Real-time Progressive 3D Semantic Segmentation for Indoor Scenes

Quang-Hieu Pham[1] Binh-Son Hua[2] Duc Thanh Nguyen[3]
Sai-Kit Yeung[4]
[1]Singapore University of Technology and Design
[2]The Tokyo University [3]Deakin University
[4]Hong Kong University of Science and Technology

Abstract

The widespread adoption of autonomous systems such as drones and assistant robots has created a need for real-time high-quality semantic scene segmentation. In this paper, we propose an efficient yet robust technique for on-the-fly dense reconstruction and semantic segmentation of 3D indoor scenes. To guarantee (near) real-time performance, our method is built atop an efficient super-voxel clustering method and a conditional random field with higher-order constraints from structural and object cues, enabling progressive dense semantic segmentation without any precomputation. We extensively evaluate our method on different indoor scenes including kitchens, offices, and bedrooms in the SceneNN and ScanNet datasets and show that our technique consistently produces state-of-the-art segmentation results in both qualitative and quantitative experiments.

1. Introduction

Recent hardware advances in consumer-grade depth cameras have made high-quality reconstruction of indoor scenes feasible. RGB-D images have been used to boost the robustness of numerous scene understanding tasks in computer vision, such as object recognition, object detection, and semantic segmentation. While scene understanding using color or RGB-D images is a well explored topic [41, 13, 30], good solutions for the same task in the 3D domain have been highly sought after, particularly, those can produce accurate and high-quality semantic segmentation.

In this work, we propose a (near) real-time method for high-quality dense semantic segmentation of 3D indoor scene. The backbone of our work is a higher-order conditional random field (CRF) designed to infer optimal segmentation labels from the predictions of a deep neural network. The CRF runs in tandem with a revised pipeline for real-time 3D reconstruction using RGB-D images as input. In contrast to traditional dense model, our CRF accepts additional higher-order constraints from unsupervised object analysis, resulting in high-quality segmentation. An example output from our proposed method is shown in Figure 1. Experiments proved that our method is capable of producing high-quality semantic segmentation and achieve adequate temporal consistency. In summary, our contributions are:

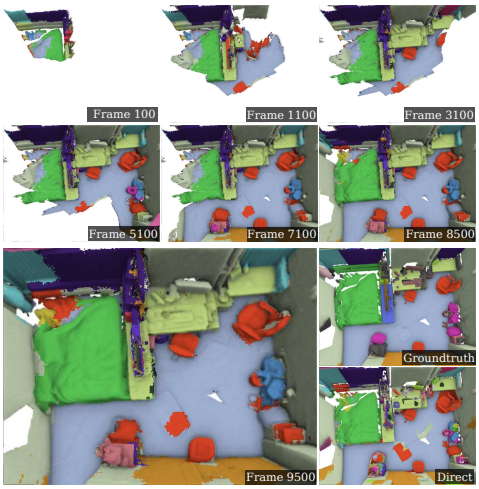


Figure 1: Progressive semantic segmentation of a 10K-frame bedroom scene in real time. Our method can resolve errors in segmentation while scanning. Note the segmentation error on the bed being gradually fixed as the user scans the scene.

- A higher-order conditional random field that can resolve noisy predictions from a deep neural network into a coherent 3D dense segmentation, using additional object-level information.
- An extended reconstruction pipeline, including an efficient voxel clustering technique, for efficient (near) real-time full-scene inference while scanning.
- A thorough evaluation of state-of-the-art real-time semantic segmentation algorithms on two large scale indoor datasets, namely SceneNN [17] and ScanNet [8].
- Beyond category-based semantic segmentation, we also extend our method to instance-based semantic segmen-

tation, and provide the first evaluation of real-time instance segmentation on SceneNN dataset.

2. Related Work

Indoor semantic segmentation. In their seminal work, Silberman *et al*. [41] proposed a technique to segment cluttered indoor scenes into floor, walls, objects and their support relationships. Their well-known NYUv2 dataset has since sparked new research interests in semantic segmentation using RGB-D images. Long *et al*. [30] adapted neural networks originally trained for classification to solve semantic segmentation by appending a fully connected layer to the existing architecture. This method, however, tends to produce inaccuracies along object boundaries. Since then, different techniques [52, 4] has been proposed to address this issue. Some recent works also explored instance segmentation [14, 5], but such techniques only work in 2D.

In the 3D domain, a few datasets for 3D scene segmentation have also been proposed [17, 8, 2]. Early techniques focused on solving the problem by exploiting 3D volumes. For example, Song *et al*. [42] and Dai *et al*. [10] proposed a network architecture for semantic scene segmentation and completion at the same time. Point-based deep learning [37, 28, 18, 47, 19] took another direction and attempted to learn point representation for segmentation directly from unordered point clouds. While the results from these neural networks are impressive, they only take as input a small point cloud of a few thousand points. To address large-scale or structural point cloud, clustering techniques such as superpoints [25] or hierarchical data structures such as octree [39] and kd-tree [22] have been proposed. Hybrid methods such as SEGCloud [43] turns the point clouds into volumes for prediction with a neural network and then propagates the results back to the original point cloud.

Instead of directly processing in 3D, multiple view techniques [24, 26, 31, 38, 9] focused on transferring 2D segmentation to 3D. Other methods further exploit object cues such as spatial context [11]. Our method is based on multi-view segmentation as such techniques scale better to large-scale scenes. Concurrently, we also aim to achieve real-time performance with progressive scene reconstruction. We would focus our discussion to the most relevant interactive and real-time techniques.

