proposals considering five different aspects: Non-planar Regions (NPR), Planar Regions (PR), Detected Planes (DP), Merged Detected Planes (MDP) and Hierarchical Clustering (HC) of 3D point clouds. Object region proposals include both bounding boxes and instance segments. Our approach achieves very competitive results and is even able to outperform supervised state-of-the-art algorithms on the challenging NYU-v2 RGB-Depth dataset. In addition, we apply our approach to the most recently released large scale RGB-Depth dataset from Princeton University - "SUN RGBD", which utilizes four different depth sensors. Its consistent performance demonstrates a general applicability of our approach.


© 2016 Published by Elsevier Inc.

## 1. Introduction

Automatically generating high quality class independent object segmentations is important for many high level computer vision problems such as object detection and recognition. For object recognition, since feature extraction relies directly on the information of its supporting region, the full object region not only conveys global features but also arguably enriches contextual features as confusing background is separated (Cinbis et al., 2013). For object detection, an object can be located at any position and scale in the image. Most of existing work (Felzenszwalb et al., 2010; Viola and Jones, 2004) is based on sliding window strategy where exhaustive searching is conducted at various scales and window aspect ratios. However, expensive computation prevents this strategy from utilizing sophisticated feature representations. As an alternative, providing a small set of high quality location hypotheses makes it possible to adopt richer features and complex learning algorithms (Cinbis et al., 2013; Dong et al., 2014; Hariharan et al., 2014).

Many previous works are dedicated to propose class independent object hypotheses. Uijlings et al. (2013) proposed a selective search strategy that hierarchically groups similar neighbor superpixels obtained from (Felzenszwalb and Huttenlocher, 2004) for

[^0]predicting object locations. In contrast, besides predicting object bounding boxes, we also aim at providing pixel-level object segments. Carreira and Sminchisescu (2012) generated a set of object segments by solving one constrained parametric min-cut (CPMC) problem for each configuration of predefined foreground and background seeds. Lin et al. (2013) simply extends CPMC by integrating depth for computing potentials. Instead of treating all image region uniformly, we tactically generate hypotheses according to classified regions. Gupta et al. (2013) generalized gPb-UCM hierarchical segmentation (Arbelaez et al., 2011) by making effective use of depth information. Arbelaez et al. (2014) proposed Multiscale Combinatorial Grouping (MCG) to collect segments from multiscale aligned gPb-UCM segmentations. Gupta et al. (2014) extend MCG to utilize depth cues for region proposals. While (Arbelaez et al., 2014; Gupta et al., 2013, 2014) need to learn contour models or/and Pareto front for combinatorial purpose, our approach proposes object regions in an unsupervised way.

We have designed and implemented an integrated system for automatically proposing both object bounding boxes and pixellevel segments in RGB-D images. All the object candidates are generated without any training stage. The overall architecture is presented in the diagram shown in Fig. 1. The source code of this work will be available online.

We first estimate a general scene layout by fitting planes to 3D points recovered from depth maps. Hence we utilize a common strategy of distinguishing clutter regions from planar re-


Fig. 1. The diagram of the proposed system for generating object regions in indoor scenes. Taking one color image and corresponding registered raw depth map from Kinect sensors as inputs, our approach automatically generates object proposals considering five different aspects: Non-planar Regions (NPR), Planar Regions (PR), Detected Planes (DP), Merged Detected Planes (MDP) and Hierarchical Clustering (HC) of 3D point clouds. Object region proposals include both bounding boxes and instance segments. The bottom row shows several examples of generated instances and bounding boxes (green color). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
gions. In contrast to earlier works like (Hedau et al., 2009), we do not make any assumptions that edges representing joints of walls/floor/celling are visible. Such assumptions were necessary when only RGB data is given. Since we also utilize depth data, the planar surface may represent different objects like table top or other furniture tops. Then we classify planar regions into boundary and non-boundary planes, where a boundary plane is a plane with no objects behind it, e.g., walls and floors. Depending on the scene a table top can also be a boundary plane. Crude bounding box (BB) object proposals are obtained by fitting BBs to planar regions and to segments obtained from Multi-Channel Multi-Scale (MCMS) segmentations and 3D point cloud clustering with the guidance of the estimated scene layout. Finally, we utilize GrabCut (Rother et al., 2004) to generate segment proposals and refined BB proposals. GrabCut is an excellent foreground object segmenter that is able to dynamically model global object and background properties. However, it has two major limitations. It was developed as (1) interactive human in the loop approach, and it is based on the assumption that (2) the input image contains only one salient object and its background. We address both limitations in the proposed framework and turn GrabCut into a fully automatic, unsupervised segmenter. A general outline of the proposed approach is as follows:

1. Estimate scene layout (Section 2.2)
(a) fitting planes to reconstructed 3D points
(b) classify planar regions into boundary and non-boundary planes
2. Generate crude BB object proposals (Section 2.3)
(a) Multi-Channel Multi-Scale (MCMS) segmentations
(b) Euclidean point cloud clustering
(c) five strategies to generate crude BB proposals
3. Use extended GrabCut to generate segment proposals and refined BB proposals (Section 2.1)

We evaluate the proposed approach on standard NYU-v2 RGBD dataset (Silberman et al., 2012) and recent released large scale SUN RGBD dataset (Song et al., 2015) in Section 3.

