

# Locating 3D Object Proposals: A Depth-Based Online Approach

Ramanpreet Singh Pahwa, Jiangbo Lu, *Senior Member, IEEE*, Nianjuan Jiang, *Member, IEEE*, Tian Tsong Ng, Minh N. Do, *Fellow, IEEE*

**Abstract**— 2D object proposals, quickly detected regions in an image that likely contain an object of interest, are an effective approach for improving the computational efficiency and accuracy of object detection in color images. In this work, we propose a novel online method that generates 3D object proposals in a RGB-D video sequence. Our main observation is that depth images provide important information about the geometry of the scene. Diverging from the traditional goal of 2D object proposals to provide a high recall, we aim for precise 3D proposals. We leverage on depth information per frame and multi-view scene information to obtain accurate 3D object proposals. Using efficient but robust registration enables us to combine multiple frames of a scene in near real time and generate 3D bounding boxes for potential 3D regions of interest. Using standard metrics, such as Precision-Recall curves and F-measure, we show that the proposed approach is significantly more accurate than the current state-of-the-art techniques. Our online approach can be integrated into SLAM based video processing for quick 3D object localization. Our method takes less than a second in MATLAB on the UW-RGBD scene dataset on a single thread CPU and thus, has potential to be used in low-power chips in Unmanned Aerial Vehicles (UAVs), quadcopters, and drones.

**Index Terms**—Robot vision, depth cameras, object proposals.

## I. INTRODUCTION

THE rapid development of low-powered Unmanned Aerial Vehicles (UAVs), drones and service robots has introduced a need for automatic detection of interesting objects present in a scene. Such applications may not only assist in navigation of these devices but also help in localizing and identifying the objects that are present in a scene. This paper presents a framework where we integrate depth information captured using a depth sensing device with 2D color images, build a 3D global map of the environment and estimate 3D bounding boxes for the objects of interest that may be present in a given scene by exploiting multi-view scene geometry.

Copyright 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

R. S. Pahwa, J. Lu and N. Jiang are with Advanced Digital Sciences Center (ADSC), Singapore, 138632 (e-mail: ramanpreet9@gmail.com, {jiangbo.lu,nianjuan.jiang}@adsc.com.sg). (*Corresponding author: J. Lu.*)

T. T. Ng is with Institute for Infocomm Research (I<sup>2</sup>R), Singapore. (e-mail: ttng@i2r.a-star.edu.sg).

R. S. Pahwa and Minh N. Do are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, 61801, USA. (e-mail: ramanpreet9@gmail.com, minhdo@illinois.edu).

This study is supported by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A\*STAR). This work was mainly done when R. Pahwa was interning with the Advanced Digital Sciences Center, Singapore.

The idea of 2D object proposals is in fact not new. It has been presented in the form of unsupervised image segmentation for more than a decade. It is until recently, 2D object proposal techniques have become popular in object detection systems. Instead of finding precise and exclusive boundaries for objects, modern 2D object proposal techniques quickly identify regions (potentially highly overlapped) that are very likely to contain an object [1]–[6]. Most importantly, generic object proposals are class independent. This helps in reducing the search space to allow for more sophisticated recognition algorithms and helps in pruning away false positives making the detection step easier [19]. The ability to perceive objects in the scene regardless of their categories is crucial for robots and vehicles, as it can assist the AI agents to understand and explore the real world environment on-the-fly. Recently, Song and Chandraker [7] present an impressive framework where they combine cues from Structure from Motion (SFM), object detection and plane estimation to estimate rigid 3D bounding boxes for moving cars in an outdoor real environment. However, their approach requires a prior knowledge of the number of cars present in a given scene. As they use monocular SLAM to estimate camera poses and track 3D objects in the scene using feature points, their system may suffer in low-texture environments and lack 3D object dimensions and distances that are critical for actual autonomous navigation.

An effective way to obtain true dimension and distance of 3D objects is to use depth cameras. The use of depth cameras are ubiquitous nowadays. Due to their popularity, depth camera research has also exploded in the last few years with discoveries of various applications in computer and robot vision [12]–[17], including object recognition [18]. Unfortunately, existing object proposal solutions designed for RGB images are sub-optimal for depth camera video input, as no depth information is used and no dense temporal consistency is imposed. Scene geometry should be used by a robot to its advantage and the robot can maneuver around the scene to gain critical information about the objects that may be present in the scene [21].

In this work, we propose an online 3D object proposal technique for RGB-D video input of a static scene with the main focus on precision. We use the per-frame result given by existing 2D proposal techniques as our input. Most of the 2D object proposal methods focus on high recall and output several object proposals around an object of interest. Selecting the best object bounding box among these object proposal candidates is non-trivial. We leverage on segmentation cues provided by depth information and aggregate them over con-

secutive frames in 3D by estimating the camera poses using RGB-D SLAM. By doing this, we achieve high precision with light computational overhead. Some recent works use deep learning in a supervised manner to incorporate segmentation cues right from the start for 2D object proposals and object detection [19], [20]. These methods can achieve online object recognition albeit with the use of heavy parallel processing and expensive GPUs. While appreciating the technical elegance of such approaches, we adopt a more practical approach with the objective of providing an efficient and versatile system that can work with any off-the-shelf 2D proposal techniques. In fact, one major distinction of our method from these existing single color image based methods is that we use a RGB-D video sequence. Depth information helps to localize and segment objects more easily with access to scene geometry, which still remains a challenge for color-only approaches. The proposed method can also segment occluded objects in a single frame by taking advantage of the frames in the video sequence. This allows consolidating information across frames jointly to segment out the occluded objects correctly, without placing a burden on the per-frame object segmentation process. Overall, our approach is both effective and efficient to find objects of interest in 3D by using a RGB-D video sequence.

More specifically, this work focuses on indoor, static scenes containing a major supporting plane. However, it can be easily extended to remove multiple supporting planes. The objects are allowed to occlude each other and partially seen in some images as long as their complete view is covered over a consecutive number of frames. Note that we do not assume their distance to the camera.

We make the following contributions:

- We integrate depth information with state-of-the-art 2D object proposal techniques to improve object proposals per frame.
- During the scene capturing process, we exploit the indoor scene geometry to automatically remove any supporting planes such as tables which are not a part of the objects of interest.
- We use the camera pose estimated by a depth based SLAM technique to efficiently register the frames to a global point cloud and output multi-view consistent 3D object proposals as the camera moves around in the scene.

To showcase our results, we perform density-based 3D clustering on the top-ranked points, and display our proposed 3D objects bounded by tight 3D bounding boxes. One key aspect of our approach is that it is efficient using only a single thread CPU, so our system has a good potential of being integrated on top of existing depth based SLAM methods.

Our paper is structured as follows. In Section II, we review other works that are related to our research topic. In Section III, we provide a brief overview of our problem formulation. Section IV and V present the proposed 2D depth based filtering and 3D fusion and refinement steps in detail, highlighting our contributions and observations at each stage. We report our results and comparison with other state-of-the-art methods in Section VI. Finally, in Section VII we conclude this paper and discuss future research directions.

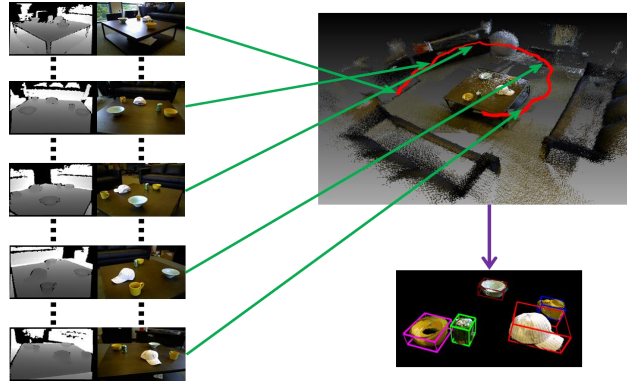


Fig. 1: Scene data is captured using a RGB-D camera. The aligned color and depth images are used for camera pose estimation (illustrated by the red trajectory). Our approach uses this information along with any generic 2D object proposals to fuse and filter the data in 3D and output precise 3D object proposals (denoted by colored 3D bounding-boxes).

## II. RELATED WORK

**Object proposals:** A lot of work has been done recently on 2D object proposals. Traditionally, sliding windows are used along with a detector to identify objects in a given scene. Many of the state-of-the-art techniques in object detection have started using a generic, object-class agnostic proposal method that finds anywhere between 100-10,000 bounding boxes in an image. These areas are considered to have the maximum likelihood to contain an object in them. Such methods vary widely from using linear classifiers, BING [1], graph cuts, CPMC [2], graph cuts with an affinity function [33], normalized cuts, MCG [3] to using random forests, edge-boxes [4], geodesic transform [5] and superpixels [6]. Ren *et al.* [20] use deep learning in a supervised manner to find 2D object proposals and perform object detection simultaneously.

Hosang *et al.* [23] provide a systematic overview of these object proposal techniques and tested these state-of-the-art algorithms. They demonstrate that most of these algorithms have limited repeatability. Even changing one pixel exhibited markedly different outcomes. They found edge-boxes and geodesic proposals to give the best results. Based on these observations, we decided to use edge-boxes [4] as it is reported to be significantly faster and slightly more accurate than geodesic object proposals.

**Simultaneous localization and mapping (SLAM) with scene geometry:** Choudhary *et al.* [24] propose an online object discovery approach that extends standard SLAM to utilize the discovered objects for pose estimation and loop closures. They use 3D feature descriptors, CSHOT [36], to find the camera pose for every frame. They use 3D voxels to roughly segment out objects and use them as landmarks for loop closure.

