SnapNet-R: Consistent 3D Multi-View Semantic Labeling for Robotics

Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, Aurelien Plyer, David Filliat

▶ **To cite this version:**

Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, Aurelien Plyer, et al.. SnapNet-R: Consistent 3D Multi-View Semantic Labeling for Robotics. IEEE International Conference on Computer Vision Workshop (ICCVW), Oct 2017, Venise, Italy. 10.1109/ICCVW.2017.85 . hal-01808539

HAL Id: hal-01808539
https://hal.science/hal-01808539v1

Submitted on 5 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

SnapNet-R: Consistent 3D Multi-View Semantic Labeling for Robotics

Joris Guerry*, Alexandre Boulch*, Bertrand Le Saux*, Julien Moras*, Aurélien Plyer*, David Filliat†
* ONERA - The French Aerospace Lab, FR-91761 Palaiseau, France
† ENSTA ParisTech, Inria FLOWERS team, Université Paris-Saclay, F-91762 Palaiseau, France

Abstract

In this paper we present a new approach for semantic recognition in the context of robotics. When a robot evolves in its environment, it gets 3D information given either by its sensors or by its own motion through 3D reconstruction. Our approach uses (i) 3D-coherent synthesis of scene observations and (ii) mix them in a multi-view framework for 3D labeling. (iii) This is efficient locally (for 2D semantic segmentation) and globally (for 3D structure labeling). This allows to add semantics to the observed scene that goes beyond simple image classification, as shown on challenging datasets such as SUNRGBD or the 3DRMS Reconstruction Challenge.

1. Introduction

Visual scene understanding is a key capability to let intelligent robots evolve and interact in their environment. After decades of progress, we have reached a point where the acquisition of complex 3D scenes with fine details is now possible through multiple ways: precise laser scanners, commodity Red-Green-Blue-Depth (RGB-D) sensors or reconstruction techniques based on stereo and Simultaneous-Localization and Mapping (SLAM). Theses kinds of reconstruction are sufficient for simple navigation and collision avoidance, but a new step is required for the robot to act purposefully: semantic analysis of the 3D data.

In this paper we present a novel approach for semantic labeling of the scene perceived by a robot. It is built on an efficient multi-view approach called SnapNet [5] which established state-of-the-art results for urban, remote sensing data such as semantic3d [20]. We propose new multi-view sampling strategies and better image and geometry integration. But most importantly, we show how 3D structure reconstruction and 2D semantics can mutually benefit from each other.

3D → 2D : Instantaneously, when the robot gets a single RGB-D capture of its environment, we generate new virtual views which are consistent with the 3D structure of the perceived scene. It improve 2D semantic labeling at training (by data augmentation) and at prediction (by voting procedure).

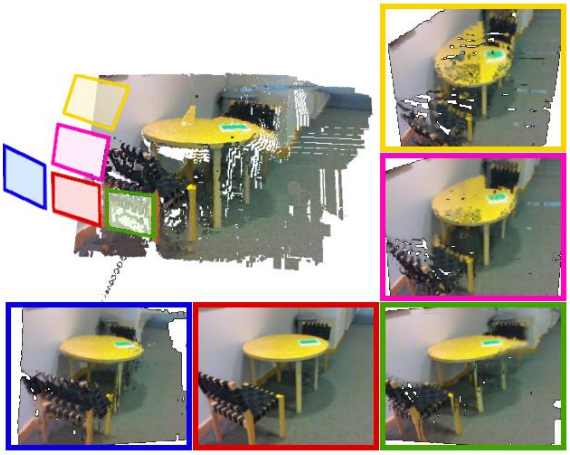


Figure 1. Illustration of the sampling strategy over single view data from SUNRGBD (real distances and angles are not respected for illustration purpose).

2D → 3D : At reconstruction time after capture of multiple images, adding semantics to the images allows to filter the points used for 3D reconstruction and to obtain more precise point clouds, which are in turn easier to label with semantics.

Overall, the contributions of this paper are:

- SnapNet-R, an improved multi-view Convolutional Neural Network (CNN) with simpler image and geometry integration as compared to SnapNet [5];
- 3D-consistent data augmentation for 2D semantic labeling with state-of-the-art performances;
- massive 3D-consistent sampling of 2D virtual views for 3D semantic labeling of reconstructed point clouds, validated on challenging data.