To summarize, the main contributions of our approach are: 1) A novel scene structure guided framework for generating bottom-up object region candidates in cluttered indoor scenes. The framework is completely unsupervised, so there is no need to access ground truth information for region proposals, and no bias resulting from the selection of training data. 2) The number of proposed object
regions is much less than the state-of-the-arts while the performance is comparable. Hence the proposed framework has a great potential for high-level computer vision tasks such as object detection and recognition. 3) A novel 3D plane segmentation algorithm that is able to detect and segment predominant planar structures of indoor scenes. It is demonstrated to be robust to noise in structured light and other depth sensors.

## 2. Object region proposals in RGBD images

### 2.1. Grabcut extension

In this section we describe our extension of GrabCut that generates final object segments and BB proposals. The input are initial crude BBs generated by component two.

GrabCut (Rother et al., 2004) is an iterative GraphCut (Boykov and Funka-Lea, 2006) based segmentation algorithm. Given a region of interest (ROI) in an image, pixels inside ROI are initially labeled as "unknown" and outside are labeled as "background". The goal of GrabCut is to identify the object pixels within this "unknown" region. In general, two Gaussian Mixture Models (GMMs) of $K$ components ( $K=5$ typically) are used to model foreground and background color distributions, respectively. Model parameters $\pi, \mu, \Sigma$ are weights, mean and covariance matrices of the $2 K$ Gaussian components:
$\underline{\theta}=\{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha=0,1, k=1 \ldots K\}$,
where $\alpha$ represents the foreground or background. A Gibbs energy function $E$ is defined on the graph $G$ in Eq. (2), where each pixel is taken as a node.

$$
\begin{equation*}
E=\sum_{i=1}^{n} D\left(p_{i}, \alpha, \underline{\theta}\right)+\sum_{(u, v) \in C} \gamma *\left[\alpha_{u} \neq \alpha_{v}\right] * \exp \left(-\beta\left\|p_{u}-p_{v}\right\|^{2}\right) \tag{2}
\end{equation*}
$$

The data term $D$ encodes the probability of pixel $p_{i}$ belonging to foreground or background. It is defined as GMM of $K$ components. The smoothness term encourages regional coherence when pixels have similar properties. $\gamma$ is a constant for balancing data term and smoothness term. $C$ represents the set of pairs of adjacent pixels (we use 4 -connectivity), and the constant $\beta$ is set as inverse of expectation of pixel differences over $C$ defined in Eq. (3). At each iteration, the optimal label assignment is obtained by minimizing


Fig. 2. Image samples comparison. The first three images are from GrabCut dataset. The last one from NYU-V2 dataset presents a typical cluttered indoor scene.


Fig. 3. Examples for foreground segmentation comparison between GrabCut (GC) and its 3D extension (GC3D) both initialized with BBs in yellow frames. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
energy $E$ using GraphCut. Then GMMs parameters in Eq. (1) are updated according to the label assignment.
$\beta=\frac{\sum_{(u, v) \in C} 1}{2 \sum_{(u, v) \in C}\left(\sqrt{\left\|p_{u}-p_{v}\right\|^{2}}\right)}$
GrabCut is an interactive segmentation algorithm in that it needs human to provide some hint such as a bounding box around the object candidate. Moreover, it is designed for images consisting of one single salient object with nearly uniform background, e.g., see Fig. 2.

We observe that when GrabCut is initialized with BBs around object proposals both requirements are met. Our initial guess for object locations is obtained as crude image segments described in Section 2.3. Therefore, we initialize it with BBs around crude segments. In order to increase the chance to cover the whole object by the BB region, we in practice slightly enlarge the BB region. The initial foreground object model is then estimated on the BB region while the initial background model is estimated on the remaining part of the image. It is worth noting that while the whole image is needed for foreground and background model estimations, the object segments are only based on local solution to Eq. (2), i.e., the nodes of graph $G$ are pixels within this region. By solving Eq. (2) locally for each proposal BB we convert GrabCut into a fully automatic, multiple object segmenter.

Although original GrabCut algorithm shows good performance on foreground segmentation, it often fails to segment objects which have similar color distributions as background, or sometimes decomposes objects into several separated components in image plane. For example, in Fig. 3, the foreground derived from GrabCut consists of several disconnected pieces and some parts that should belong to the toilet instance are missing.

In order to avoid assigning different labels to pixels that are spatially close, we extend GrabCut by utilizing depth information. We first fill missing data in raw depth map using colorization scheme of (Levin et al., 2004) and extract 3D points ( $x, y, z$ ). Then 3D point coordinates (in cm unit) are simply concatenated with RGB channels at each pixel. Hence we consider 6 dimensional GMMs.

Although on average the extended GC3D outperforms the original one due to utilization of depth data, e.g., as is shown in Fig. 3, the toilet instance has been segmented well even if it has similar color distribution to the background, the performance of GC3D may degrade when noise in depth is present. One example is shown in the right scene of Fig. 3, where a small piece of background is mis-classified. In this case the original GrabCut works well, since the color of the foreground object differs signif-
icantly from the background. Therefore, we output the segments from both GrabCut and GC3D as our final segment candidates.