Song and Chandraker [7] argue that the structure from motion (SFM) cues (3D points and ground plane) and object cues (bounding boxes and detection scores) complement each other. The former is accurate for nearby environments while the latter is more accurate for objects that are far away. They

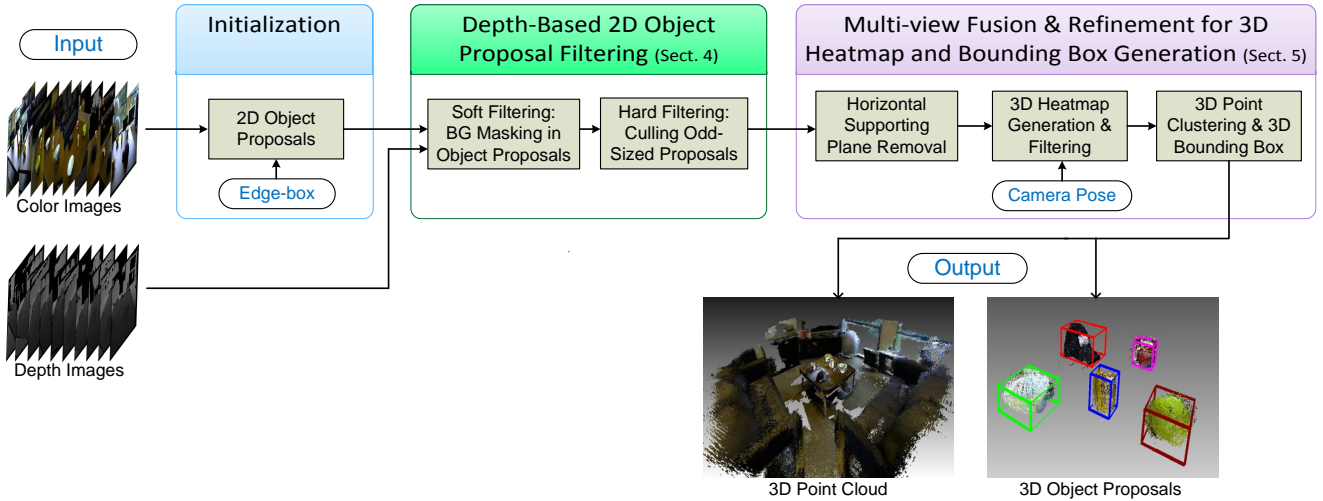


Fig. 2: Our proposed framework for 3D object proposals. We utilize a generic 2D object proposal scheme such as Edge-boxes [4] to obtain 2D object proposals. These proposals undergo robust filtering and culling based on scene geometry obtained from corresponding depth images. A horizontal supporting plane is identified and removed in each image and a 3D heatmap is sequentially obtained. Neighboring frames are fused together using camera poses to obtain 3D point cloud and a consolidated heatmap. Top 3D points are selected from the consolidated 3D heatmap based on a ranking scheme and fast 3D clustering is performed on these points to obtain 3D object proposals.

combine these two approaches and propose a framework for 3D object localization.

Pillai and Leonard [25] developed a monocular SLAM-aware system that used temporal information to output multi-view consistent object proposals using efficient feature encoding and classification techniques. They only utilize color information of the scene. Their approach works well if the objects are far apart so that the monocular SLAM method can provide a reliable semi-dense depth estimate for each view. However, their approach may fail on more crowded environments (lots of objects) or when objects are cluttered together in a small region. Using depth information enables us to handle these situations as we can exploit the 3D scene geometry to identify regions of interests, redundant vertical and horizontal planes and cluster them separately. We perform our evaluation on the same dataset for a direct comparison. We demonstrate that our results show significant improvement when compared with theirs. Another, and perhaps the most significant difference in our approach is that we use depth images along with color images to output 3D object proposals while they compute 2D object proposals only.

**Single-view RGB-D object proposals:** [26]–[28] found depth based proposals to be inferior to color based proposals. They conclude that depth and color combined together can potentially give worse results when compared to using color alone due to the inherent noise present in depth images.

Recently, a number of hybrid 2.5D approaches have been proposed, which transform depth input into more useful information such as “height above ground”, “angle with gravity” and “surface normals” [28] to be used together with color information per pixel for object proposals. Fouhey *et al.* [26] proposed an adaptive approach where they fuse depth with

color using a maximum likelihood estimation technique. They demonstrate that using a weighted scheme gave them better results than using just color or using color and depth without adaptive weights. We utilize the depth information to estimate the plane normal and horizontally planar points. This enables us to transform our 3D bounding boxes to align vertically with the global point cloud. Chen *et al.* [8], [9] introduced a MRF formulation with hand-crafted features for cars, pedestrian and cyclists detection in street scenes. They use pre-learned box templates of different sizes from trained data and use the estimated point cloud per frame to generate 3D object proposals. On the contrary, our approach is designed towards indoor scenes with no prior training or knowledge about objects we detect in the scenes. Song and Xiao [10], [11] train exemplar-SVMs on synthetic data on hundreds of rendered views. They slide these exemplars in 3D space to search for objects by densely evaluating 3D windows. They perform object detection on pre-trained objects per frame and require the gravity direction to be known. However, we leverage on scene information by aggregating information across video frames and output 3D object proposals without having any prior information about the objects present in a given scene.

### III. OVERVIEW OF THE PROPOSED APPROACH

We start by giving an overview of the proposed algorithm. At high level, the proposed algorithm is designed to fuse the depth information with the generic object proposals obtained from using color images in an indirect manner. This enables us to exploit using 3D geometry of the scene without impacting the results due to noisy or unavailable depth information at various image pixels. Figure 1 illustrates the basic setup of the problem studied in this paper and an example result obtained



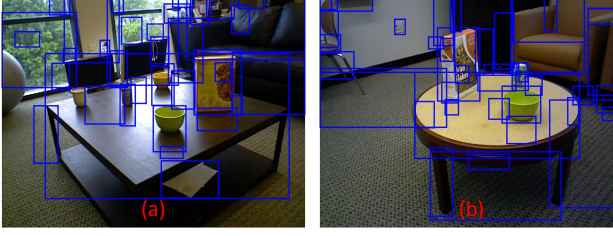


Fig. 3: Top ranked object proposals are displayed after Non-Maximum Suppression (NMS). Object proposals are displayed in decreasing order of their confidence. Object proposal with the highest confidence is displayed first. Any object proposal that shares more than 10% region is not shown. Using a simple NMS approach is unable to select the best object proposals.

by our algorithm. Our framework consists of a few novel modules in its pipeline (as shown in Fig. 2), and they will be presented in detail next. First, we introduce the initialization process in this section. Sec. IV-A presents the representation for the local 2D heatmap. Sec. IV-B and IV-C describe our depth based filtering process. Sec. V discusses how we obtain a global 3D heatmap and our final 3D object proposals.

Before presenting the main components of our approach, we first introduce the initialization process with 2D object proposals. We collect  $N$  video frames per scene using a RGB-D camera. Every  $i^{th}$  video frame consists of a color image  $I_i$ , a depth image  $Z_i$ , and the pose of the camera  $P_i$ , using a depth based SLAM method such as Dense Visual SLAM [22]. The camera pose contains the rotation and translation measurements of the camera in the world coordinate frame of reference:  $P_i = [R_i, t_i]$ . The RGB-D camera is assumed to be pre-calibrated. The camera intrinsic parameters - the focal length in  $x$  and  $y$  directions and optical center are denoted by  $f_x, f_y, [c_x, c_y]$ , respectively. Together, these are represented by the intrinsic calibration matrix  $K$ .

We use a generic 2D object proposal technique (edge-boxes [4]) to obtain  $M$  2D object proposals per frame. Edge-boxes provide a  $1 \times 5$  measurement vector per bounding box:

$$e_i^j = [x^j, y^j, w^j, h^j, c^j]_i, \quad j \in 1, \dots, M, \quad i \in 1, \dots, N. \quad (1)$$

where,  $[x, y]$  denote the top-left pixel coordinate of the bounding box, and  $w, h$  refer to the width and the height in  $x$  and  $y$  directions respectively.  $c$  corresponds to the confidence value of the bounding box which is proportional to the probability of that bounding box containing an object in it. A few top ranked 2D proposals are shown in Fig. 3. These 2D object proposals per image are treated as an input for our framework.

#### IV. DEPTH-BASED 2D OBJECT PROPOSAL FILTERING

In this section, we describe how we use depth information per frame to improve existing 2D object proposal techniques. As we only look at a single frame here, the subscript  $i$  is dropped for the clarity sake from the single view formulation.

##### A. 2D Weighted Heatmap for Pixelwise Objectness

One problem with using object proposals is the redundancy of majority of the proposals. The object proposal techniques

are usually designed for high recall and Jaccard index (IoU). This results in several object proposals in image areas where an object might be located. Object recognition techniques are used next on each of these bounding boxes to detect and identify if they contain an object. We ask a question: is it possible to quickly improve or reject some of these proposals using scene geometry without performing expensive object detection and recognition techniques?

To select the best possible proposal, techniques similar to non-maximum suppression (NMS) are generally used. Object proposals with the highest confidence locally are chosen, while any neighboring proposals that share a common region with the locally chosen proposal are rejected. However, there is no guarantee that the object proposals with higher confidence will actually be the best fitting object proposals for a given scene as shown in Fig. 3. [25] also observed that object proposals techniques often fail to find good 2D boundaries of objects present in a video scene due to motion blur. For example, none of the top 1,000 proposals in Fig. 3(a) resulted in a good bounding box for the yellow bowl ( $\text{IoU} > 0.5$ ). The technique described in this paper is able to overcome these motion blur based problems and find good 3D and 2D object proposals in such cases well.

Thus, instead of deciding right away the best 2D proposals in a given scene, a weighted approach is used to create a *heatmap*. A heatmap is a two-dimensional real-valued matrix indicating the probability of each image pixel  $[u, v]$  to be occupied by an object. We consolidate the confidence of each pixel by summing over all the object proposals  $\{e^j = [x^j, y^j, w^j, h^j, c^j]\}_{j=1}^M$  for a given image. We denote this heatmap as “baseline heatmap”  $\hat{H}$ , and it is obtained by:

$$\hat{H}[u, v] = \sum_{j=1}^M c^j \delta_o^j[u, v] \quad (2)$$

$$\delta_o^j[u, v] = \begin{cases} 1 & \text{if } u \in [x^j, x^j + w^j], \quad v \in [y^j, y^j + h^j] \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where  $c^j$  denotes the confidence of the  $j^{th}$  object proposal, and  $\delta_o^j[u, v]$  is a binary filter that checks if the pixel  $[u, v]$  is contained in the current object proposal  $e^j$ . An example of a baseline heatmap is provided in Fig. 4(c). There are various advantages of using a heatmap based approach. Firstly, it is computationally fast and extremely simple to implement. The baseline heatmap can be computed in constant time using integral images [37], regardless of the size of bounding boxes. Secondly, the heatmap provides a much more fine-grained, per-pixel objectness map rather than an array of bounding boxes per frame. This makes it easier to transfer this information to a 3D global heatmap. Thirdly, the heatmap is an accumulation of statistics computed from all bounding boxes. It turns out that sometimes no bounding box may cover an object properly. In such cases, the heatmap obtained from all these bounding boxes may result in a far more intelligent boundary of the object’s location in the scene. Thus, instead of matching bounding boxes across video frames one by one, we project the entire heatmap and consolidate the confidence values of the 3D points in the global world coordinate frame of reference.