Real-time semantic segmentation. Our real-time semantic segmentation system requires an online dense 3D reconstruction system. KinectFusion [33] showed us how to construct such system. To overcome the spatial constraints in the original KinectFusion implementation, which prohibits large-scale 3D scanning, Nießner *et al*. [34] used voxel hashing to reduce the memory footprint. Valentin *et al*. [45] proposed an interactive scanning system where the segmentation is learnt from user inputs. Unlike them, our method is completely automatic without the need of user interaction, and thus more suitable for robotics applications. Our method is based on a segmentation prediction with 2D deep neural networks, a 2D-3D label transfer and optimization with a conditional random field (CRF). To our knowledge, the closest works to ours in this aspect is from the robotics community [16, 48, 46, 32, 15, 50]. Early methods [16, 48, 46] utilized random forest classifiers to initialize the CRF but their end-to-end pipeline performance was far from real time. Similar to our approach, McCormac *et al*. [32] utilized segmentation predictions from a deep neural network and achieved real-time performance on sparse point cloud. In comparison, our method preserves surface information completely by working with an on-the-fly sparse volume representation from Voxel Hashing [34], and introduce a higher-order conditional random field model to refine 3D segmentation.

Conditional random field. The CRF model, often containing unary and pairwise terms, is commonly used as post-processing step [7] to address noise in semantic segmentation. Krähenbühl and Koltun [23] proposed an efficient message passing method to perform inference on a fully-connected model. Recently, with the immense advances in deep learning, it is possible to embed CRF into neural networks [56, 3] and its parameters can be learnt jointly with the network via back-propagation. While representing CRF by a recurrent neural network [56, 3] is advantageous, applying such end-to-end framework to our problem poses some challenges. First in the context of progressive 3D reconstruction and segmentation, 2D predictions from multiple views have to be combined to produce the labeling of 3D model, which is not supported in the previous method where only the segmentation of one single image is predicted. Second, their methods is computationally demanding which does not fit our real-time requirement. Third, the number of 2D images used to calculate the unaries is not fixed, compared to using only one input image as in previous approaches. In this work, we instead run the CRF separately on 3D after processing 2D semantic predictions from a convolutional neural network.

CRF is also extended with high-order potentials to further improve coherency in the label prediction. For example, Zhu *et al*. [57] explored high-order CRF for co-segmentation on images. Yang *et al*. [50] uses a hierarchical CRF with potentials from super-pixels on images for fast outdoor scene segmentation. The CRF model we propose in this work is a higher-order CRF that includes object cues for indoor scenes and works in tandem with the geometry reconstruction. Our idea is that to obtain a coherent, high-quality segmentation, vertices in the same object should be consider as a whole in the model. Moreover, noises and inconsistencies should be fixed regularly as the user scans through the scene.

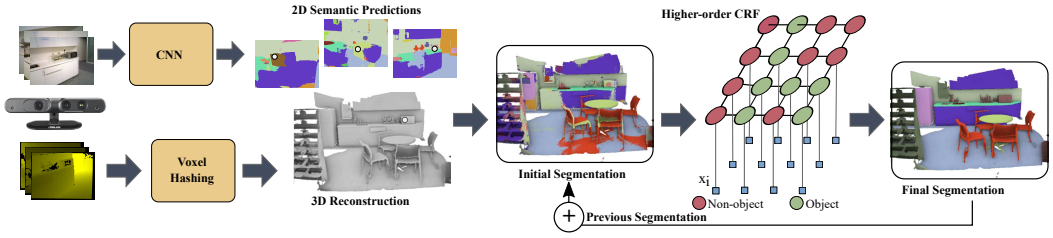


Figure 2: Overview of our progressive indoor scene segmentation method. From continuous frames of an RGB-D sensor, our system performs on-the-fly reconstruction and semantic segmentation. All of our processing is performed on a frame-by-frame basis in an online fashion, thus useful for real-time applications.

3. Real-time RGB-D Reconstruction

We now introduce our proposed method for the progressive dense semantic segmentation problem. An overview of our framework is shown in Figure 2.

3.1. Semantic label fusion

Our online scanning system is built on top of the Voxel Hashing [34] pipeline, reconstructing both geometric and semantic information of the scene in real time. In principle, given an incoming frame prediction from CNN, we must update the semantic label for each active voxel accordingly, using the same integration process as described in KinectFusion [33]. For this problem, McCormac *et al*. [32] store a full discrete probability distribution in each voxel, and update it by using recursive Bayesian rule. However, doing so requires a large amount of memory and does not scale well with large number of semantic classes. We employ the update process proposed by Cavallari and Di Stefano [6], where each voxel only stores the current best label and its confidence.