The paper is organized as follows. Section 2 presents the related work on image and point cloud semantic labeling. The description of our approach can be found in

section 3: the core method SnapNet-R is explained in section 3.1 while implementations for RGB-D data and point clouds are detailed in section 3.2 and section 3.3 respectively. Finally, we evaluate our segmentation method in section 4.

2. Related work

Semantic labeling consists in point-wise classification of 2D or 3D data. In 2D, it became popular in the 2000s with challenges such as the CamVid dataset [7] or the Pascal Visual Object Challenge [14], a task consisting in giving a pixel-wise class label. In 3D, the related task is to identify the class membership of each 3D point. It gained interest with the availability of laser scanners which made possible the acquisition of large point clouds for application such as urban cartography [19] or robotics [1]. In the following, we describe the most recent and efficient approaches for tackling this problem, focusing on neural networks and robotics.

In **2D**, fully-convolutional networks (FCN) [32] were a milestone in the field of segmentation with dense prediction: by keeping the 2D spatial structure all along the network, they offered a simple and highly efficient approach. They were followed by several encoder-decoder architectures which exploit the symmetry of input and output spaces: U-Net [37], SegNet [3], etc. Alternatively, Lin *et al.* [31] obtained state-of-the-art results using a multi-scale patch-based architectures with Conditional-Random-Fields (CRFs). While these networks perform in the Red-Green-Blue (RGB) domain, robotics commonly use Depth (i.e. distance from the sensor, denoted by D) as an additional 3D information, as reflected by reference datasets such as SunRGBD [41]. To exploit RGB-D, Gupta *et al.* [18] proposed an object detection method based on Region-CNN [16] with depth encoding and semantic labeling output. Recently, the FuseNet architecture [22] combined an encoder-decoder with early integration of depth at encoding stage. Later developments then lead to the Multi-View Consistent network (MVCNet) [33] which takes benefit of unlabeled frames captured by a mobile robot by warping semantic predictions on these images in a common reference view (for which labels are known) based on a trajectory estimate. With respect to these approaches, our method uses the depth to perform 3D-consistent data augmentation, which allows us to transform a FuseNet (chosen for its state-of-the-art performances) in a multi-view CNN. Unlike MVCNet, our method does not require to be trained on video sequences to extract real, nearby frames but instead it creates virtual views from single RGB-D images, and still imposes 3D consistence.

In **3D**, designing the most discriminating features for training a supervised classifier has been a standard approach for long. For example, in [8] expert features such as normalized height or luminance were selected and aggregated. Generic descriptors able to represent the points and their neighborhood were also proposed: for example the fast point feature histograms [38] or the signature histograms [44]. But now, by using a deep learning framework, representations and classifier are learned all at once. In this field, three approaches compete.

First, **voxel-based** methods use a voxelization of the 3D space to create 3D tensors in order to feed a 3D convolutional neural network (CNN) [29, 45, 34], mainly for object classification. Following this idea of 3D encoding, which may be computationally expensive, Hackel *et al.* [20] proposed to use local, multiscale 3D tensors around the actual 3D points. In the voxel semantic labeling task (which is slightly different from point labeling) of the ScanNet benchmark [11], the proposed baseline network predicts labels for a whole column of voxels at once according to the voxels' neighborhood. Cherabier *et al.* [9] reduce computational cost of this kind of approach by dividing the space and working on smaller 3D blocks.

Second, the **multi-view** strategy consists in applying neural networks to 2D tensors which are collections of images of the scene. For retrieving and classifying shape models of objects, Multi-View CNN (MVCNN) [43] takes several pictures all around a 3D meshed object and then performs image classification using a deep network. The PANORAMA representation [39] introduces another trick: projections on bounding cylinders oriented following the 3 principal directions of the volume. SnapNet [5] randomly takes numerous views all around the scene, creates virtual RGB and geometry-encoded images, and process it through U-net networks. Our approach has common features with these works: we generate snapshots of the 3D scene in order to use a 2D CNN with images as input. But instead of assigning a single label per 3D shape as in MVCNN, we compute dense 3D point labeling. And unlike SnapNet, we directly process RGB-D data in a single network.