### 2.2. Scene layout estimation

Structured indoor environments are often filled with man-made structures and objects, which can be approximately represented with planar segments. We first focus on extracting predominant planar regions such as wall, floor, blackboard, cabinet etc from dense point clouds derived from the depth image, not only because planar regions themselves are meaningful but also they are helpful for generating object hypotheses by focusing on point cloud not explained by major planes. As is well known, comparing to laser range finder, depth information from Kinect and similar sensors has low depth resolution and a limited distance range. To deal with such kind of noise contained in the depth image, traditional plane segmentation methods (Khan et al., 2014; Silberman et al., 2012) resort to appearance based cues from RGB image. For example, Silberman et al. (2012) infers the assignment of points to planes by modeling Graph-Cuts with color and depth information, while (Khan et al., 2014) utilizes detected line segments in color image to decide about region continuity. However, we believe that integrating color information here is a double sword, since the RGB image maybe noisy. Therefore, we use only 3D point clouds for plane detection and propose a plane segmentation algorithm that is designed to work with point clouds generated by Kinect like sensors.

Plane Segmentation: We first determine the direction of gravity (Gupta et al., 2013) and then rotate the point clouds to make them aligned with room coordinates. A normal vector $N_{p}$ is estimated for each point $p$ that has valid depth information, which we call a valid point. To initialize plane candidates, we uniformly sample triple point sets on the depth map and store them in set $T=\left\{\left(p_{i 1}, p_{i 2}, p_{i 3}\right), i=1,2, \ldots\right\}$. Then for each $t_{i} \in T$ we find inliers $S_{i}$ in the 3D space and a plane candidate $P_{i}$ in RANSAC framework (Fischler and Bolles, 1981). Each inlier is represented by a pixel in the depth map and a corresponding 3D valid point. See steps 1-6 in Algorithm 1. The definition of inliers follows below.

In general, a point is considered as an inlier when its distance to the plane is within certain constant range (Hähnel et al., 2003; Poppinga et al., 2008). However, as indicated in Khoshelham and Elberink (2012), depth resolution (i.e., minimum depth difference that can be measured by a sensor) is inversely proportional to the depth, which is defined in Eq (4), where $f$ is focal length, $b$ is base length of Kinect sensors, $m$ is the parameter of a linear normalization and $Z$ represents depth value. Therefore, we vary plane inlier distance tolerance based on depth resolution rather than heuristically choosing one constant threshold.
$D_{\text {tol }}=\left(\frac{m}{f b}\right) * Z^{2}$
We define a point $p$ to be an inlier of plane $P_{i}$ if $d\left(p, P_{i}\right)<D_{\text {tol }}(d)$, where $d$ is the Euclidean distance in 3D space. We then remove plane candidates which have small number of pixels and merge spatially close and nearly coplanar planes. However, many fake planes which consider points of other non-planar objects as inliers exist due to noisy surface normals and depth. To filter out fake planes (steps $10-20$ of Algorithm 1), we first compute connected

```
```