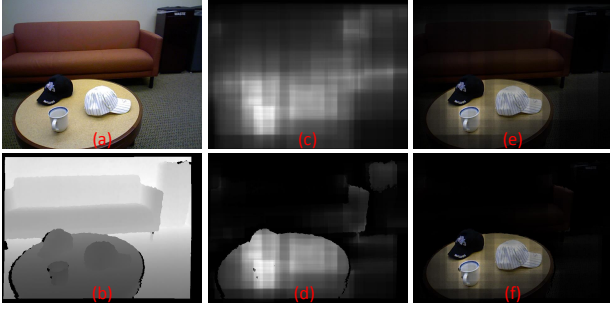


Fig. 4: (a) Color image (b) Depth image (c) Baseline heatmap  $\hat{H}$  (d) Improved heatmap  $H_{2D}$  (e) Baseline heatmap overlaid on color image (f) Our heatmap overlaid on color image. We show a comparison between the baseline heatmap  $\hat{H}$  and our improved heatmap  $H_{2D}$ . Our depth-assisted heatmap is able to filter out the background from the objects of interest using depth based statistics for every object proposal.

The baseline heatmap provides a good idea where the objects of interest might be located in the scene. However, the baseline heatmap is still rather coarse as seen in Fig. 4(c). This is because object proposals tend to enclose a lot of non-object regions such as texture on the wall and floor or sides of the table as seen in Fig. 4(e). To resolve this issue, a depth based two-step filtering process is performed to refine these initial 2D proposals: *soft filtering* and *hard filtering*.

#### B. Soft Filtering: Background Masking in Object Proposals

Soft filtering refers to making a binary decision for each pixel in an object proposal. Every pixel of an object proposal region is classified into a foreground or background pixel by performing a quick segmentation using depth information. Let  $Z_{min}$  and  $Z_{max}$  be the minimum and maximum depth range of the current object region. We compute its mean:  $Z_\mu = \frac{Z_{min} + Z_{max}}{2}$ . As our system is geared towards online applications, we avoid employing more complex segmentation techniques such as mean-shift or k-means. An object proposal is considered as containing background parts if the difference  $\Delta Z (= Z_{max} - Z_{min})$  is above a certain threshold  $\epsilon_\Delta$ .

For such object proposals containing background portions, a pixel  $[u, v]$  is classified as a background pixel if the pixel's depth value  $Z[u, v]$  is more than  $Z_\mu$  and vice versa as given in Eq. 4. The background pixels in each bounding box proposal are assigned zero confidence value while the confidence value for foreground pixels is retained:

$$\delta_s^j[u, v] = \begin{cases} 0 & \text{if } Z[u, v] > Z_\mu^j \text{ and } \Delta Z^j > \epsilon_\Delta \\ 1 & \text{otherwise,} \end{cases} \quad (4)$$

where  $\epsilon_\Delta$  is a small threshold. The thresholds are very generous and only mask off background pixels when there is a high probability for a proposal to contain the background. Figure 5 demonstrates the effect of this soft filtering method. The blue and magenta bounding boxes represent the regions that are detected to contain background parts and undergo background masking. The red and green bounding boxes represent regions with  $\Delta Z < \epsilon_\Delta$ , and do not undergo this soft filtering process as they do not contain any significant background.



Fig. 5: Depth based filtering of 2D object proposals. We only display a handful of object proposals for clarity purposes. In the soft filtering process, the blue and magenta bounding boxes are detected to contain background and undergo background suppression while the red and green bounding boxes are detected to contain no major background. After the background masking, we check the actual size of each bounding box using depth information. The red and magenta bounding boxes are identified to be either too small or too big in size or do not contain depth information. These 2D object proposals are discarded. The blue regions, after background suppression, and green regions are detected to potentially contain good information and are accepted as valid object proposals.

#### C. Hard Filtering: Culling Odd-Sized Object Proposals

Hard filtering refers to making a binary decision, accept or reject, for every object proposal of an image. The 2D bounding boxes given by object proposal techniques can vary from anywhere between  $2 \times 2$  pixels to spanning the entire image. To reduce the number of irrelevant bounding boxes, some approaches [25] ignore any small bounding boxes less than  $20 \times 20$  pixels or any relatively large bounding boxes. However, sometimes this will result in losing important information when the object is located at a distance from the camera or when the camera is zoomed onto an object momentarily.

To overcome this problem, the size statistics of each bounding box are computed using the depth information available. An object proposal is discarded if it is estimated to be outside the desired range as follows,

$$\delta_h^j = \begin{cases} 0 & \text{if } \frac{w^j * Z_\mu^j}{f_x}, \frac{h^j * Z_\mu^j}{f_y} < \epsilon_{min} \text{ or } \frac{w^j * Z_\mu^j}{f_x}, \frac{h^j * Z_\mu^j}{f_y} > \epsilon_{max} \\ 1 & \text{otherwise,} \end{cases} \quad (5)$$

where  $[\epsilon_{min}, \epsilon_{max}] = [2cm, 1m]$ , and  $w^j, h^j, Z_\mu^j$  correspond to width, height and mean depth of the foreground object in the  $j^{th}$  proposal.  $f_x$  and  $f_y$  are the known focal length parameters in the  $x$  and  $y$  direction. Specifically, any proposal whose approximate cross-section size is bigger than  $1m \times 1m$  or smaller than  $2cm \times 2cm$  is rejected. Note that a proposal of size  $3m \times 1cm$ , such as a long stick, is still considered to be a desirable object and accepted as a valid proposal. The parameters  $\epsilon_{min}$  and  $\epsilon_{max}$  are empirically set in this paper, but they can be adapted to best fit different RGB-D sensing cases. Object proposals that do not contain any depth information are also rejected. This usually happens when the regions inside these bounding boxes are too far from the depth camera. In Fig. 5, the red and magenta bounding

boxes represent proposals that are rejected based on our hard filtering process. Meanwhile, the green and blue bounding boxes represent proposals that are accepted as valid proposals.

Thereafter, the confidence for each object proposal window is updated pixel by pixel, and our improved weighted 2D heatmap is computed for every pixel  $[u, v]$  as follows:

$$\mathbf{H}_{2D}[u, v] = \sum_{j=1}^M \delta_h^j \cdot \delta_s^j[u, v] \cdot c^j \delta_o^j[u, v]. \quad (6)$$

In summary, our depth-based filtering approach reduces the computation required and improves the precision of our object proposals as the irrelevant object proposals are filtered out. A baseline heatmap (without the depth based filtering process) and our weighted heatmap are shown in Fig. 4(c,d). The objects of interest (cup and caps) stand out in our heatmap. Additionally, the background is correctly detected and assigned low confidence as shown in Fig. 4(f).

## V. MULTI-VIEW FUSION AND REFINEMENT FOR A GLOBAL 3D HEATMAP

In this section we describe our second main contribution – how a sequence of weighted 2D heatmaps  $\{\mathbf{H}_{2D}\}$  is fused in 3D using depth information and camera pose efficiently.

Let  $\mathbf{x}_w = [x, y, z]^\top$  be a 3D point in the world coordinate frame. It can be projected onto a 2D camera plane, and we denote the projected 2D point by  $\mathbf{x}_c = [u, v]^\top$  and compute it as  $\begin{bmatrix} \mathbf{x}_c \\ 1 \end{bmatrix} = \lambda \mathbf{K} \mathbf{P} \begin{bmatrix} \mathbf{x}_w \\ 1 \end{bmatrix}$ . Here  $\lambda$  is a proportionality constant, while  $\mathbf{K}$  and  $\mathbf{P}$ , as defined earlier, denote the intrinsic calibration matrix and the camera pose, respectively. For notational convenience, we represent this projection onto the camera plane as a function  $\pi$ :  $\mathbf{x}_c = \pi(\mathbf{K}, \mathbf{P}, \mathbf{x}_w)$ .

Similarly, a 2D pixel  $\mathbf{x}_c$  can be projected and transformed onto the world coordinate frame by using its depth value  $Z[\mathbf{x}_c]$ :

$$\mathbf{x}_p = \begin{bmatrix} \frac{Z[\mathbf{x}_c]}{f_x}(u - c_x) \\ \frac{Z[\mathbf{x}_c]}{f_y}(v - c_y) \\ Z[\mathbf{x}_c] \end{bmatrix}, \quad \mathbf{x}_w = \mathbf{R}^\top \mathbf{x}_p - \mathbf{R}^\top \mathbf{t}, \quad (7)$$

where  $\mathbf{x}_p$  is the 3D point in the camera frame of reference. It is then transformed to the world coordinate frame using the camera pose  $\mathbf{P}$ . We define the projection and transformation as a function  $\pi^{-1}$ :

$$\mathbf{x}_w = \pi^{-1}(\mathbf{K}, \mathbf{P}, \mathbf{x}_c, Z[\mathbf{x}_c]). \quad (8)$$

### A. Horizontal Supporting Plane Removal

After the weighted 2D heatmap  $\mathbf{H}_{2D}$  is computed for the current frame, the image pixels are projected onto 3D using Eq. 8. The 3D points are registered with the global point cloud  $\mathbf{H}_{3D} \in \mathbb{R}^{N \times 8} : [x, y, z, r, g, b, c, f]$ . The first three components  $[x, y, z]$  represent the global 3D location of the point  $\mathbf{x}_w$ . The next three components  $[r, g, b]$  represent its color.  $c$  records the consolidated heat value (confidence) of the 3D point. Finally,  $f$  denotes the number of times (frequency) that a point has been seen in RGB-D frames. We initialize the

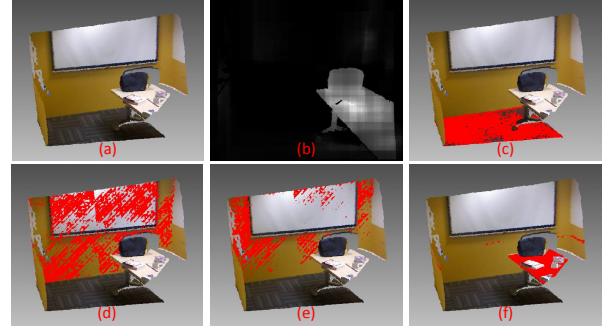


Fig. 6: Our plane removal process takes into account the heatmap of the scene. A sample 3D point cloud and its weighted heatmap is shown in (a) and (b) respectively. We display the most dominant planes in order of number of inliers in (c)-(f). The plane in (f) is selected for plane removal as its heatmap based confidence is highest among top ranked planes.