Third, **point-based** methods work directly on unordered point sets, using architectures with fully-connected and pooling layers instead of convolutional layers. Thus, PointNet [42] can output classes for the whole 3D shape or perform semantic segmentation of a 3D scene. However, it lacks the ability to capture local context at different scales. To go around this drawback and improve generalization to complex scenes, PointNet++ [36] introduces a feed-forward network which performs alternatively hierarchical grouping of points and PointNet layers optimisazion.

3. Approach: multi-view segmentation networks for 3D semantics

Our approach consists in 3D-consistent view generation for improving the quality of semantic labeling. It may be applied in two cases of semantic labeling : 2D (image and depth map) or 3D point cloud.

In the first case, the objective is semantic segmentation of a unique 2D image for which some depth information is available (such as RGB-D data captured by Microsoft Kinect or Realsense R200 devices). For a single image, we generate other views of the scene such that the views are geometrically consistent with the 3D nature of the observed scene. Thus, all views correspond to what would be seen of the scene from a different point of view. It may be considered as data augmentation, but it differs from standard data augmentation (such as crop, translations, flip...) because it does not benefit only from the information contained in the 2D image plane but can also extract knowledge from the 3D spatial structure through the set of different appearances.

In the second case, the objective is semantic labeling of each point of a point-cloud. We focus in this study on point cloud reconstructed from a sequence of RGB-D images or couples of stereo images, for example using SLAM [13] which have been extensively studied in the last years. Compared to the RGB-D mono-image case, we exploit the higher completeness of the point cloud, allowing us to massively generate viewpoints very different from original cameras positions. We extend the SnapNet approach [5] with a new network architecture and a two-step procedure to generate a labeled point cloud: first 2D labeling of RGB-D images generated from stereo images for filtering a large part of outliers and then, 3D labeling using SnapNet on the points.

3.1. SnapNet-R

In this section we present SnapNet-R, a novel version of the previous SnapNet framework [5]. The SnapNet approach consists in independently doing the 2D semantic segmentation of each generated image from the 3D scene and then efficiently re-projecting the per-pixel attributed label back to its corresponding point in the 3D point cloud. This method is based on three main parts : the 2D image generator, the semantic segmentation technique and the efficient 3D back-projector.

Concerning the 2D image generator, the sampling strategy to pick viewpoints and create virtual views has to be adapted to the available data and to the application. In a complete 3D reconstructed scene (given as a point cloud) the generator has to take into account the observation scale (e.g. getting closer to small objects like cars or far enough from big objects like houses in order to have enough context information) and the orientation angle (e.g. not to look toward the empty sky), but can globally evolve freely in the environment. However, such a strategy no longer holds in single RGB-D image scenes like those from the SUNRGBD dataset [41]. The virtual viewpoints must be chosen close to the original camera pose. Indeed, a RGB-D image contains very sparse information: only points directly seen from a single point of view are captured, and every azimuth or attitude shift from this original pose can lead to significant holes in the newly generated image as seen in Figure 2.

For semantic segmentation, the original approach uses two SegNet [3], one for each modality (RGB and geometric features), and fuses them by a residual correction module [2]. The geometric features are handcrafted: normals, local noise and depth to the virtual camera. Thus, this protocol implies three different steps : training the RGB expert, training the geometric expert and finally training the fusion module. Here, we replace all theses components by a FuseNet SF5, introduced by Hazirbas *et al.* [23], which takes directly RGB and depth map as input and so only needs one training stage.

Finally, the process of back-projection of semantic segmentation information back into 3D point cloud stays unchanged.

Figure 2. Illustration of bad sampling strategies with single view data. Upper left: good sample close to real camera point of view, upper right: seen through dresser, down left: seen from behind, down right: seen from bedside lamp.

3.2. RGB-D geometric data augmentation.

The SUNRGBD dataset is composed only of single view images (an overview is available in Figure 5). This implies a different sampling strategy than for a full dense point cloud scene. Besides, the SnapNet approach was first designed to work on few big scenes and extract as many images from each scene as wanted. But SUNRGBD is composed of 5285 training images and 5050 testing images where each one of them is actually a scene. Thus, we can't sample hundreds of samples in each scene without taking the risk of exploding the final number of generated images. Therefore, we chose a sampling strategy that only generates 5 images for each scene and thus results in 26425 images for training and 25250 images for testing.