Algorithm 1 Plane Segmentation of Indoor scenes

```
```

Algorithm 1 Plane Segmentation of Indoor scenes
Input: Raw depth map and its 3 d point cloud $\left\{p_{i}, i=1,2, \ldots\right\}$ in
Input: Raw depth map and its 3 d point cloud $\left\{p_{i}, i=1,2, \ldots\right\}$ in
room coordinate system.
room coordinate system.
Output: A series of major plane segments.
Output: A series of major plane segments.
1: compute distance tolerance $D_{\text {tol }}$ accordingto Eq. (4) and normal
1: compute distance tolerance $D_{\text {tol }}$ accordingto Eq. (4) and normal
vector $N_{p}$ for each valid point.
vector $N_{p}$ for each valid point.
uniformly sample triple point sets $T$ on image grid.
uniformly sample triple point sets $T$ on image grid.
for $t \in T$ do
for $t \in T$ do
get plane candidate $P_{i}$ and its inlier set $S_{i}=\left\{p \mid d\left(p, P_{i}\right)<\right.$
get plane candidate $P_{i}$ and its inlier set $S_{i}=\left\{p \mid d\left(p, P_{i}\right)<\right.$
$\left.D_{\text {tol }}(p),\left\langle N_{P_{i}}, N_{p}\right\rangle<t h_{N}\right\}$
$\left.D_{\text {tol }}(p),\left\langle N_{P_{i}}, N_{p}\right\rangle<t h_{N}\right\}$
discard plane $P_{i}$ if the inlier number in $S_{i}$ is less than
discard plane $P_{i}$ if the inlier number in $S_{i}$ is less than
$t h_{\text {min_pts }}$.
$t h_{\text {min_pts }}$.
end for
end for
sort $\left\{P_{i}\right\}$ w.r.t \# of inliers in decreasing order and remove heav-
sort $\left\{P_{i}\right\}$ w.r.t \# of inliers in decreasing order and remove heav-
ily overlapping ones.
ily overlapping ones.
merge spatially close and nearly parallel plane candidates.
merge spatially close and nearly parallel plane candidates.
remove points that have multiple plane IDs from sets $\left\{S_{i}\right\}$.
remove points that have multiple plane IDs from sets $\left\{S_{i}\right\}$.
for each survived $P_{i}$ do
for each survived $P_{i}$ do
compute connected components $C C_{i}=\left\{c_{i 1}, \cdots, c_{i j}, \cdots\right\}$ of $S_{i}$
compute connected components $C C_{i}=\left\{c_{i 1}, \cdots, c_{i j}, \cdots\right\}$ of $S_{i}$
in the depth map.
in the depth map.
for each component $c_{i j}$ do
for each component $c_{i j}$ do
remove $c_{i j}$ from $S_{i}$ if its size is small. $\mathrm{O} / \mathrm{W}$, estimate plane
remove $c_{i j}$ from $S_{i}$ if its size is small. $\mathrm{O} / \mathrm{W}$, estimate plane
$P_{c_{i j}}$ by RANSAC.
$P_{c_{i j}}$ by RANSAC.
4: $\quad$ if $\operatorname{acos}\left(N_{P_{i}}, N_{P_{c_{i j}}}\right)>10^{\circ}$ then
4: $\quad$ if $\operatorname{acos}\left(N_{P_{i}}, N_{P_{c_{i j}}}\right)>10^{\circ}$ then
5: add new plane $P_{c_{i j}}$ to the plane set if $\operatorname{size}\left(c_{i j}\right)>$
5: add new plane $P_{c_{i j}}$ to the plane set if $\operatorname{size}\left(c_{i j}\right)>$
$t h_{\text {min_pts }} ;$ remove points $c_{i j}$ from $S_{i}$.
$t h_{\text {min_pts }} ;$ remove points $c_{i j}$ from $S_{i}$.
end if
end if
end for
end for
discard $P_{i}$ if the remaining inliners is less than $t h_{\text {min_pts }}$.
discard $P_{i}$ if the remaining inliners is less than $t h_{\text {min_pts }}$.
end for
end for
re-estimate plane parameters for $P_{i}$ by RANSAC on its current
re-estimate plane parameters for $P_{i}$ by RANSAC on its current
inliers.
inliers.
re-sort planes w.r.t their number of inliers in descreasing order.
re-sort planes w.r.t their number of inliers in descreasing order.
assign pixels to planes one by one if $d\left(p, P_{i}\right)<3 \cdot D_{\text {tol }}(p)$ and
assign pixels to planes one by one if $d\left(p, P_{i}\right)<3 \cdot D_{\text {tol }}(p)$ and
$\left\langle N_{P_{i}}, N_{p}\right\rangle<t h_{N}$
$\left\langle N_{P_{i}}, N_{p}\right\rangle<t h_{N}$
3: for each plane, remove its components where $d_{\text {avg }}>D_{\text {tolavg }}$ and
3: for each plane, remove its components where $d_{\text {avg }}>D_{\text {tolavg }}$ and
angle ${ }_{\text {avg }}>t h_{a}$
angle ${ }_{\text {avg }}>t h_{a}$
24: filter out plane component whose size is less than $t h_{\text {min_pts }}$.

```
```