3.2. Progressive super-voxel clustering

Now we explain in details our super-voxel clustering method, which will provide a new domain to define our CRF with higher-order constraints. Our super-voxel clustering method resembles previous local k-means clustering techniques such as VCCS [35] or SLIC [1]. The main difference in our super-voxel clustering method is that, to amortize the computation cost, we create super-voxels in a progressive manner, performing one clustering iteration at a time, which will adapt better to the changes in the current reconstructed scene. In our system, we consider common features such as voxel color and position to define the distance measure D:

$$D = \sqrt{\frac{\alpha D_c}{n_c} + \frac{\beta D_s}{n_s}} \tag{1}$$

where D_c and D_s are the color and spatial distances, with n_c and n_s act as the normalizers; α and β control the relative weighting of color and spatial distances. In all of our experiments, we set α and β to 1; the normalization values n_c and n_s are based on the chosen voxel size which is $0.008m$ and the CIELab color space. Here one can further utilize voxel normals for the distance measure but we found that the quality of the clustering does not improve much despite of the expensive cost to compute normals per voxel. Another possible extension is to consider features provided by the 2D semantic segmentation network in the distance measure. However, the memory storage per voxel would be very costly because each feature vector often has at least tens of floating point numbers. Some compressions might help in this case.

Suppose that an existing set of super-voxels are already provided. For an incoming RGB-D frame at time t, after camera pose estimation, we can find out the current active set V_t of voxels using an inside/outside check on the current camera frustum. Our goal is to assign each of these voxels into a super-voxel (or cluster). This process is as follows: first new seeds are sampled on uninitialized regions, based on a chosen spatial interval S. For each active voxel, we assign it to the nearest cluster according to the distance in Equation 1. Next, we update the centers information based on the new cluster assignment. This process is repeated for every incoming RGB-D frame, providing a "live" unsupervised over-segmentation of the scene.

Our progressive super-voxel building scheme fits well into the common dense RGB-D reconstruction pipelines such as KinectFusion [33] or Voxel Hashing [34], and can be implemented efficiently on the GPU. In practice, we only consider voxels close to the surface, based on their distance-to-surface values. Performing inference on these super-voxels significantly reduces the domain size of our CRF, and thus paves the way for real-time semantic segmentation.

3.3. Real-time object proposal

For 3D object proposal, Karpathy *et al*. [21] presented a method for discovering object models from 3D meshes of indoor environments. Their method first generates object candidates by over-segmenting the scene on different thresholds. The candidates are then evaluated and suppressed based on geometric metrics to produce the final proposals. Kanezaki [20] proposed an extension of selective search for object proposal on 3D point cloud.

One common drawback of these methods is their high computation cost, since they require a costly object analysis

on different scales. This process has to be done for every update, which hinders real-time performance. In this work, we explore on a new direction for object proposal, in which we propose object based on statistical evidences.

Our object proposal is come from a simple observation: given an object and multiple observations, it should be identified as an object in most of the corresponding 2D semantic predictions. Hence, for each incoming RGB-D frame, we update the objectness score of a voxel given its current predicted label. Specifically, we decrease the objectness score if the prediction is a non-object label, *i.e*. wall, floor, or ceiling; and increase it otherwise. To perform object proposal, we employ an efficient graph-based segmentation algorithm from Felzenszwalb and Huttenlocher [12]. The edge weight between two super-voxels i and j is defined as $w_{i,j} = w^{\alpha}_{i,j} + w^{\eta}_{i,j} + w^{\omega}_{i,j}$ where $w^{\alpha}_{i,j}$, $w^{\eta}_{i,j}$, and $w^{\omega}_{i,j}$ are the edge weight for voxel color, normal, and objectness, respectively. We normalize the each of the weights accordingly. To reduce computation cost, we only compute the terms using representative values from super-voxel centroids.

4. Higher-order CRF Refinement

Using CRF as a post-processing step is a common technique in semantic segmentation. However, for real-time applications, there are two limitations that we must address. First is the classification errors caused by inconsistencies, sometimes known as "bleeding", that is also reported by Valentin *et al*. [44]. The second issue is scalability, since the number of vertices in the graph grows to millions during scanning, causing CRF optimizations to become much slower over time. In this work, we address both limitations by introducing a CRF model with *higher-order constraints* on *super-voxels* to perform online segmentation. This model is lightweight and very easy to compute, allowing it to work on a wide range of indoor scenes, while remaining computationally efficient for real-time use.

Let $\mathcal{M}^t$ be the 3D geometry at time t with N^t super-voxels. In a semantic segmentation problem, we attempt to assign every super-voxel with a label from a discrete label space, denoted $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$. Let $\mathbf{X}^t = \{x^t_1, \ldots, x^t_N\}$ define a set of random variables, one for each super-voxel, where $x^t_i \in \mathcal{L}$. An assignment of every x^t_i will be a solution to the segmentation problem at time t. For shorter notation, we will drop the superscript time notation from now on.

Given the above definitions, we define a graph $\mathcal{G}$ where each vertex is from $\mathbf{X}$. In addition, let $\mathcal{C}$ be the set of cliques in $\mathcal{G}$, given by an object proposal method. For every clique $r \in \mathcal{C}$, we can select a corresponding set of random variables $\mathbf{x}_r$ that belongs to r. Our CRF model introduces three new types of higher-order potential, namely objectness potential ψ^O, consistency potential ψ^C and object relationship potential ψ^R. These terms are later explained in Section 4.1, 4.2, and 4.3, respectively. Our complete CRF model is then defined as

$$E(\mathbf{X}) = \sum_i \varphi(x_i) + \sum_{i<j} \psi^P(x_i, x_j) + \sum_{r\in\mathcal{C}} \psi^O(\mathbf{x}_r) + \sum_{r\in\mathcal{C}} \psi^C(\mathbf{x}_r) + \sum_{r,q\in\mathcal{E}(\mathcal{C})} \psi^R(\mathbf{x}_r, \mathbf{x}_q) \tag{2}$$

where $\varphi(x_i)$ and $\psi^P(x_i, x_j)$ are the unary and pairwise terms used in the traditional dense CRF model. The unary term represent the prediction from a local classifier. In our case, it is obtained from fusing CNN predictions during reconstruction.