3D heatmap  $\mathbf{H}_{3D}$  for the first frame with unit frequency for all the valid image pixels.

The next task is to estimate dominant horizontal supporting planes, with the motivation that they are usually not objects of interest in a given scene. Given three 3D points  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ , a 3D plane  $\mathbf{p} = [n_x, n_y, n_z, b]^\top$  passing through these three points is computed as follows,

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \frac{(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)}{\|(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)\|_2} \\ b = -\mathbf{n}^\top \mathbf{x}_1.$$

A 3D point,  $\mathbf{x}_i$ , is considered lying on the plane  $\mathbf{p}$  if

$$\left| \mathbf{p}^\top \cdot \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right| < \epsilon_p, \quad (9)$$

where  $\epsilon_p$  is a small threshold to account for noisy data. The 3D points that lie on this plane are considered inliers and the points that exceed the threshold are considered as outliers. The inliers for the  $j^{th}$  detected plane  $\mathbf{p}^j$  are denoted as  $S^j$ .

One can classify each pixel into different segments using region classifiers and further refine them using shape priors [35]. However, such techniques are not designed to be real time. In contrast, our method quickly identifies the most dominant plane present in the heatmap using RANSAC. Three neighboring points are selected in a given frame to estimate a plane passing through these points. Then, inliers and outliers are computed for the plane using Eq. 9. This process is repeated 10,000 times to find the top five distinct dominant planes. The top planes for a sample frame in the order of decreasing number of inliers are shown in Fig. 6(c-f).

To improve the object localization, we aim to separate the objects of interest from a plane that may be present in the top ranked points. As shown in Fig. 6(b), objects of interest often lie on a supporting plane such as the table underneath the objects. This supporting plane is often contained in the 2D object proposals even after our depth based filtering process. Since a scene capture can start with a camera angle in any direction, we do not make any underlying assumptions about



the direction and location of the horizontal planes in a given scene. We observed that using the RANSAC based technique to find the plane with most inliers often resulted in selecting the horizontal plane containing the floor or vertical plane containing room walls. Due to complicated indoor lighting conditions, the table color may also vary significantly from pixel to pixel. Hence, using a color based technique may not always be successful in finding our plane of interest. In addition, it is challenging to know how many dominant redundant planes there might be in a given scene. Thus, removing all the dominant planes is also not an ideal solution.

Since we are only interested in removing the supporting plane underneath objects of interest, the obtained 2D heatmap is utilized to our advantage. Based on the inlier criterion defined in Eq. 9, we estimate the plane of interest for the first image frame by selecting the plane  $\mathbf{p}^{j^*}$  that has the highest accumulated heat value as follows,

$$j^* = \arg \max_j \sum_{i|\mathbf{x}_i \in S^j} H_{2D}(\mathbf{x}_i). \quad (10)$$

This heatmap based plane estimation assists us in consistently finding the correct plane of interest as shown in Fig. 6. For the next frame, the camera pose is used to compute the planar points by projecting onto the camera plane without needing to recompute the plane again. The pixels corresponding to the supporting plane are assigned zero confidence to obtain a filtered 2D heatmap  $\tilde{H}_{2D}$ :

$$\tilde{H}_{2D}[u, v] = \begin{cases} 0 & \forall [u, v] \in \{\pi(\mathbf{K}, \mathbf{P}, \mathbf{x}_i) | \mathbf{x}_i \in S^{j^*}\} \\ H_{2D}[u, v] & \text{otherwise} \end{cases} \quad (11)$$

The plane parameters  $\mathbf{p}^{j^*}$  are recomputed every ten frames to account for camera drift that SLAM methods often suffer from. The plane parameters are stored in a separate matrix  $\mathbf{P}^*$  to be used later. Instead of repeating the entire plane estimation process, we utilize the information of knowing where the plane lies to compute the new plane of interest quickly and efficiently. We are able to remove the correct supporting plane using this automated plane removal approach consistently. One situation where this approach fails is where a camera trajectory starts with looking at the floor and slowly pans towards the table. The floor is selected as the plane of interest in initial frames when the table and other objects are not seen in camera's field of view. Once the table occupies enough regions in the camera's view, so that its accumulated confidence is higher than the floor, the table is selected from there-on as the plane of interest. In these intermediate frames the partially seen table is not removed and thus retains its high confidence value in the heatmap. We discuss how to resolve this issue in Sec. V-D. We call the frames where we recompute plane parameters as *keyframes*. Unlike SLAM methods where keyframes are used to estimate the camera pose of the current frame, the keyframes are only used to project the plane onto the current frame, and assign the corresponding planar pixels zero confidence as shown in Fig. 7(d).

### B. 3D Heatmap Generation via Multi-view Fusion

After the filtered 2D heatmap  $\tilde{H}_{2D}$  is computed, the corresponding depth information available is used to project the points in 3D. The next step is to fuse this information with the existing 3D heatmap  $\mathbf{H}_{3D}$ .

A standard approach to fuse this information with the current 3D heatmap is to find the closest points using techniques such as ICP [31]. However, such an approach quickly becomes computationally expensive, which deviates from our goal of developing a fast approach for 3D object proposals. Another way to tackle this problem is to use the poses estimated by depth based SLAM methods to allocate points in 3D voxels or Octrees. However, due to the sheer amount of 3D points and span of the room, such an approach also requires a large amount of memory and incurs heavy computation loads. This issue is resolved by using image warping and creating a 2D index table. First, the mapping from the 2D heatmap to a 3D heatmap is initialized for the first frame. This bijective mapping, called *indexMap*, stores the location of each pixel of the current 2D heatmap in the global  $\mathbf{H}_{3D}$  heatmap.

Using Eq. 12, the previous frame's depth information is utilized to warp the image onto the current  $i^{th}$  frame. We round the projected pixel location to the nearest integer and compare their depth and color values per pixel. Let us assume that pixel  $\tilde{\mathbf{x}}_c$  of the previous frame is warped onto  $\mathbf{x}_c$  of the current frame.

$$\mathbf{x}_c = \pi(\mathbf{K}, \mathbf{P}^i, \pi^{-1}(\mathbf{K}, \mathbf{P}^{(i-1)}, \tilde{\mathbf{x}}_c, \mathbf{Z}^{(i-1)}[\tilde{\mathbf{x}}_c]) \quad (12)$$

Based on this information, we compute the difference in intensity and depth at each matched pixel:

$$\Delta \mathbf{I}^i[\mathbf{x}_c] = \|\mathbf{I}^i[\mathbf{x}_c] - \mathbf{I}^{(i-1)}[\tilde{\mathbf{x}}_c]\|_2 \quad (13)$$

$$\Delta \mathbf{Z}^i(\mathbf{x}_c) = |\mathbf{Z}^i[\mathbf{x}_c] - \tilde{\mathbf{Z}}^{(i-1)}[\tilde{\mathbf{x}}_c]|, \quad (14)$$

where  $\tilde{\mathbf{Z}}^{(i-1)}[\tilde{\mathbf{x}}_c]$  denotes warped depth value in the current frame. If projected pixel's color and depth information is within a threshold ( $\epsilon_I, \epsilon_Z$ ) of the current pixel's information, the two corresponding pixels are considered a match and the index is copied to the current pixel. In case where more than one pixel from previous frame matches a pixel in the current frame, then the pixel corresponding to a lower warped depth value (foreground) is chosen as the matching pixel. This matching is shown in Fig. 7(b). If no pixel from the previous frame matches the current pixel, the pixel is identified as a new point and added to the global 3D heatmap along with the current *indexMap*. Since this approach requires depth information at each pixel, if the depth information is not available or contains noisy depth, this can lead to wrongly matched pixels. We choose to ignore these pixels as our primary goal is obtaining fast and efficient 3D proposals, and performing this step for every image can incur more computational load. Nevertheless, if more robust results are required, one can fill the holes using a nearest-neighbor method or an adaptive filter using the corresponding color images [34].