    24: filter out plane component whose size is less than \(t h_{\text {min_pts }}\).
    ```
```

components $C C_{i}=\left\{c_{i 1}, \cdots, c_{i j}, \cdots\right\}$ of pixels in $S_{i}$ in the depth map. Then we fit a plane $P_{c_{i j}}$ to 3D points in each connected component $c_{i j}$ and estimate the plane parameters, including its normal $N_{P_{c_{i j}}}$ We assume that $N_{P_{C_{i j}}}$ should be at least similar to $N_{P_{i}}$. Hence, if the angle between $N_{P_{i}}$ and $N_{P_{c_{i j}}}$ is large, we remove the connected component $c_{i j}$ from $C C_{i}$. We then re-estimate plane parameters of $P_{i}$ based on inlier points in survived components.

For plane segmentation, which is performed on the depth map, we assign to plane $P_{i}$ corresponding pixels in image plane if $d(p$, $\left.P_{i}\right)<3 D_{\text {tol }}(p)$. The goal is to avoid artificial holes on plane segments on the depth map. Since now preliminary plane segments are available, we further remove false positive plane segments by checking statistical features, i.e., average to-plane distance $d_{\text {avg }}$ and average normal angle angle ${ }_{\text {avg }}$ between the average of normals of points in $S_{i}$ and plane normal $N_{P_{i}}$. More details are illustrated in Algorithm 1. To our best knowledge, we are the first to segment multiple indoor planes by considering quadratic sensor noise model and relying purely on 3D point cloud. Khoshelham and Elberink (2012) only proposed the depth noise model but did not apply it to multiple plane segmentation. Silberman et al. (2012) use a linear noise model to detect planes and use color information for pixel assignment.

Plane Classification: After major planar regions are detected, we further classify them into boundary and non-boundary planes,
where a boundary plane is a plane with no objects behind it. Supposed that the normal vector of a plane points towards the viewer, we compute the ratio $r$ of points on the other side of the plane to the total number of points in the room. Ideally, a planer region is a boundary plane if $r$ is zero. We set $r$ to 0.01 to tolerate the sensor noise.243

### 2.3. Initial crude region and BBs proposals

Indoor scenes are usually composed of several predominant planar geometric structures such as ceiling, floor, wall, cabinet, etc and many small cluttered things including clothes, bottles, cups, etc. Based on this prior knowledge, we propose to generate object regions by different strategies with respect to the geometric properties of image regions, rather than treating all image regions uniformly. Since low level image segmentations often indicate cues for object candidate shapes and locations, we adopt Multi-Channel Multi-Scale (MCMS) segmentations for obtaining crude object segments. Note that segments obtained from MCMS segmentation are crude (either too coarse or too fine), and they do not represent final instance segments we are looking for. We utilize five different strategies, described below, to select crude segments for initializing object BB proposals.

For objects in Non-Planar Regions (NPRs) (e.g., cups, faucets etc.) all segments except those that have small overlapping area with NPRs are used, while for objects in Planar-Regions (PRs) (e.g., pictures, papers, etc.) only segments that are generated from RGB channel segmentation and lie in the planar areas are reserved. Segments from Detected Planes (DPs) can be used directly for objects such as ceiling, wall, floor, etc. However, sometimes big objects are inclined to be decomposed into several planar regions (e.g., bed and sofa), and then it is very likely that the proposed bounding boxes are not covering the whole object.

To address this problem, we focus our attention on nonboundary planes, which usually represent big objects like bed or other furniture. For each non-boundary planar region, we then find its border points, which are used to compute minimum distance to other non-boundary planer regions. This distance is used to merge non-boundary planar regions that are close in 3D space (within 5 cm ) to obtain Merged Planar Regions (MPRs). BBs are then fitted to MPRs.

In addition, we apply Hierarchical Clustering (HC) to 3D point cloud to obtain object instances that are ambiguous in the color image while separated well in 3D world.

### 2.3.1. Multi-Channel multi-Scale (MCMS) image segmentations

Indoor scenes typically consists of a relatively large number of alike objects that are often cluttered and in disorder, which makes our task of finding a small set of high quality class independent object candidates non-trivial. Moreover, the contents in images are intrinsically organized in a hierarchical way. For example, in Fig. 5, the "bed" can refer only to mattress and box part or include everything on its top such as sheet and pillows. Besides, indoor scene objects are always in different sizes, colors and shapes, and presented under various light conditions. Therefore, it seems impossible to get object partitions from a generic segmentation strategy that relies on a single signal. Based on these observations, we propose to initialize object locations by using low level segmentations from multiple signal channels and image scales.

In this paper, we get low level segments based on two unsupervised segmentation methods: graph based segmentation (GBS) Felzenszwalb and Huttenlocher (2004) and watershed based segmentation (WBS) Meyer (1992) for their high computing efficiency, but other excellent generic image segmenters such as gPb-UCM (Arbelaez et al., 2011) could also be used in our framework. For GBS, except for using color image alone, we use depth map and


Fig. 4. An example of Euclidean clustering of 3D point cloud. (a) Color image: two adjacent blue chair instances within yellow bounding box share similar appearance. (b) The plane segmentation (refer to Section 2.2). (c) 3D point clusters at 5 cm scale. (d) Proposed bounding boxes (red) based on point clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