The pairwise (smoothness) potential $\psi^P(x_i, x_j)$ is parameterized by a Gaussian kernel

$$\psi^P(x_i, x_j) = \mu_{ij} \exp\left(-\frac{\mid p_i - p_j \mid}{2\theta^2_\alpha} - \frac{\mid n_i - n_j \mid}{2\theta^2_\beta}\right) \tag{3}$$

where μ_{ij} is the label compatibility function between x_i and x_j given by the Potts model; p_i and n_i are the location and normal of the i^{th} super-voxel; θ_α and θ_β are standard deviations of the kernel.

4.1. Objectness potential

The term $\psi^O(\mathbf{x}_r)$ captures the mutual agreement between the objectness score of a clique and its semantic label. Ideally, we would want a clique with low objectness score to take a non-object label, *i.e*. wall, floor, or ceiling; and inversely. To model the objectness potential of a clique, we first introduce latent binary random variables $y_1, \ldots, y_{|\mathcal{C}|}$. y_k can be interpreted as follows: if the k^{th} proposal has been found to be an object, then y_k is 1, otherwise it will be 0. Let $\mathcal{O}$ be the subset of $\mathcal{L}$, which comprises of object classes in the label space. We can then define our objectness potential

$$\psi^O(\mathbf{x}_r) = \begin{cases} \frac{1}{|\mathbf{x}_r|} \sum_{i\in\mathbf{x}_r} [x_i \notin \mathcal{O}], & \text{if } y_r = 1, \\ \frac{1}{|\mathbf{x}_r|} \sum_{i\in\mathbf{x}_r} [x_i \in \mathcal{O}], & \text{if } y_r = 0, \end{cases} \tag{4}$$

where $[\cdot]$ is a function that converts a logical proposition into 1 if the condition is satisfied, otherwise it would be 0. The purpose of this term is to correct misclassification errors in the local classifier, based on external unsupervised information from object proposal.

4.2. Label consistency

The term $\psi^C(\mathbf{x}_r)$ enforces regional consistency in semantic segmentation. Since we want vertex labels in the same clique to be homogeneous, the cost function penalizes label based on its frequency in the clique. Let $f_r(l_k)$ be the normalized frequency of label $l_k \in \mathcal{L}$ inside the r^{th} clique,

which is of the range between 0 and 1. The consistency cost will be the entropy of the underlying distribution:

$$\psi^C(\mathbf{x}_r) = -\sum_{l_k \in \mathcal{L}} f_r(l_k) \log f_r(l_k) \quad (5)$$

This term dampens infrequent labels in a clique. In experiments, We observed that the label consistency cost helps fixing low frequency errors in the output segmentation.

4.3. Region relationship

The relationship potential ψ^R encodes the relation between two regions (cliques) and their semantic labels. This cost is applied on neighboring regions, based on super-voxel connectivity. In our model, the term $\psi^R(\mathbf{x}_r, \mathbf{x}_q)$ is defined based on the co-occurrence of class labels in the regions. Specifically, let $\mathcal{E}(\mathcal{C}) \subset \mathcal{C} \times \mathcal{C}$ be the edges between connected cliques. The object relationship cost between $\mathbf{x}_r$ and $\mathbf{x}_q$ is defined as follows,

$$\psi^R(\mathbf{x}_r, \mathbf{x}_q) = -\sum_{l_i \in \mathcal{L}} \sum_{l_j \in \mathcal{L}} \log \left(f_r(l_i) f_q(l_j) \Lambda_{l_i,l_j} \right) \quad (6)$$

where Λ_{l_i,l_j} is the co-occurrence cost based on the class labels l_i and l_j and designed such that the more often l_i and l_j co-occur, the greater Λ_{l_i,l_j} is. This cost acts like a prior to prevent uncommon label transition, *e.g.* chair to ceiling, ceiling to floor, etc; and can be learnt beforehand. f_r and f_q are the label frequencies, as presented in (5).

In our CRF model, each term is accompanied with a weight to balance their values that we omit them in our formulas for better clarity. We learn these weights by grid search, and keep them unchanged in all of the experiments.

Finally, semantic segmentation can be done by minimizing the energy function $E(\mathbf{X})$ defined in (2). In this paper, we adopt the variational mean field method [23] for efficiently optimizing $E(\mathbf{X})$. Details of the inference process can be found in the supplementary material.

4.4. Temporal consistency

We support temporal consistency with a simple modification of the unary term as follows. To minimize storage, let us only consider time $t-1$ and time t. The unary term becomes a weighted sum that takes as input the final labels at time $t-1$ ($\mathbf{X}_{CRF}$, after CRF of time $t-1$) and the CNN predicted labels at the time t ($\mathbf{X}_{predicted}$, before CRF): $\mathbf{X}^t_{unary} = \tau \mathbf{X}^t_{predicted} + (1-\tau)\mathbf{X}^{t-1}_{CRF}$ where $\mathbf{X}$ are the label probabilities, and $\tau \in [0, 1]$ is a scalar value. Smaller τ favors temporal consistency. We set τ empirically by plotting the segmentation accuracy with multiple τ. Our experiment (see supplementary) shows that $\tau = 0.5$ strikes a balance between accuracy and temporal consistency.