For all matched points, the original 3D point's confidence value is increased by the current matched pixel's confidence value and the counter is incremented by a unit. In addition, the color and location of the 3D point is adjusted by a

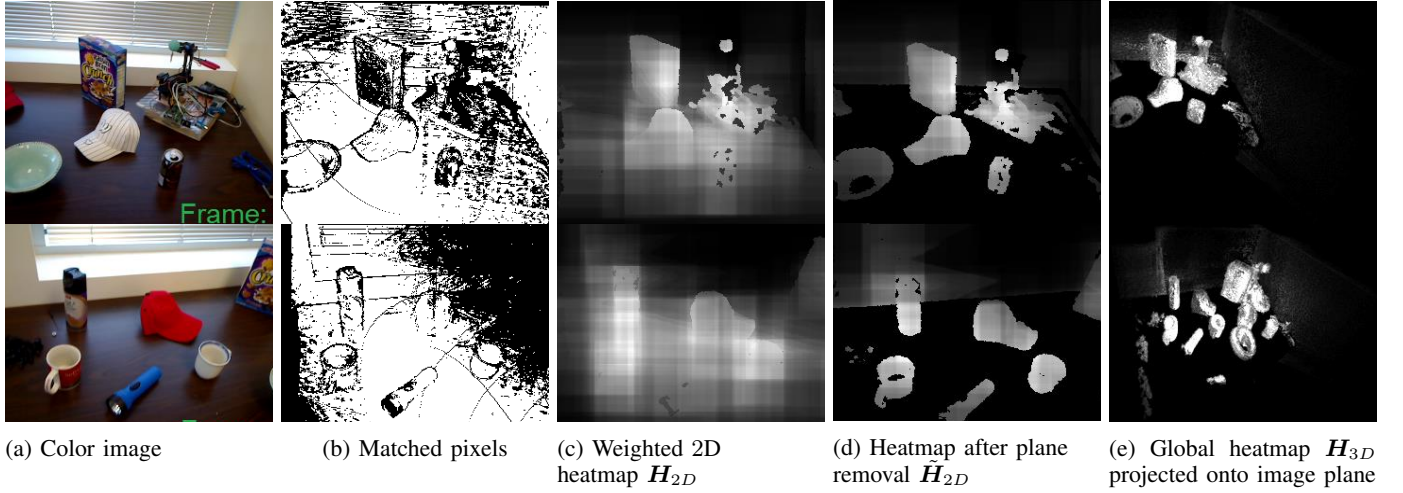


Fig. 7: Left column displays different RGB frames from *table\_1* scene in [39].  $2^{nd}$  column shows the matched pixels of the current frame with the previous frame.  $3^{rd}$  column highlights our weighted 2D heatmap after depth based filtering.  $4^{th}$  column displays our refined heatmap after plane removal. Finally, the  $5^{th}$  column shows the current global heatmap of the entire scene. The global heatmap accumulates the confidence of each point in the scene as the camera moves around in the scene.

weighted average of the current 3D location of the pixel and the global location of the matched point. This accounts for any minor drift error that might occur while estimating the camera trajectory in an indoor environment.

### C. 3D Heatmap Filtering using Average Confidence Measure

Once our weighted 3D heatmap is obtained, any points that are seen less than five times or 5% of the total number of frames, whichever is lower, are identified as unwanted points and discarded. We further use a metric: pseudo-average confidence to rank the global 3D points. The pseudo-average confidence of the  $i^{th}$  point is computed as:

$$\bar{c}(i) = \frac{H_{3D}(i; c)}{H_{3D}(i; f) + \tau}, \quad i = 1, \dots, N, \quad (15)$$

where  $H_{3D}(i; c)$  denotes retrieving the  $c$  (confidence) element of the  $i^{th}$  point stored in the 3D heatmap  $H_{3D}$ , similarly for  $H_{3D}(i; f)$  that returns the  $f$  (frequency) element.  $\tau$  is a constant introduced to gently favor points seen more often.

The pseudo-average confidence ranks points seen more often higher than points seen less often. Intuitively, this makes sense as we should see the 3D points lying on objects of interest more often than other points. Only those points are retained that have  $\bar{c}(i) \geq \epsilon$ . This ensures that we obtain a good precision while maintaining an acceptable recall value. If a better precision or recall is desired, one can increase or decrease this threshold respectively. An example of the top ranked 3D point cloud is shown in Fig. 8(a).

### D. 3D Point Clustering and 3D Bounding Box Generation

As discussed in Sec. V-A, it is possible that the scene capture starts without the objects and table in the camera's field of view. The scene capture may start with room floor or walls in its entire field of view and then move to the area of interest. In cases where the table is only partially seen (a

few pixels) when the camera field of view is moving towards the objects of interest, the accumulated heatmap value of the planes corresponding to walls and floors may be temporarily higher in a few image frames. In such cases, the floor and walls will be removed from heatmap but the table will be left untouched. Thus, some outer regions of the table may still be present in our final ranked points as seen in Fig. 8(a). To remove these unwanted points, the plane parameters stored in the matrix  $P^*$  obtained from our previous keyframes are used. The plane parameters for each keyframe may correspond to the floor or vertical wall when the table is not seen. Thus, we first identify different planes that were removed using K-means (walls, rooms, table etc.) For each unique plane, the best fitting plane is estimated using these different plane parameters. Thereafter, a final plane removal step is performed on our top ranked points by finding any points that satisfy Eq. 9 for the estimated plane. The final filtered 3D point cloud is shown in Fig. 8(b).

After plane removal, we perform density based clustering using density-based spatial clustering of applications with noise (DBSCAN) [38] is performed on our filtered top ranked points. DBSCAN groups together points that are closely packed and marks points as outliers that lie alone in low-density regions. This allows us to reject any points that may belong to uninteresting regions such as walls and table that may exist in our top filtered points. The advantage of using a density based approach over other techniques such as K-means is that we do not need to specify the number of clusters that are present in our filtered data. Thus, depending on the scene complexity, this approach can automatically select the relevant number of regions of interest in the scene. DBSCAN can run in an overall average runtime complexity of  $O(n \log(n))$ . An example of DBSCAN based clustering is shown in Fig. 8(c).

After density based clustering, a tight 3D bounding box is estimated for each cluster. However, finding a good bounding box is challenging in the camera's frame of reference, since



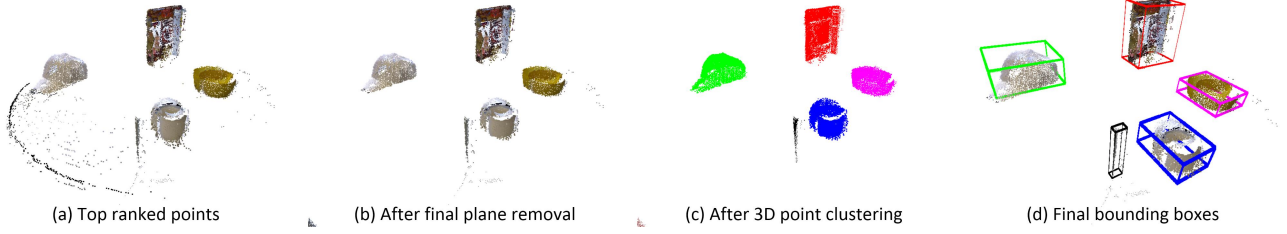


Fig. 8: Once we obtain the top ranked points, we perform a final plane removal step to remove any leftover supporting plane points. We perform density-based spatial clustering to find the most dominant point clusters and estimate their bounding boxes.

the horizontal direction (parallel to the floor) may not be the same as the camera’s horizontal direction. First a normal to the supporting horizontal plane,  $\mathbf{n}_c$ , is estimated by using the entire 3D point cloud obtained after our data collection as explained in Sec. V-A. Thereafter, each cluster is transformed to the world coordinate frame where the orthogonal directions  $X$ ,  $Y$  and  $Z$  match the standard normal vectors  $[1, 0, 0]^T$ ,  $[0, 1, 0]^T$  and  $[0, 0, 1]^T$  respectively using Eq. 16. Thereafter, we find the minima and maxima in each of the three orthogonal directions and draw a bounding box around the clusters. This bounding box is then transformed back to the original coordinate frame.

$$\begin{aligned} \mathbf{v} &= \mathbf{n}_c \times \mathbf{n}_w, & s &= \|\mathbf{v}\|_2, & c &= \mathbf{n}_c \cdot \mathbf{n}_w, \\ \mathbf{R} &= \mathbf{I} + [\mathbf{v}]_{\times} + \frac{(1-c)}{s^2} [\mathbf{v}]_{\times}^2 \end{aligned} \quad (16)$$

where  $\mathbf{n}_c$  denotes the normal to the detected horizontal supporting plane in the camera’s coordinate frame, and  $\mathbf{n}_w = [0, 1, 0]^T$  is the world coordinate’s normal direction.  $[\mathbf{v}]_{\times}$  is the skew symmetric matrix of the vector  $\mathbf{v}$ . If the bounding boxes of neighboring clusters intersect, we combine those clusters together. This usually happens when an object breaks into two sub-parts due to pose estimation errors or in the presence of specular objects such as a soda can. Any bounding boxes that have a volume of less than  $1\text{cm}^3$  are rejected as these small patches are usually a part of the walls or floor plane. The final refined 3D point cloud with the respective bounding boxes are shown in Fig. 8(d). This modular approach assists in finding tight bounding boxes in the direction of gravity for each object of interest.

## VI. EXPERIMENTAL EVALUATION

We use four datasets to conduct our experimental analysis and evaluation: object segmentation dataset [30], UW-RGBD scene dataset [29], RGBD scene dataset 2011 [39], and our own dataset.

The object segmentation dataset (OSD) [30] consists of 111 labeled RGBD images in six subsets. The images vary from having two small objects on a table to containing more than ten objects stacked side by side and on top of each other. The dataset provides pixelwise labeling of each instance of the objects. We use these labeled images to create 2D ground-truth bounding boxes for each object present in the dataset to enable learning and for precision-recall evaluation. The depth images are pre-aligned with the color images. We primarily used this dataset to train our parameters.

The UW-RGBD scene dataset [29] contains 14 scenes reconstructed from RGB-D video sequences containing furniture and some table-top objects such as caps, cereal boxes and coffee mugs. The scenes contain depth and color frames from a video collected by moving around the scene. The dataset provides a globally labeled 3D point cloud. We used Dense Visual SLAM [22] to obtain the camera pose per frame to fuse the frames together. Our 3D point cloud is slightly misaligned with the ground-truth point cloud by a few  $\text{mm}$ . It is not trivial to align these point clouds due to different error characteristics of the point clouds. Thus, for our current evaluation, we decided to ignore this mismatch as the misalignment is marginal. We primarily use this dataset for our 3D evaluation and comparison with existing state of the art techniques in 2D by projecting our bounding boxes on the image plane.

The RGBD scene dataset 2011 [39] consists of eight realistic environment scenes of a lab, kitchens and office rooms. The objects of interest are placed on kitchen slabs and tables in a room. The dataset provides ground-truth 2D bounding boxes for various objects of interest such as soda cans, caps, and flashlight. However, some objects such as laptops, computer mice and other kitchen equipment are labeled as background in this dataset. The authors used depth and color images to segment and identify the objects present in each frame independently. We use this dataset to show our results in a more cluttered environment, and compare our results in 2D by projecting our bounding boxes on the image plane and comparing with the 2D ground-truth bounding boxes.