As previously said, because of the very sparse aspect of the 3D point cloud generated from RGB-D data, the sampling strategy cannot be random. Bad examples of random selection of viewpoints in SUNRGBD dataset are available in Figure 2. Also, since we decided not to produce too many images per scene we arbitrarily defined 5 camera poses:

- the first viewpoint is the original camera viewpoint
- from the first viewpoint we get 2 more viewpoints by changing the azimuth angle of +10°and -10°
- from the first viewpoint we get 2 more viewpoints by changing the attitude angle of +10°and +20°.

This sampling strategy is illustrated in Figure 1. No up-looking viewpoints are picked because objects mainly stand on the floor and are best viewed from above. Besides, the scenes are inside-rooms so objects are at close range and thus, looking upper could make us miss them.

Furthermore, the depth sensors suffer from bright surfaces like mirrors, windows, screens or glass objects and the measurement can be very noisy resulting in black artifacts in depth map and thus holes in the RGB images resulting from 3D re-projection. That is why we applied inpainting preprocessings : based on Navier-Stokes method [4] for the depth images and by using mean color of non-black neighbors on RGB images (before training and testing).

3.3. 3D scene reconstruction and labeling

We focus here on standard scenarios for robotics, where a robot equipped with various sensors moves around, as in the 3D Reconstruction Meets Semantics (3DRMS) dataset (cf. section 4.1). The image sets for reconstruction of 3DRMS consist of RGB and Gray stereo pairs. A pose estimation is given for each acquisition. In this section, we propose three different reconstruction and labeling pipelines, each of them corresponding to a different robotic use case: from a SLAM fashion reconstruction to global multi-view estimation. These pipelines are presented on Figure 3.

3.3.1 Reconstruction and semantic from navigation

The first pipeline –**Classif 2D**– represents a type of reconstruction that can be used in robotic navigation. Labels and points are accumulated for each acquisition image pair. This pipeline follows 4 steps:

Depth estimation: For each camera pair, we rectify the RGB/Gray images and the ground-truth label images. These rectified data will be used for the whole following tasks. Then, we use the Efficient Large-scale Stereo Matching (ELAS) algorithm [15] in order to compute the disparity map and add a depth channel to the RGB images, resulting in raw RGB-D data.

2D semantic segmentation: Using the train sets of these rectified RGB-D data we train a Fusenet SF5 for semantic segmentation (with same condition as experiment 1 in Table 1). Despite the low number of training images, we are able to generate rough label prediction maps on all sets.

Point cloud reconstruction: For both train and test sets, by taking into account the given global position of cameras and the computed rectifications, we accumulate the points from all RGB-D images into a global coordinates system (a voxel grid with voxel size of 0.01 coordinate unit). Due to the small baseline between acquisition cameras, such an accumulation produce point clouds with a lot of outliers (see Fig. 4 up left). Therefore, we implemented a filtering strategy: first, for each image we filter the points that were too close ($< 1m$) or too far ($> 3.5m$) from the camera. The closest points are mainly due to the on-board vehicle used during recording and the farthest were too noisy because of the small baseline. Then, we filter each image-relative point set according to two filters: label based and geometry based. **Based on labels** provided by the 2D semantic segmentation, we discard unlabeled and background points and points in a 3 pixel margin between two different labeled objects. **Based on geometry:** we compute the normals using [6] and reject points which are too aligned with camera observation axis (dot product between normal and direction vector from camera center to considered point > 0.7) A snapshot of the result on test set is presented in Figure 4 up right.

Direct labeling: like in SnapNet-R, the label at each point is computed by voting from the FuseNet semantic map predictions.