4.5. Instance segmentation

Beyond category-based semantic segmentation, we extend our technique to support instance-based semantic segmentation in real time, which we refer to as *instance segmentation* for brevity. The key change is that CRF model now outputs instance IDs instead of class segmentation labels. Other terms and the optimization process are kept unchanged.

A straightforward approach for instance segmentation would be utilizing a deep neural network that can perform instance-based segmentation in 2D, and then propagate the predictions from 2D to 3D as in the category-based semantic segmentation case. However, this approach requires us to track the instance IDs over time, which is in fact a challenging problem, since the networks, e.g., [14], can only predict one frame at a time.

Our solution is to combine category-based semantic segmentation network with the following instance-based segmentation to yield instance IDs. For each vertex x_i in the CRF, we have to define probabilities over every possible instance IDs. The label space, $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$, would be the set of all instance IDs in the current 3D reconstruction. Performing CRF inference on the entire set of instance IDs would be infeasible. Here we reduce the problem size by first filtering out the instance IDs that are not in the current camera frustum at time t, giving a reduced label space $\mathcal{L}^t$. Our higher-order CRF will only optimize instance labels of super-voxels in the camera frustum, instead of the entire scene as before. The result is then fused into the current model.

Another issue in progressive instance segmentation is how to update the label space $\mathcal{L}$, since online scanning will continuously introduce new instances to our model. We tackle this problem by creating a special *unknown* instance ID. All of the newly scanned voxels will be initialized with unknown. After each CRF inference step, the largest connected component, which is based on category, belongs to the unknown instance will be spawned as a new instance. We also update the set of instance IDs accordingly.

5. Experiments

Experiment setup. In SemanticFusion [32], the authors chose to evaluate on NYUv2 dataset, a popular 2D dataset for semantic segmentation task. However, evaluation in 2D by projecting labels from 3D model to 2D image is not completely sound; since 2D images cannot cover the entire scene, and there are potential ambiguities when doing 2D-3D projection. To tackle this problem, we perform our evaluation on SceneNN [17] and ScanNet [8], which are two 3D mesh datasets with dense annotations. Our evaluation can act as a reference benchmark for real-time 3D scene segmentation systems.

ID	Direct	SF	Ours	
	Class	Class	Class	Instance
011	0.770	0.776	**0.800**	0.521
016	0.607	0.625	**0.680**	0.342
030	0.584	0.597	**0.658**	0.568
061	0.751	0.777	**0.809**	0.591
078	0.497	0.515	**0.535**	0.349
086	0.622	0.646	**0.668**	0.350
096	0.659	**0.668**	0.666	0.265
206	0.766	**0.778**	0.775	0.417
223	0.669	0.689	**0.729**	0.409
255	0.423	0.439	**0.558**	0.486

Table 1: Comparison of category-based semantic segmentation accuracy on typical scenes in SceneNN dataset. We report performances on office, kitchen, bedroom, and other scenes. Our proposed CRF model consistently outperforms the naive approach that directly fuses neural network predictions to 3D (Direct) [6], and SemanticFusion (SF) [32]. Please also refer to the supplementary document for weighted IoU scores. The final column reports the average precision scores of our instance-based segmentation results.

Acc.	SegNet		FCN-8s		SSCNet	
ID	Base	Ours	Base	Ours	Base	Ours
011	0.747	**0.837**	0.667	0.743	0.475	0.497
016	0.556	0.714	0.580	0.623	0.648	**0.798**
030	0.554	0.668	0.584	**0.704**	0.505	0.510
061	0.549	**0.841**	0.324	0.457	0.700	0.693
078	0.542	**0.666**	0.551	0.663	0.515	0.588
086	0.587	**0.686**	0.491	0.631	0.543	0.517
096	0.615	**0.683**	0.577	0.619	0.631	0.658
206	0.659	0.812	0.626	0.828	**0.861**	0.834
223	0.648	0.758	0.693	**0.760**	0.644	0.639
255	0.521	0.654	0.577	**0.718**	0.547	0.661

Table 2: Accuracy scores of offline semantic segmentation task on SceneNN [17]. Our proposed CRF model consistently improves the accuracy of the initial predictions from SegNet, FCN-8s [30] and SSCNet [42]. Please refer to the supplementary document for weighted IoU scores and more results on ScanNet [8].

We adopt two common metrics from 2D semantic segmentation for our 3D evaluation, namely vertex accuracy (A) and frequency weighted intersection over union ($wIoU$). Due to space constraint, we only show our accuracy evaluation in this section. Please refer to our supplementary document for the wIoU evaluation.

Implementation details. To get the 2D segmentation predictions, we use SegNet [4] trained on SUN RGB-D dataset. We chose SegNet as it has better accuracy for indoor scenes but more compact and faster alternatives [36, 49, 27, 40, 55, 51, 54] could be used. The CRF inference is the work by Krähenbühl and Koltun [23]. For the best performance and responsiveness for real time use, we run one iteration of the CRF inference in each frame, and the CNN predictions every K frames (with K = 10 in our experiment). This aligns with the fact that the geometry change is usually subtle in each frame, and label propagation with CRF per frame is sufficient for a good prediction while maximizing responsiveness. After K frames when geometry changes more significantly, we update the segmentation with the more accurate but costly CNN predictions.