The existing RGBD datasets are intended for benchmarking category-independent object segmentation and identification purposes, and thus only provide limited test cases. The effect of errors in SLAM algorithms is also ignored. Therefore, we collected our own sequence with more challenging situations where the scene is crowded and the objects are placed in proximity. We also take into consideration of imperfect SLAM and demonstrate that our method is only marginally affected given inaccurate camera poses estimation. We plan to make our dataset public in future.

### A. Evaluation of Single-frame 2D Object Proposals

**OSD.** We first evaluate results on the object segmentation dataset (OSD) [30] to demonstrate how our depth based filtering improves edge-boxes [4]. We use this dataset to train our parameter  $\epsilon_{\Delta}$  in Eq. 4. We used 1000 object proposals per

image on the training subset and found  $\epsilon_{\Delta} = 0.5m$  to give us the best results on the test dataset as well. Using depth based 2D object proposal filtering as described in Sec. IV, we are successful in rejecting 5.1% of the object proposals provided by edge-boxes across 111 images.

From here on, the different parameters discussed in Sec. IV and V are fixed as :  $\{\epsilon_{\Delta}, \epsilon_{min}, \epsilon_{max}, \tau\} = [0.5m, 2cm, 1m, 10]$  and  $\{\epsilon_p, \epsilon_I, \epsilon_Z, \epsilon\} = [0.005, 0.05, 0.01, 0.25]$ .

**UW-RGBD scene dataset.** We also computed the acceptance and rejection rate on the UW-RGBD dataset for our depth based filtering process. Based on the background masking and culling odd-sized proposals as described in Sec. IV-B and IV-C, 7% object proposals are rejected, 31.3% are fully accepted and 61.7% object proposals undergo partial filtering to mask the background on the entire dataset.

**RGBD scene dataset 2011.** We use the ground-truth 2D bounding boxes provided in the RGBD scenes dataset to report the average precision, recall and success rate. As here we treat and evaluate each frame independently, we do not perform multi-view fusion as described in Sec. V. Instead, after depth based filtering, we remove the unwanted supporting plane in 3D (Sec. V-A) and cluster the 3D points to compute 3D bounding boxes (Sec. V-D). We project the 3D points inside each of the 3D bounding boxes back to the image plane, and compute the 2D bounding boxes around these pixels. As noted earlier, most 2D object proposal techniques aim for a high recall. In contrast, our goal is fast precise 3D object proposals, we typically obtain 5-20 object proposals depending on the scene complexity.

Let  $BB_g(i)$  be the  $i^{th}$  ground-truth 2D object proposal,  $BB_e(j)$  and  $BB_o(j)$  be the  $j^{th}$  edge-boxes and our 2D object proposals, respectively. Let  $M_e$  and  $M_o$  be the total number of object proposals computed by edge-boxes and our method. First, we use the standard definition of Intersection over Union (IoU):

$$IoU(i) = \max_j \left( \frac{BB_g(i) \cap BB_o(j)}{BB_g(i) \cup BB_o(j)} \right) \quad \begin{matrix} \forall j \in 1, \dots, M \\ \forall i \in 1, \dots, N \end{matrix} \quad (17)$$

where  $M$  and  $N$  are the total number of output object proposals and ground-truth object proposals respectively in a given scene. Detection rate (DR), sometimes also referred to as average recall is defined as:

$$DR = \frac{1}{N \cdot K} \sum_{k=1}^K \sum_{i=1}^N (IoU(i; k) \geq 0.5) , \quad (18)$$

where  $K$  is the total number of scenes in the dataset and  $IoU(i; k)$  refers to the IoU for the  $i^{th}$  ground-truth object proposal in the  $k^{th}$  scene. We define a modified Intersection over Union ( $IoU_o$ ) for the  $j^{th}$  object proposal as follows,

$$IoU_o(j) = \max_i \left( \frac{BB_g(i) \cap BB_o(j)}{BB_g(i) \cup BB_o(j)} \right) \quad \begin{matrix} \forall j \in 1, \dots, M \\ \forall i \in 1, \dots, N \end{matrix} \quad (19)$$

The difference between our modified IoU and standard IoU is that we estimate the best intersection per output object proposal, while the standard IoU is computed per ground-truth object proposal. Our definition heavily penalizes any redundant object proposals. We obtain a zero IoU score for

Num. of proposals	Edge-boxes [4]		Ours	
	SR	DR	SR	DR
50	0.21	0.72	0.66	0.80
100	0.16	0.78	0.67	0.83
500	0.08	0.88	0.69	0.86
1000	0.05	0.89	0.67	0.86
2000	0.03	0.90	0.67	0.86

TABLE I: Comparison between edge-boxes [4] and our technique per frame on the RGBD scene dataset 2011 [39]. SR and DR refer to success rate and detection rate respectively.

object proposals that do not intersect with ground-truth object proposals. Based on this definition, similar to Eq. 18, we define Success Rate (SR) as:

$$SR = \frac{1}{M \cdot K} \sum_{k=1}^K \sum_{j=1}^M (IoU_o(j; k) \geq 0.5) , \quad (20)$$

where  $K$  is the total number of scenes in the dataset and  $IoU_o(j; k)$  refers to our modified IoU for the  $j^{th}$  object proposal on the  $k^{th}$  scene. Success rate can also be interpreted as signal to noise ratio (SNR) and has been used previously in [40] for 2D object proposal analysis. We use different numbers of input 2D edge-boxes proposals varying from 50 to 2,000 and report our results in Table I.

On average we obtained 6.57 output 2D object proposals for our method. As shown in Table I, using scene geometry enables us to significantly improve edge-boxes [4]. We show remarkable improvement in success rate (SR). Our success rate is stable regardless of the number of input object proposals while the success rate drops drastically for edge-boxes. This is because the object proposals are chosen in order of their confidence values: the more number of input object proposals, the less likely they are to contain an object of interest in them. Our detection rate (DR) is higher than edge-boxes when using less numbers of proposals. It is slightly less than edge-boxes when using 1,000 or more proposals. This is due to occluded objects present in the scene. Since this analysis is frame independent, our method only looks at the current frame and finds tight bounding boxes around objects seen in the frame currently. With a higher number of input proposals, edge-boxes is able to propose 2D bounding boxes for these fully or partially occluded objects which increases its detection rate. We are able to overcome this issue by leveraging on multi-view information, and also our ultimate goal is not per-view detection rate.

## B. Evaluation of 2D and 3D Object Proposals from Multi-view Frames

**UW-RGBD scene dataset.** After we obtain a 3D heatmap per frame, we use multi-view information to fuse this information together. We use the pose estimated by Dense Visual SLAM [22]. It uses depth images along with color information to create a globally consistent 3D map and outputs a camera pose per frame. However, like other real-time SLAM algorithms that do not perform expensive optimization of camera poses and 3D point clouds, it also suffers from camera drift errors. Our algorithm is designed such that it is tolerant

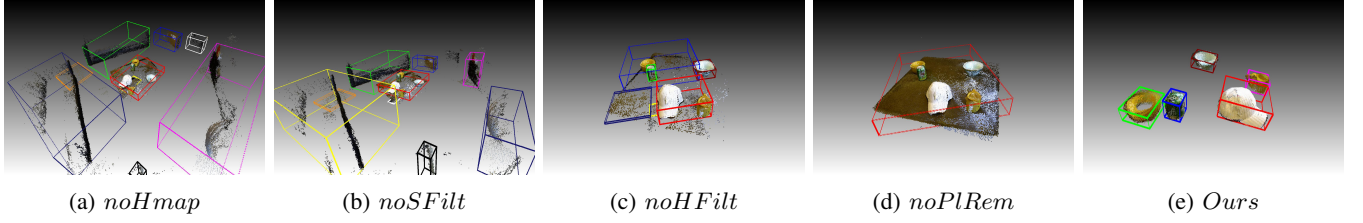


Fig. 9: We show final results for a sample scene in the UW-RGBD scene dataset for our approach after removing one component. We obtain results in (a) without depth based filtering as described in Sec. IV. Results in (b) and (c) are obtained when background masking and bounding box rejection are not performed, respectively. (d) shows results without plane removal. (e) shows our result when all of these components (background masking, bounding box rejection, and plane removal) are used.

Method	VGA	$\downarrow 2$
Overall	<b>3.03s</b>	<b>0.973s</b>
EdgeBoxes [4]	0.251s	0.078s
Depth based filtering	0.373s	0.195s
Plane removal	0.14s	0.06s
Allocating 3D points:		
Confidence and frequency	0.506s	0.132s
Location and color	1.481s	0.359s

TABLE II: Analysis of average run-time performance per frame of our approach. Our experiments were conducted on a single-core Intel Xeon E5-1620 CPU.

Method Used	2D				3D			
	$IoU$	$IoU_o$	SR	DR	$IoU$	$IoU_o$	SR	DR
<i>noHmap</i>	0.32	0.19	0.17	0.30	0.17	0.07	0.01	0.19
<i>noSFilt</i>	0.51	0.25	0.29	0.60	0.33	0.15	0.13	0.31
<i>noHFilt</i>	0.77	0.47	0.60	<b>1.00</b>	0.53	0.32	0.35	0.53
<i>noPlRem</i>	0.13	0.10	0.02	0.02	0.03	0.02	0.00	0.00
<i>Ours</i>	<b>0.78</b>	<b>0.50</b>	<b>0.62</b>	<b>1.00</b>	<b>0.55</b>	<b>0.34</b>	<b>0.45</b>	<b>0.66</b>

TABLE III: Comparison of 2D and 3D average IoU, modified IoU, success rate and detection rate for objects of interest present in the UW-RGBD dataset. We report the benefits of each of our contributions: background suppression, rejection of odd-sized bounding boxes, and supporting plane removal.

to small drift and noisy 3D point clouds. We generate object proposals at the same time as SLAM, and do not require the whole sequence to be captured first.

The entire process takes 3.03s for VGA resolutions on average in MATLAB. However, a majority of the time ( $> 2.0s$ ) is spent in storing and accessing the 3D point cloud. A detailed time evaluation of various techniques used in our approach is reported in Table II. As we aim for a fast and efficient algorithm that is capable of online processing of 3D object proposals, we downsample the images by 2. This reduces the time taken per frame to less than one second in MATLAB on a single-core CPU. We demonstrate our results after downsampling in Fig. 11.