combined RGB-D channels for computing the edge weights of neighboring pixels at different scales respectively. To be more specific, in total we collect superpixels from 10 different layers based on GBS including 4 scales from color channel, 3 scales from depth channel and 3 scales from RGB-D fusion channels. In RGB-D fusion channels, we normalize associated 3D point coordinates extracted from raw depth into [0,255], and compute affinity weights as the maximum gradient value of RGB and depth channels. In practice, the segmentations from multi-scale GBS are helpful for finding most of object locations but are inclined to ignore some salient objects that only occupy small number of pixels in images. To fixed the problem, we adopt WBS as a complementary segmentation tool, which shows more respect to salient object boundaries.

In WBS, we first smooth input maps using a $9 \times 9$ Gaussian mask and then compute gradient magnitude maps. Since we care more about strong boundaries, we normalize gradient maps into $[0,1]$ range and keep values that are above a predefined threshold (we use 0.1 in this paper). This is also useful for avoiding generating segments that are too fine. Then we apply watershed algorithm to gradient maps estimated from intensity image in CIELAB color space, rawDepth map, inpainted depth map, and normals map, respectively. For each gradient map, we obtain one single layer segmentation. As is mentioned in Section 2.3, using superpixels from color channel GBS only for object proposal in planar regions is an effective strategy for reducing redundant proposals obtained from other signals. But we do not apply the same strategy to WBS segmentations.

### 2.3.2. Euclidean clustering of point cloud

The goal of point cloud clustering is to partition 3D points into several meaningful structures. Taking advantage of 3D geometry of 3D scenes, it is able to remove ambiguities between object instances caused by similar colors or poor illuminations in indoor environments. Take the two chair instances that are within the yellow bounding box in Fig. 4, for example. While it is very difficult to distinguish them based on color image alone, they are well separated in the 3D world. We adapt the Euclidean clustering algorithm in Rusu (2010) for generating object candidates from 3D points.

We first remove detected predominant planes (both horizontal and vertical) from point cloud before clustering. Then we create a Kd-tree representation for the remaining 3d points. As depth data from Kinect sensor are noisy, we filter out sparsely distributed or isolated points (less than 30 points within $1 \mathrm{~cm}^{3}$ ) and get a point cloud $P$. Starting from any point $p_{i} \in P$ as one cluster, we search for its unlabeled neighbors that are within certain radius $d_{t h}$ and add them into the cluster. Then we keep searching neighbors for each member of current cluster until the size of cluster is stable. Clustering terminates when all points in $P$ are assigned a cluster label. Similar to 2D segmentation, we set multiple radii $d_{t h}$ for getting multiple scale clusters ( $d_{t h} \in\{2,5,10\} c m$ ). In Fig. 4, we present one example of Euclidean clustering in a typical office environment, where both blue chair instances and green plant instances are well identified. Moreover, planar instances such as door and

Table 1
Performance comparison of plane segmentations on NYU Depth V2 dataset. Jaccard Index (JI) is used as metric for evaluating obtained planar segments w.r.t. both Exactly Planar Classes (EPC) and Exact and Nearly Planar Classes (E+NPC).

| Method | Silberman et al. (2012) | Khan et al. (2014) | Ours |
| :--- | :--- | :--- | :--- |
| EPC JI | $34.15 \%$ | $33.87 \%$ | $\mathbf{3 6 . 7 2 \%}$ |
| E+NPC JI | $30.91 \%$ | $32.33 \%$ | $\mathbf{3 2 . 6 7 \%}$ |

white board are also identified. We use red bounding boxes to mark identified instances.

## 3. Experiments

We compare our method with the state-of-the-art methods on the NYU Depth V2 dataset (Silberman et al., 2012). Since some of baselines generate their object proposals with supervised learning, for fair comparison, we follow the standard split (i.e., 795 training images/654 test images), and report results on test set, except for plane segmentation evaluation which is measured on the whole dataset. To demonstrate, the general applicability of our approach, we also test on a large scale dataset "SUN RGBD" Song et al. (2015) without changing any parameters.

In our approach, we provide two sets of bounding boxes: one called BB-init, which are all bounding boxes used to initialize foreground segmentations (FG) in Section 2.1, and the other called BBfull that includes bounding boxes fitted to segments obtained by FG plus bounding boxes fitted to segments obtained by plane and watershed segmentations.

### 3.1. Evaluating plane segmentations

We compare with two state of art works (Khan et al., 2014; Silberman et al., 2012) with respect to plane segmentation on RGBD images. For qualitative evaluation, we provide segmentation results under different indoor scenarios in Fig. 5. Both (Silberman et al., 2012) and (Khan et al., 2014) utilize color image with depth map for region smoothness consideration. However, they either fail to detect certain predominant planar regions or have planar regions spread across multiple object boundaries, while our method shows more respect to geometric boundaries and have most major planes detected (e.g., the window frame plane in the office). In addition, we provide quantitative evaluation in Table 1. Following (Khan et al., 2014), we consider both Exactly Planar Classes (EPC) (e.g., floor, ceiling, wall, cabinet etc) and Exact and Nearly Planar Classes (E+NPC) (e.g., bookshelf, books, sofa, bed etc) for evaluation. We compare the obtained planar segments with planar object instances by averaged Jaccard Index. In both cases, our method outperforms the other two methods.

## Failure cases analysis

In the Fig. 5, we present 4 scenes that have failure detection cases. One case is false positive. For example, in the fifth row, the


Fig. 5. Examples of qualitative plane segmentations for RGB-D indoor scenes. The 1st column are original color images. The 2nd column presents plane segmentations by Silberman et al. (2012). The 3rd column shows plane segmentations by Khan et al. (2014). We present our segmentation results in the last column. The black pixels mark non-planar objects. The last four rows show some failure cases..(For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