Online semantic segmentation. We compare our approach to the following methods: (1) Direct label fusion [6]; and (2) SemanticFusion [32]. To give a fair comparison, all of our online results are reconstructed using the same camera trajectories and semantic predictions from SegNet.

We present the performance comparison of our algorithm in various indoor settings. Results are shown in Table 1. Our method outperforms SemanticFusion and the direct fusion approach in almost all of the scenes. Qualitative results also show that our method is less subjective to noise and inconsistencies in segmentation compared to other approaches, especially on object boundaries.

Offline semantic segmentation. We further investigate our model robustness subject to different types of initial segmentation. We perform the experiment in offline setting, taking unary predictions from different neural networks and refine them using our proposed higher-order CRF. For the offline experiment, since the meshes are already provided, we run CRF inference directly on a per-vertex level to produce highest segmentation quality. All of the neural networks are trained on the NYUv2 dataset [41].

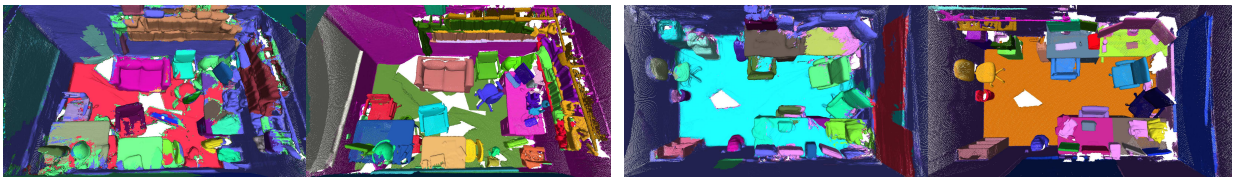
Results from SegNet [4], SSCNet [42], and FCN-8s [30] are shown in Table 2. Note that SSCNet produces a $60 \times 36 \times 60$ volume low resolution segmentation for entire scene due to memory constraints, so we need to re-sample to a higher resolution. In contrast, our 2D-to-3D approach can achieve segmentation on high resolution meshes at almost real-time rate. Again, our method improves SegNet by 10% in accuracy, SSCNet by 8%, and FCN by 9%. This shows that our proposed CRF performs robustly to different kinds of unary. See Figure 6 for more detailed qualitative comparisons.

Per-class accuracy. We measured per-class accuracy of our method and SemanticFusion [32] (see Table 3 below). The results show that our method consistently outperforms SemanticFusion. On average, we increase accuracy by 6% compared to SemanticFusion and 11% compared to the direct fusion method.

Runtime analysis. Runtime analysis is performed on a desktop with an Intel Core i7-5820K 3.30GHz CPU, 32GB RAM, and an NVIDIA Titan X GPU. The average runtime breakdown of each step in the pipeline is demonstrated in Figure 4. Specifically, it takes 309.3ms on average to run a single forward pass of neural network. Building supervoxels takes 34.1ms. CRF with higher-order constraints requires additional 57.9 ms. As can be seen, over time when

	wall	floor	cabinet	bed	chair	sofa	table	door
Direct	0.710	0.914	0.471	0.309	0.430	0.555	0.557	0.313
SF	0.728	0.944	0.570	0.343	0.463	0.578	**0.701**	0.386
Ours	**0.750**	**0.965**	**0.620**	**0.375**	**0.649**	**0.661**	0.698	**0.513**
	window	bookshelf	picture	counter	blinds	desk	curtain	pillow
Direct	0.252	0.839	0.202	0.266	0.215	0.236	0.643	0.268
SF	0.315	0.940	**0.225**	0.371	0.210	0.281	0.830	**0.294**
Ours	**0.425**	**0.947**	0.121	**0.551**	**0.231**	**0.408**	**1.000**	0.253
	clothes	ceiling	books	fridge	television	paper	nightstand	sink
Direct	0.197	0.705	0.426	0.700	0.212	0.119	0.076	0.380
SF	0.236	0.800	0.524	0.803	0.277	**0.320**	0.090	**0.388**
Ours	**0.290**	**0.858**	**0.603**	**0.823**	**0.643**	0.097	**0.145**	0.342
	lamp	shelves	bag	structure	furniture	prop	Average	
Direct	0.284	0.000	0.226	0.121	**0.110**	0.275	0.367	
SF	0.391	0.000	0.214	0.169	0.016	0.291	0.423	
Ours	**0.583**	0.000	**0.364**	**0.262**	0.018	**0.312**	**0.484**	

Table 3: Per-class accuracy of 40 NYUDv2 classes on SceneNN dataset from direct fusion, SemanticFusion (SF) and ours. Note that some of the classes are missing from the evaluation data. Best view in color.



Prediction Ground truth Prediction Ground truth

Figure 3: Instance-based semantic segmentation on SceneNN dataset [17].

more regions in the scene are reconstructed, our semantic segmentation still takes constant running time on average.

We compared our online approach to the reference offline approach that runs CNN prediction every frame (Table 2). We see that the accuracy of our online method (Table 1) is about 5% lower on average, but the speed gain is more than 8 times. Our system runs at 10-15Hz. With the same CNN predictions, direct fusion method [6] runs at 17-20Hz, and SemanticFusion [32] runs at 14-16Hz. Note that such methods do not constrain label consistency.