First, we show the average IoU, both in 2D and 3D, obtained per object of interest for the UW-RGBD scene dataset [29] to demonstrate the effectiveness of each of our contributions in Table III. We estimate the 2D bounding boxes by projecting the 3D points inside each bounding box onto the image plane and computing a 2D bounding box around it. We repeat this procedure to estimate ground-truth 2D bounding boxes as well by using the ground-truth labeled point cloud. The left column in Table III represents one step that was skipped to estimate the 3D bounding boxes. *noHmap* refers to our approach without using the weighted heatmap  $H_{2D}$ . In this approach, we skipped both soft filtering and hard filtering as described in Sec. IV. *noSFilt* computes the results without performing background masking using depth information, and *noHFilt* refers to our approach without rejecting odd-sized bounding boxes that are mainly part of the background. *noPlRem* computes our results without performing plane removal per frame as described in Sec. V-A. Finally, *Ours* refers to our method that integrates all the steps together to compute the 3D object

proposals. We observe that each step is vital in improving the accuracy of our algorithm. A good supporting plane estimation plays an important role in removing the table underneath the objects for tight bounding box estimation. We show the final top ranked points along with the estimated bounding boxes in Fig. 9. We also compare our results with [25]. Our approach considers chairs as objects of interest as seen in Fig. 11(b). However, since [25] treat chairs as background objects, we also leave them out of our analysis for a fair comparison in Fig. 10. Once we obtain the 3D bounding boxes, we compute all the points that lie inside the current bounding box and project them onto the current image plane. Thereafter, we estimate the 2D bounding box that surrounds these projected points. We repeat this procedure for the ground-truth bounding boxes and compute the standard IoU using Eq. 17. Our 2D proposals (with and without downsampling) consistently outperform [25]. In fact, 3D intersection is extremely sensitive to noise. A small misalignment of 12.6% in each of three orthogonal directions reduces the 3D IoU to below 0.5. While our recall rate is good for low 3D IoU, it drops below 0.5 quickly due to the small mismatch between our point cloud and the ground-truth point cloud.

Like [29], we also use the ground-truth labeling per point to obtain precision-recall results for our 3D object proposals. Let us consider the ground truth points labeled as soda can, bowl, cap, chair as points of interest ( $\hat{P}$ ) and the background such as sofa, table, floor as redundant points ( $\hat{N}$ ). We overlay our 3D bounding boxes on the ground-truth point cloud and check if a point of interest  $i$  lies inside or outside our proposals. We consider the points of interest that lie inside our object proposals as True Positives ( $TP$ ). The redundant points that



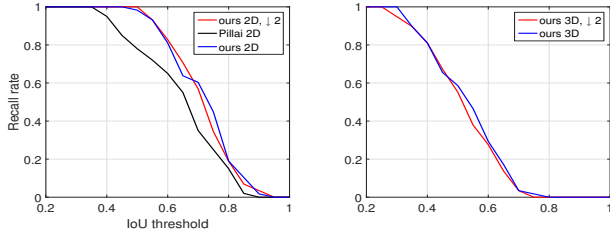


Fig. 10: Our 2D and 3D recall rate with varying threshold for IoU on the UW-RGBD dataset. Our 2D proposals comfortably outperform [25]. As we exploit scene geometry, downsampling images by 2 does not have an impact on our recall rate.

lie inside our object proposals False Positives ( $FP$ ):

$$TP = \sum_{i=1}^{N_g} (\hat{P}(i) \in BB_o(j)) \quad \forall j \in 1 \dots M \quad (21)$$

$$FP = \sum_{i=1}^{N_g} (\hat{N}(i) \in BB_o(j)) \quad \forall j \in 1 \dots M, \quad (22)$$

where  $N_g$  refers to the number of ground-truth points in a given scene.  $TP$  and  $FP$  are used to compute the average precision (AP), average recall (AR) and F-measure.

We compare our object proposals with state-of-the-art segmentation and classification results reported on the UW-RGBD dataset in Table IV. Even for 3D P-R measurements, we obtain the best mean AP, while maintaining an acceptable recall. The ground-truth labeling considers objects such as desktops as part of the background. This decreases our average precision as these detected objects are considered as false positives. If we ignore the scenes 13 and 14 in the dataset where we see these objects, our average precision increases from 93.46 to 98.76. Another issue affecting our recall performance is the presence of chairs. The camera's field-of-view is concentrated mainly on the objects on the scene, and hence the chair points (especially those under the seat) are not seen frequently and are filtered out by our technique. This reduces our recall rate if we consider chairs as objects of interest. We also compute our recall without chairs as objects of interest and are reported in Table IV. Our 3D recall is lower than [29] because of the small misalignment between our point cloud and the ground-truth point cloud. This misalignment results in some points of interest ( $\hat{P}$ ) that are not included in our true positives ( $TP_k$ ), yielding an artificially lower recall rate.

Our technique is agnostic about the number of objects that may be present in the scene and hence, can scale well for crowded scenes that contain lots of objects. We obtain on average 6.57 3D object proposals per scene. Our results can be further improved with an improved camera pose estimation, as in some cases the objects break into two or more discontinuous point clusters due to noisy camera poses. This results in multiple distinct object proposals for one object, essentially dividing the object into two or more pieces.

**RGBD scene dataset 2011.** We use RGBD scenes as well to show our results in a more cluttered environment. RGBD scenes contain seven different scenes ranging from a lab to a kitchen. We ignore one scene - *desk2* as the camera pose

Method	Run-time	AP	AR	F-measure
DetOnly [32]	1.8s	61.7	81.9	70.38
HMP2D+3D [29]	4.0s	92.8	<b>95.3</b>	<b>94.03</b>
Pillai & Leonard [25]	1.6s	81.5	59.4	68.72
Ours	3.03s	93.46	76.19	83.95
Ours w/o chairs,desktop	3.03s	<b>98.76</b>	81.66	89.40
Ours(↓ 2)	<b>0.97s</b>	91.91	78.49	84.67
Ours(↓ 2) w/o chairs,desktop	0.97s	96.95	82.86	89.35

TABLE IV: Analysis of our 3D object proposals on the UW-RGBD scene dataset in comparison to [32], [29], and [25]. We achieve efficient run-time performance if we downsample the data for our analysis. Our experiments were conducted on a single core Intel Xeon E5-1620 CPU.

obtained gave unsatisfactory results. We compare our results in 2D with [39] in Fig. 12. We make an underlying assumption that our precise 3D object proposals can be correctly identified as the relevant objects inside those bounding boxes. As the authors perform object detection, we manually measure the precision and recall for each object individually by using the ground-truth labeling per point as discussed previously. Our performance is significantly better than [39]. We also observed that our technique picks up other objects such as laptop, computer mouse, and kitchen devices as objects of interest, which are treated as background in their analysis.

**Our dataset.** We collected our own dataset using Microsoft Kinect v1. We captured six video sequences varying from 400 frames to 800 frames. Some of the scenes are shown in Fig. 11. We placed multiple objects such as a kettle, books, coffee mugs and a messenger bag in various locations in an indoor room. We purposely placed these objects close to each other to showcase that our approach can cluster these nearby objects separately. We also placed coffee mugs on a whiteboard attached to the wall (second scene from right) to highlight that our approach is also able to recognize them as objects of interests despite not being placed on the table.

Our approach is able to identify small objects such as power bank lying on the table consistently. The plane removal step assists in segmenting out the books placed flat on the table. This would be extremely difficult to do in the global point cloud where the table 3D points may not lie on a unit plane due to depth noise and SLAM errors. However, due to the presence of depth noise, some table pixels are not filtered out, and become a part of the nearby objects. This leads to clustering two neighboring objects together as seen in Fig. 11(e), where the power bank and books are clustered together as one object.

## VII. CONCLUSION

In this paper, we have developed a novel multi-view based 3D object proposal technique using depth information along with initial 2D proposals. To our knowledge, this is the first technique that truly produces 3D object proposals without using trained object segmentation and recognition classifiers. In future work, we aim to optimize our system towards real-time 3D object proposals over even larger environments by exploring multi-scale representations for memory and computational efficiency. Ultimately, we intend to integrate our system with SLAM to improve its accuracy by treating the object proposals as fixed landmarks in the scene.

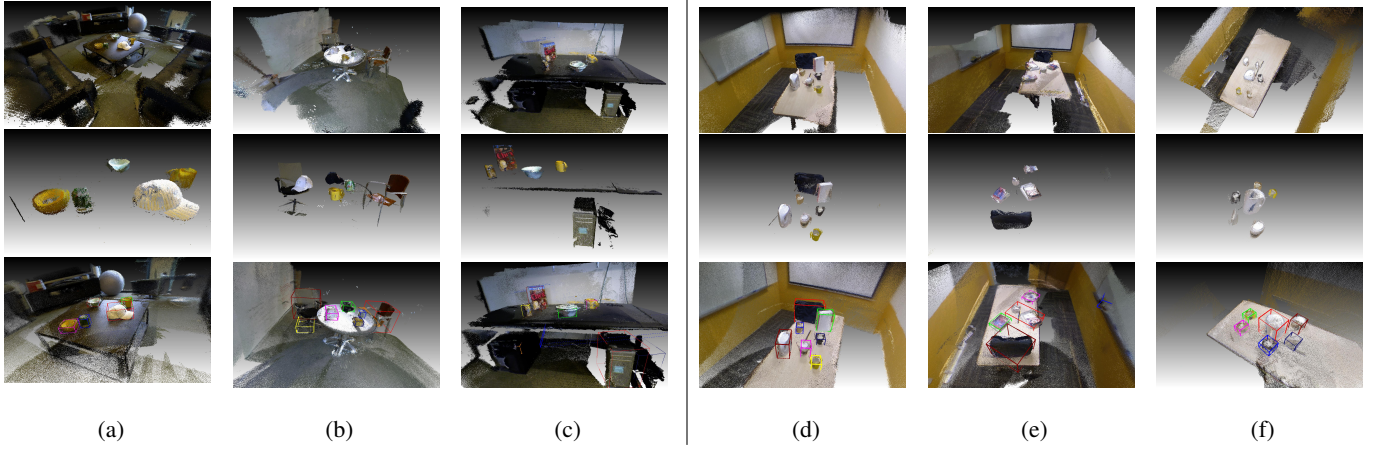


Fig. 11: Our 3D Object proposals for the UW-RGBD dataset (first three columns) and our dataset (last three columns). First row shows the point cloud of the different scenes. Second row shows our top ranked filtered points and finally third row highlights the resulting 3D object proposals using density-based clustering. Our technique also picks up objects like desktop PC and chairs that are considered as background for most object segmentation algorithms.