man's body and part of his arm has been identified as one plane. And in the 6th row, the surface of the ladder is merged with the green bag since they are co-planar in the space. The other case is missing detection. Taking the 7th row for example, a majority part of scene is lacking of depth data since infra-red light was lost under a strong sun shine. Another example is from last row where the table is transparent so that the raw depth does not reflect a real plane surface.

### 3.2. Evaluating object region proposals

### 3.2.1. NYU-V2 Dataset

In this section, we compare our object proposal approach with five state-of-the-art class independent object proposal methods on

NYU-V2 RGBD dataset. MCG (Arbelaez et al., 2014), MCG3D (Gupta et al., 2014), and gPb3D (Gupta et al., 2013) are supervised methods, and CPMC (Carreira and Sminchisescu, 2012), CPMC3D (Lin et al., 2013) are unsupervised methods (excluding segments ranking). Following MCG (Arbelaez et al., 2014), for object segmentation evaluation, we compute global Jaccard Index (i.e., intersection over the union of two sets) at instance level as the average best overlap for all the ground truth instances in the dataset, in order to avoid bias on object sizes. For object location proposals, we define bounding box proposal recall score as the ratio of positive predictions that exceed 0.5 Jaccard score, over the number of all ground truth object instance locations. As is shown in Table 2 and Fig. 6, our method achieves the best performance (91.1\%) for object location proposals while our number of maximum proposals is

Table 2
Performance comparison of best global Jaccard Index at instance level for both bounding box and segment proposals on NYU-V2 RGBD dataset.

|  | Gupta et al. (2013) | Carreira and <br> Sminchisescu (2012) | Lin et al. (2013) | Arbelaez et al. (2014) | Gupta et al. (2014) | Ours-BB-init | Ours-BB-full |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  |  | 0.706 | 0.473 | 0.879 | 0.901 | 0.893 |  |
| Global Best (bbox) | 0.74 | 0.646 | 0.478 | 0.737 | $\mathbf{0 . 9 1 1}$ |  |  |
| Global Best (seg) | 0.67 | 885 | 138 | 4202 | $\mathbf{0 . 7 7 9}$ |  |  |
| \# Proposals | 1051 |  |  | 7482 | 1575 |  |  |



Fig. 6. Quantitative evaluation of object region proposals with respect to the number of object candidates on NYU-V2 RGBD dataset. Left: recall curves on proposed bounding boxes evaluation. Right: average best Jaccard Index curves on proposed segments evaluation. Note the curves of MCG3D and CPMC3D are based on supervised ranking of segments, while the other curves including ours do not use any ranking.
only $\mathbf{4 0 \%}$ of the rank- 2 method MCG3D (Gupta et al., 2014). Moreover, our initial bounding boxes require even less proposals ( $\mathbf{2 1 \%}$ of (Gupta et al., 2014)) while the recall score only degrades $2 \%$ w.r.t the best performance.

For object instances proposal, our method also show very competitive performance: our score is $0.9 \%$ less than the best performance but our number of proposals is less than half of theirs. It is worth noting that we do not rank our bounding box proposals in our result presentation, while (Gupta et al., 2014; Lin et al., 2013) perform supervised ranking. Since we already provide high quality object segmentations with much less number of proposals in a complete unsupervised framework, ranking proposals is beyond the scope of this paper.

In addition, we provide results of global Jaccard index at class level for both object location and segmentation proposals in Fig. 7. We divide 894 classes into 40 classes following the definition of (Gupta et al., 2013) including 37 specific object classes and 3 abstract classes: "other struct", "other furniture" and "other props", which include $68,82,707$ subclasses respectively. We obtaine best performance on 26 classes for object location proposals and 9 classes for segment proposals. It is worth noting that our method achieves best performances on the three abstract classes for object location proposals. It indicates that our approach is general to different object types since abstract classes cover $95.8 \%$ subclasses and $32.3 \%$ instances on the test set.

Except for quantitative evaluation, we also provide qualitative evaluation for proposed object regions in Fig. 8. The first six scenes show objects that have been segmented successfully, and in the last two rows we list several failures cases. The grabcut segmenter is inclined to fail either when the foreground and background have similar color information, or when the foreground object is too small or has irregular shapes (e.g, plants).

## Ablation Study

In order to understand the individual impact of the five proposal strategies on the performance of our RGB-Depth object proposal system, we evaluate our algorithm on the NYU-V2 RGB-D dataset by removing one strategy each time. The corresponding results are listed in the Table 3. As can be seen all the strategies contributes to the performance. The ranking of strategies in decreasing significance order is NPR, PR, DP, HC, and MPR.

### 3.2.2. SUN RGBD Dataset

We also test our unsupervised approach without changing any parameters on the recently released SUN RGBD dataset. SUN RGBD is a large scale indoor scenes dataset with a similar scale as PASCAL VOC. It contains 10,335 RGB-D images in total, which are collected from four different active sensors: Intel RealSense, Asus Xtion, Microsoft Kinect v1 and v2. While the first three sensors obtain depth map using IR structured light, the Kinect v2 (kv2) estimates the depth based on time-of-flight. With respect to raw depth data quality, kv2 can measure depth with the highest accuracy but at the same time there are a lot of small black holes in the depth map due to light absorption or reflection. The RealSense has the lowest raw depth quality.

As can be seen in Table 4, in general, our approach exhibits similar performance to the NYUV-2 dataset. We observe that while the bounding box predictions show consistent performance, the accuracy of instance proposals degrades around $2 \%$. This reasonable degradation might be due to higher variance in sensor depth resolution. The average number of proposals is similar to the number on NYUV-2 dataset except for the tests on RealSense data, where it increases by around $50 \%$. This is expected as the effective depth range of RealSense is very short (depth becomes very noisy or missing beyond 3.5 m ).