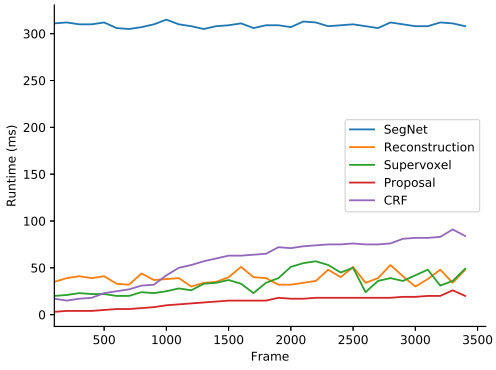


Figure 4: Runtime analysis of our progressive semantic segmentation system.

Temporal accuracy. Since our method can be run in real time, we evaluate the segmentation accuracy over time. For every scene, we measure the accuracy every 100 frames. The progressive segmentation results are shown in Figure 6.

The results suggest that our method *consistently* outperforms other methods in a long run, not just only at a certain time period. In addition, we observe that the accuracy over time sometimes still fluctuates slightly due to the lack of full temporal constraints among the CNN predictions. Addressing this issue could be an interesting future work.

Ablation study. To further understand the performance of our CRF model, we carry out an ablation study to evaluate the effects of each CRF term on the result segmentation. We execute three runs on 10 scenes, each run enables only one term in our CRF model, and record their performances. Figure 5 visualizes the results on these 10 scenes. In general, running full higher-order model achieves the best performance. Enabling individual term is able to outperform the base dense CRF model. The consistency term contributes the most in the performance boost, which validates our initial hypothesis that object-level information is crucial when performing dense semantic segmentation.

Instance segmentation. To evaluate our instance segmentation results, We use the average precision metric [29] with minimal 50% overlap. The results are shown in Table 1. Figure 3 visualizes the instance segmentation in two indoor scenes using our approach. Such results could serve as a baseline to compare with more sophisticated real-time 3D instance segmentation technique in the future.

6. Conclusion

Our proposed system demonstrates the capability to integrate semantic segmentation into real-time indoor scanning

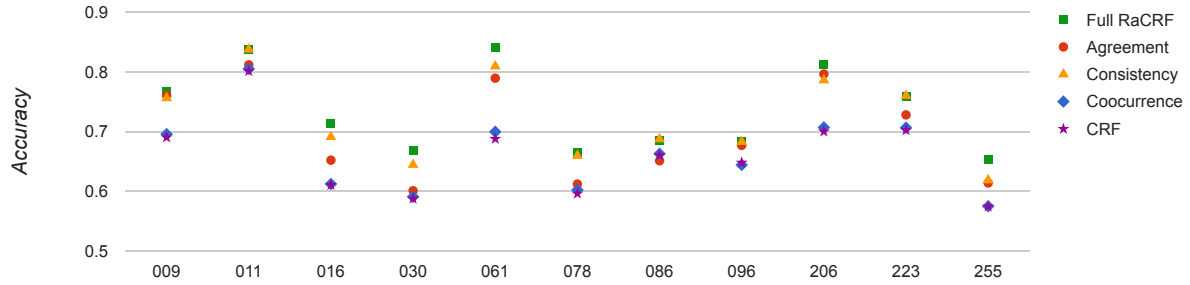


Figure 5: Ablation study on the effects of different CRF terms. There is usually a noticeable gap between the performances of the conventional dense CRF and ours. In addition, individual term helps improving the segmentation accuracy. This study also shows the importance of consistency in semantic segmentation.