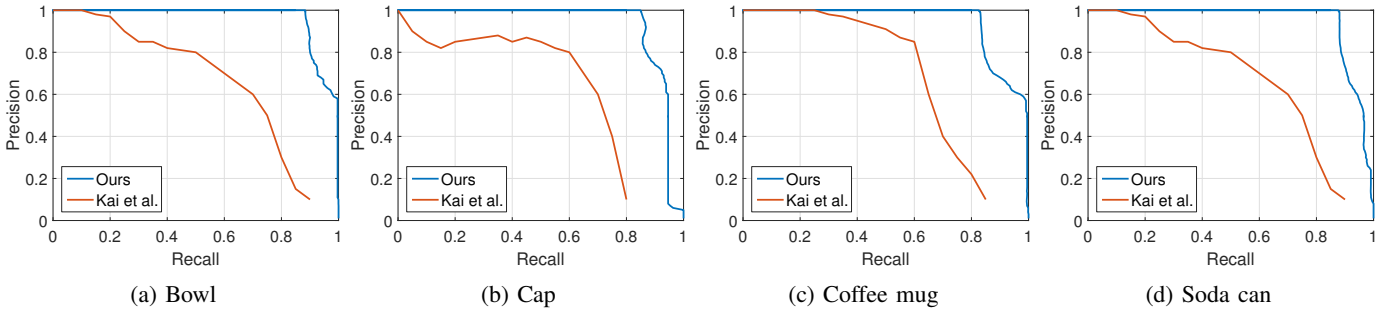


Fig. 12: We compare P-R measurements for the four objects in the RGBD scenes dataset 2011. We projected our 3D bounding boxes onto the image plane to obtain our 2D bounding boxes and used the ground-truth bounding boxes to obtain these results.

## REFERENCES

- [1] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* 2014, pp. 3286–3293. [1](#), [2](#)
- [2] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Jul 2012, pp. 1312–1328. [1](#), [2](#)
- [3] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2014. [1](#), [2](#)
- [4] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision (ECCV)*, Sep 2014. [1](#), [2](#), [3](#), [4](#), [9](#), [10](#), [11](#)
- [5] P. Krhenbhl and V. Koltun, "Geodesic object proposals," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 725–739. [1](#), [2](#)
- [6] D. Oneta, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-Temporal Object Detection Proposals," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 737–752. [1](#), [2](#)
- [7] S. Song and M. Chandraker, "Joint sfm and detection cues for monocular 3D localization in road scenes," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3734–3742. [1](#), [2](#)
- [8] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D Object Proposals using Stereo Imagery for Accurate Object Class Detection" in *arXiv preprint arXiv:1608.07711*, 2016. [3](#)
- [9] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals for accurate object class detection" in *IEEE International Conference on Neural Information Processing Systems (NIPS)*, pp. 424–432, 2015. [3](#)
- [10] S. Song, and J. Xiao, "Deep Sliding Shapes for amodal 3D object detection in RGBD images," *arXiv preprint arXiv:1511.02300*, 2015. [3](#)
- [11] S. Song, and J. Xiao, "Sliding shapes for 3D object detection in depth images," in *European Conference on Computer Vision (ECCV)*, pp. 634–651, 2014. [3](#)
- [12] G. Yahav, G. Iddan, and D. Mandelbom, "3D imaging camera for gaming application". In *Digest of Technical Papers of International Conference on Consumer Electronics (ICCE)*, Jan 2007, pp. 1–2. [1](#)
- [13] A. D. Wilson, "Using a depth camera as a touch sensor," in *ACM International Conference on Interactive Tabletops and Surfaces*, 2010. [1](#)
- [14] Y. Song, F. Liu, and S. Yan, "Body Surface Context: A New Robust Feature for Action Recognition From Depth Videos," in *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2014, pp. 952–964. [1](#)
- [15] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," in *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013. [1](#)
- [16] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 1697–1702. [1](#)
- [17] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, "Interactive 3D modeling of indoor environments with a consumer depth camera," in *International Conference on Ubiquitous Computing (UbiComp)*, 2011, pp. 75–84. [1](#)
- [18] L. Bo, X. Ren and D. Fox, "Depth kernel descriptors for object recognition," in *IEEE/RSS International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 821–826. [1](#)
- [19] W. Kuo, B. Hariharan, and J. Malik, "Deepbox: Learning objectness with convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2479–2487. [1](#), [2](#)

- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *IEEE International Conference on Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99. [2](#)
- [21] K. Xu, H. Huang, Y. Shi, H. Li, P. Long, J. Caichen, W. Sun, and B. Chen, "Autoscanning for coupled scene reconstruction and proactive object analysis," in *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)*, Oct 2015, pp. 177:1–177:14. [1](#)
- [22] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for RGB-D cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 2100–2106. [4](#), [9](#), [10](#)
- [23] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *arXiv preprint arXiv:1502.05082*, 2015. [2](#)
- [24] S. Choudhary, A. J. Trevor, H. I. Christensen, and F. Dellaert, "SLAM with object discovery, modeling and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 1018–1025. [2](#)
- [25] S. Pillai and J. Leonard, "Monocular slam supported object recognition," in *Proceedings of Robotics: Science and Systems (RSS)*, Jul 2015. [3](#), [4](#), [5](#), [11](#), [12](#)
- [26] D. F. Fouhey, A. Collet, M. Hebert, and S. Srinivasa, "Object recognition robust to imperfect depth data," in *European Conference on Computer Vision (ECCV) Workshops: Consumer Depth Cameras for Computer Vision*, 2012, pp. 83–92. [3](#)
- [27] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3D object dataset: Putting the Kinect to work," in *IEEE International Conference on Computer Vision (ICCV) workshop*, 2013, pp. 141–165. [3](#)
- [28] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *European Conference on Computer Vision (ECCV)*, 2014. [3](#)
- [29] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. [9](#), [11](#), [12](#)
- [30] A. Richtsfeld, "The Object Segmentation Database (OSD)," 2012. Available: [9](#)
- [31] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2001, pp. 145–152. [7](#)
- [32] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3D scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1330–1337. [12](#)
- [33] I. Endres, and D. Hoiem, "Category-independent object proposals with diverse ranking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014, pp. 222–234. [2](#)
- [34] Y. Li, D. Min, M. N. Do, and J. Lu, "Fast Guided Global Interpolation for Depth and Motion," in *European Conference on Computer Vision (ECCV)*, 2016. [7](#)
- [35] R. Guo, and D. Hoiem, "Support surface prediction in indoor scenes," in *IEEE International Conference on Computer Vision (ICCV)*, 2013. [6](#)
- [36] F. Tombari, S. Salti, and L. D. Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 809–812. [2](#)
- [37] F. Crow, "Summed-area tables for texture mapping," in *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 1984, pp. 207–212. [4](#)
- [38] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 226–231. [8](#)
- [39] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. [8](#), [9](#), [10](#), [12](#)
- [40] B. Alexe, T. Deselaers, V. Ferrari, "What is an object?," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 73–80. [10](#)



**Ramanpreet Singh Pahwa** received his B.Sc. from Carnegie Mellon University, USA in 2007. He worked in Data Storage Institute and Institute for Infocomm Research, Singapore from 2007 to 2009. He is currently pursuing his Ph.D. in University of Illinois at Urbana-Champaign (UIUC), USA and working as a summer intern in Advanced Digital Sciences Center (ADSC), Singapore. His research interests include computer and robot vision, 3D reconstruction, and depth cameras.



**Jiangbo Lu** (M'09-SM'15) received his B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering, Katholieke Universiteit Leuven, Leuven, Belgium, in 2009. Since 2009, he has been working with the Advanced Digital Sciences Center, Singapore, which is a joint research center between the University of Illinois at Urbana-Champaign, USA, and the Agency for Science, Technology and Research (A\*STAR), Singapore, where he is leading a few research projects as a Senior Research Scientist. His research interests include computer vision, visual computing, image processing, robotics, interactive multimedia applications and systems, and efficient algorithms for various architectures. Dr. Lu served as an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) in 2012–2016. He received the 2012 TCSVT Best Associate Editor Award.



**Nianjuan Jiang** received her B.E. and Ph.D. degree from Department of Electrical & Computer Engineering, National University of Singapore in 2007 and 2013, respectively. She is currently a Researcher with ADSC. Her research interest includes computer vision and computer graphics, and especially efficient and robust 3D reconstruction systems.



**Tian Tsong Ng** is a research staff in Institute for Infocomm Research, Singapore, since 2007 and currently a deputy department head for Situational Awareness Analytics Department. He received his B.Eng from Malaya University in 1998, M.Phil. from Cambridge University in 2001, and Ph.D. from Columbia University in 2007.

He won the Microsoft Best Student Paper Award at ACM Multimedia Conference in 2005, the John Wiley & Sons Best Paper Award at the IEEE Workshop in Information Security and Forensics in 2009.



**Minh N. Do** (M'01-SM'07-F'14) received the B.Eng. degree in computer engineering from the University of Canberra, Australia, in 1997, and the Dr.Sci. degree in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2001.

Since 2002, he has been on the faculty at the University of Illinois at Urbana-Champaign (UIUC), where he is currently a Professor in the Department of ECE, and holds joint appointments with the Coordinated Science Laboratory, the Beckman Institute for Advanced Science and Technology, and the Department of Bioengineering. His research interests include signal processing, computational imaging, geometric vision, and data analytics.

He received a CAREER Award from the National Science Foundation in 2003, and a Young Author Best Paper Award from IEEE in 2008. He was named a Beckman Fellow at the Center for Advanced Study, UIUC, in 2006, and received of a Xerox Award for Faculty Research from the College of Engineering, UIUC, in 2007. He was a member of the IEEE Signal Processing Theory and Methods Technical Committee, Image, Video, and Multidimensional Signal Processing Technical Committee, and an Associate Editor of the IEEE Transactions on Image Processing.