Fig. 7. Classwise (40-class) performance comparisons based on the standard PASCAL metric (Jaccard Index) at object instance level for both bounding box and segment proposals on the NYU-v2 RGB-D dataset.

Table 3
Ablation study: each time we remove one of the five object proposal strategies from the full system and report how the performance degrades with respect to both bounding box and segment proposals.

|  | no NPR | no PR | no DP | no HC | no MPR | Ours-full |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Global Best (bbox) | 0.666 | 0.813 | 0.889 | 0.897 | 0.901 | $\mathbf{0 . 9 1 1}$ |
| Global Best (seg) | 0.610 | 0.699 | 0.733 | 0.748 | 0.753 | $\mathbf{0 . 7 7}$ |

Table 4
Performance evaluation of our method on the large scale SUN RGB-D dataset, the images of which are collected from four different RGB-D sensors. *: newly captured RGB-D images in Song et al. (2015).

|  | SUN RGB-D dataset (Song et al., 2015) |  |  |  |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| Sensors | Kinect v1 | Kinect v2 | RealSense | Xtion |  |  |  |  |  |
| Resources | B3DO (Janoch et al., 2013) | NYUV2 | $*$ | $*$ | SUN3D (Xiao et al., 2013) |  |  |  |  |
| Global best (bbox) | 0.929 | 0.911 | 0.908 | 0.909 | 0.912 |  |  |  |  |
| Global best (segment) | 0.742 | 0.77 | 0.746 | 0.745 | 0.752 |  |  |  |  |
| \# proposals | 2972 | 3066 | 2971 | 4628 | 2969 |  |  |  |  |



Fig. 8. Qualitative performance evaluation for proposed object segments on NYU-V2 RGBD dataset. Object proposals are highlighted with green color. And several failure cases are provided at the last two rows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

## 4. Conclusion

We propose an unsupervised unified framework for class independent object bounding box and segment proposals. Our method produces object regions with very comparable qualities to the state-of-the-arts while requiring much less proposals, which indicates its great potential for high level tasks such as object detection and recognition. The source code will be available on authors' websites.

## Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1302164.

## References

Arbelaez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (5), 898-916.
Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J., 2014. Multiscale combinatorial grouping. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, pp. 328-335.
Boykov, Y., Funka-Lea, G., 2006. Graph cuts and efficient nd image segmentation. Int. j. comput. vision 70 (2), 109-131.
Carreira, J., Sminchisescu, C., 2012. Cpmc: Automatic object segmentation using constrained parametric min-cuts. IEEE Trans. Pattern Anal. Mach. Intell. 34 (7), 1312-1328.
Cinbis, R.G., Verbeek, J., Schmid, C., 2013. Segmentation driven object detection with fisher vectors. In: Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, pp. 2968-2975.
Dong, J., Chen, Q., Yan, S., Yuille, A., 2014. Towards unified object detection and semantic segmentation. In: Computer Vision-ECCV 2014. Springer, pp. 299-314.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32 (9), 1627-1645.
Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. Int. J. Comput. Vision 59 (2), 167-181.
Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381-395.
Gupta, S., Arbelaez, P., Malik, J., 2013. Perceptual organization and recognition of indoor scenes from rgb-d images. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 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. Springer, pp. 345-360.
Hähnel, D., Burgard, W., Thrun, S., 2003. Learning compact 3d models of indoor and outdoor environments with a mobile robot. Rob. Auton. Syst. 44 (1), 15-27.
Hariharan, B., Arbeláez, P., Girshick, R., Malik, J., 2014. Simultaneous detection and segmentation. In: Computer Vision-ECCV 2014. Springer, pp. 297-312.
Hedau, V., Hoiem, D., Forsyth, D., 2009. Recovering the spatial layout of cluttered rooms. In: Computer vision, 2009 IEEE 12th international conference on. IEEE, pp. 1849-1856.
Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T., 2013. A category-level 3d object dataset: Putting the kinect to work. In: Consumer Depth Cameras for Computer Vision. Springer, pp. 141-165.
Khan, S.H., Bennamoun, M., Sohel, F., Togneri, R., 2014. Geometry driven semantic labeling of indoor scenes. In: Computer Vision-ECCV 2014. Springer, pp. 679694.

Khoshelham, K., Elberink, S.O., 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors 12 (2), 1437-1454.

Levin, A., Lischinski, D., Weiss, Y., 2004. Colorization using optimization. In: ACM Transactions on Graphics (TOG), 23. ACM, pp. 689-694.
Lin, D., Fidler, S., Urtasun, R., 2013. Holistic scene understanding for 3d object detection with rgbd cameras. In: Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, pp. 1417-1424.
Meyer, F., 1992. Color image segmentation. In: Image Processing and its Applications, 1992., International Conference on. IET, pp. 303-306
Poppinga, J., Vaskevicius, N., Birk, A., Pathak, K., 2008. Fast plane detection and polygonalization in noisy 3d range images. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE, pp. 3378-3383.

Rother, C., Kolmogorov, V., Blake, A., 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graphics (TOG) 23 (3), 309-314.
Rusu, R.B., 2010. Semantic 3d object maps for everyday manipulation in human living environments. KI-Künstliche Intelligenz 24 (4), 345-348.
Silberman, N., Hoiem, D., Kohli, P., Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. In: Computer Vision-ECCV 2012. Springer, pp. 746-760.
Song, S., Lichtenberg, S.P., Xiao, J., 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 567-576.
Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W., 2013. Selective search for object recognition. Int. j. comput. vision 104 (2), 154-171.
Viola, P., Jones, M.J., 2004. Robust real-time face detection. Int. j. comput. vision 57 (2), 137-154.

Xiao, J., Owens, A., Torralba, A., 2013. Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, pp. 1625-1632.


[^0]:    * Corresponding author.

    E-mail address: zhuo.deng@temple.edu (Z. Deng).
    http://dx.doi.org/10.1016/j.cviu.2016.07.005
    1077-3142/© 2016 Published by Elsevier Inc.