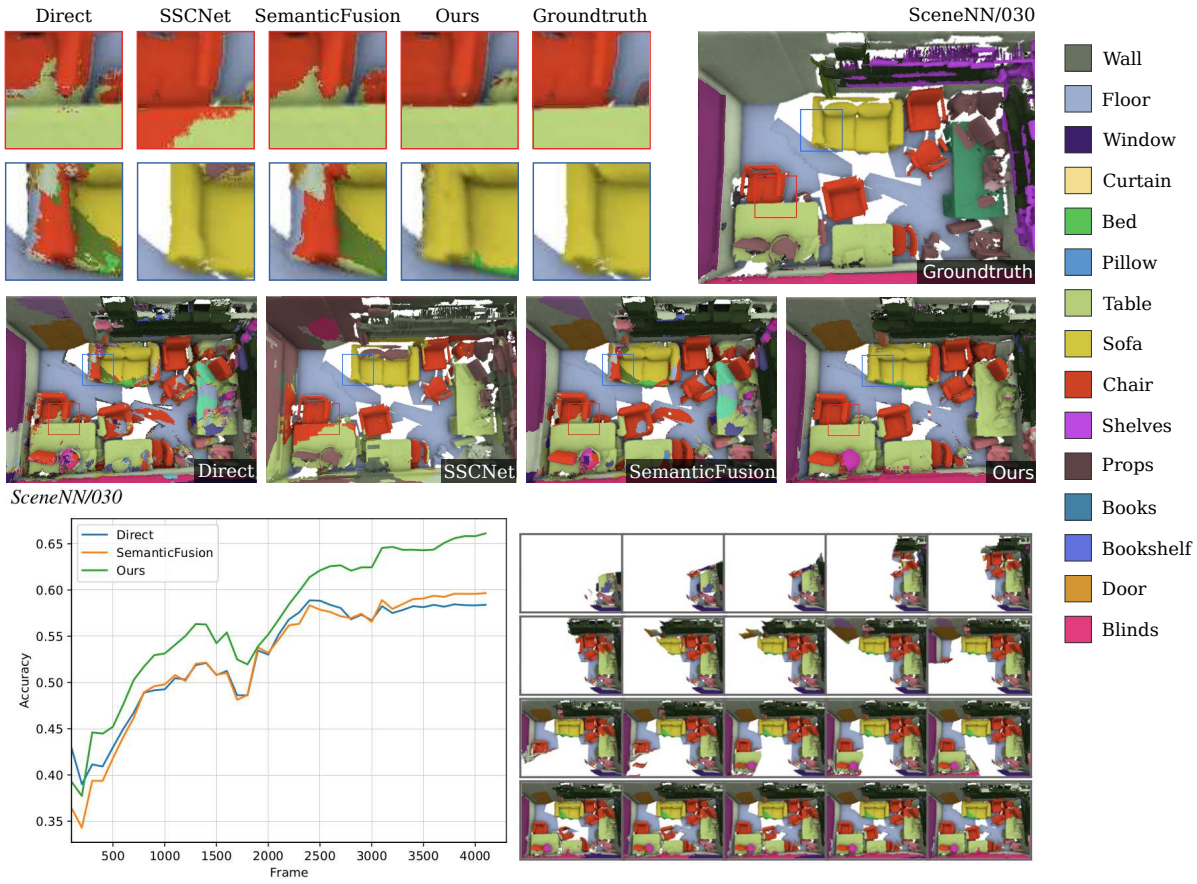


Figure 6: Qualitative results and a temporal accuracy on a selected scene from SceneNN. The top right image is the ground truth segmentation. The results from direct fusion [6], SSCNet [42], SemanticFusion [32] and ours are shown on the second row, respectively. The respective progressive semantic segmentation results of our method are shown on the bottom right. Please refer to the supplementary materials for the full qualitative results.

by optimizing the predictions from a 2D neural network with a novel higher-order CRF model. The results and ground truth category-based and instance-based semantic segmentation will be made publicly available. The results from our system can further be used in other interactive or real-time applications, e.g., furniture arrangement [53], or object manipulation and picking in robotics.

Acknowledgment. This research project is partially supported by an internal grant from HKUST (R9429).

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art

superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[3] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr. Higher order conditional random fields in deep neural networks. In *European Conference on Computer Vision*, pages 524–540. Springer, 2016.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[5] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2858–2866. IEEE, 2017.

[6] T. Cavallari and L. Di Stefano. Semanticfusion: Joint labeling, tracking and mapping. In *Computer Vision–ECCV 2016 Workshops*, pages 648–664. Springer, 2016.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[9] A. Dai and M. Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *ECCV*, 2018.

[10] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *CVPR*, 2018.

[11] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *IEEE International Conference on Computer Vision, 3DRMS Workshop, ICCV*, 2017.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[13] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2013.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[15] D. Held, D. Guillory, B. Rebsamen, S. Thrun, and S. Savarese. A probabilistic framework for real-time 3d segmentation using spatial, temporal, and semantic cues. In *Proceedings of Robotics: Science and Systems*, 2016.

[16] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *ICRA*, 2014.

[17] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung. Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)*, 2016.

[18] B.-S. Hua, M.-K. Tran, and S.-K. Yeung. Pointwise convolutional neural networks. In *CVPR*, 2018.

[19] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3d segmentation on point clouds. In *CVPR*, 2018.

[20] A. Kanezaki and T. Harada. 3d selective search for obtaining object candidates. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 82–87. IEEE, 2015.

[21] A. Karpathy, S. Miller, and L. Fei-Fei. Object discovery in 3d scenes via shape analysis. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2088–2095. IEEE, 2013.

[22] R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *ICCV*, 2017.

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*. 2011.

[24] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *ECCV*, 2014.

[25] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018.

[26] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg. Deep projective 3d semantic segmentation. In *CAIP*, 2017.

[27] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*, 2017.

[28] Y. Li, R. Bu, M. Sun, and B. Chen. Pointcnn. *arXiv:1801.07791*, 2018.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[31] L. Ma, J. Stueckler, C. Kerl, and D. Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. In *IROS*, 2017.

[32] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *ICRA*, 2016.

[33] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *The IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[34] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[35] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2027–2034, 2013.

[36] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[37] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017.

[38] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *ICCV*, 2017.

[39] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, 2017.

[40] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo. Efficient convnet for real-time semantic segmentation. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[41] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[42] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[43] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision (3DV)*, 2017.

[44] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics*, 2015.

[45] J. P. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. H. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *CVPR*, June 2013.

[46] V. Vineet, O. Miksik, M. Lidegaard, and e. a. Nießner. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *ICRA*, 2015.

[47] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[48] D. Wolf, J. Prankl, and M. Vincze. Fast semantic segmentation of 3d point clouds using a dense crf with learned parameters. In *ICRA*, 2015.

[49] Z. Wu, C. Shen, and A. van den Hengel. Real-time semantic image segmentation via spatial sparsity. *arXiv preprint arXiv:1712.00213*, 2017.

[50] S. Yang, Y. Huang, and S. Scherer. Semantic 3d occupancy mapping through efficient high order crfs. In *IROS*, pages 590–597, 2017.

[51] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018.

[52] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[53] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics*, 30(4):86, 2011.

[54] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018.

[55] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[56] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[57] H. Zhu, J. Lu, J. Cai, J. Zheng, and N. M. Thalmann. Multiple foreground recognition and cosegmentation: An object-oriented crf model with robust higher-order potentials. In *WACV*, 2014.