3.3.2 Reconstruction from navigation and 3D labeling

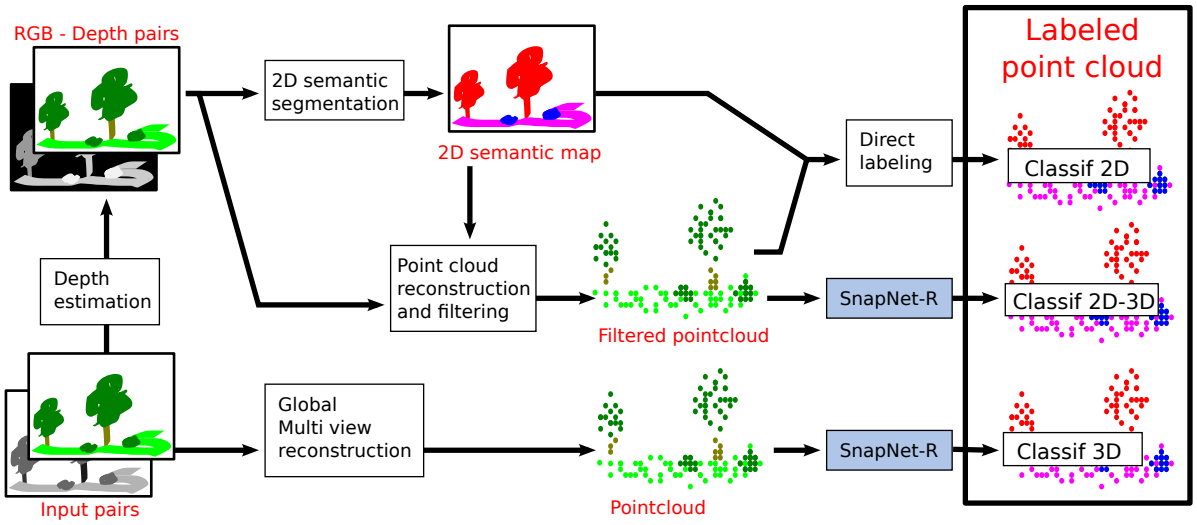
This method –**Classif 2D-3D**– is a variation from the previous pipeline. The reconstruction step is still done in a SLAM fashion but here the 2D semantic maps are only used for filtering. At inference time, to predict labels the former *direct labeling* is replaced by SnapNet-R. At training time, labeled training point clouds are computed using **classif 2D** where the predicted semantic maps are replaced by the provided ground truth.

SnapNet-R: a label is given to each point using the SnapNet-R framework. We adapt SnapNet-R to exploit the point cloud topology, more complete than RGB-D data. We compute new RGB-D pairs by randomly picking view points. The camera position is selected such that: a) it looks in the direction of the scene (particularly, the center of the image is directed towards an existing point) and b) it is positioned over ground level. For each camera direction, we take two snapshots distant of 2 and 5 meters from the targeted point.

3.3.3 Off-line reconstruction and 3D labeling

The last approach –**Classif 3D**– is purely post processing of the data. It is a two step pipeline : a global point cloud reconstruction and a 3D semantic labeling.

Global multi-view reconstruction: we used the Agisoft Photoscan software. By using all images *at once* for



Pipelines and processing blocks are described in section 3.3. Qualitative results on 3DRMS are discussed in section 4.4.

Figure 3. 3D reconstruction and labeling pipelines for the 3DRMS challenge.

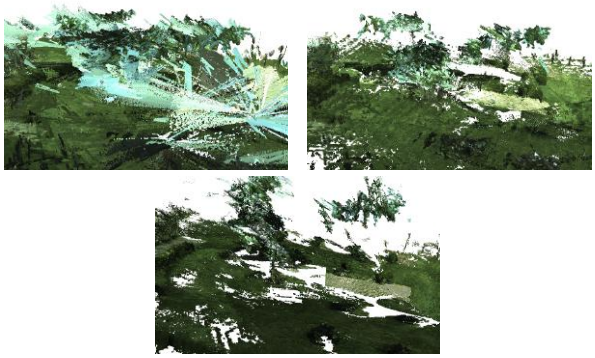


Figure 4. 3DRMS challenge point cloud (test set), incremental accumulation without filter (up left), with filter (up right) and global reconstruction (bottom).

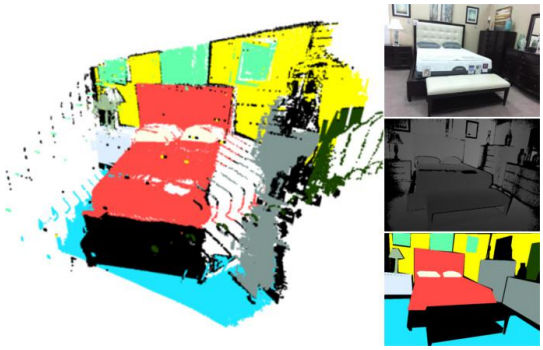


Figure 5. Sample of the SUNRGBD dataset: the 3D semantized point cloud and on right column RGB image, depth map and grounth-truth (see Fig.7 for label legend).

reconstruction, it allows to search for point matching between cameras with a greater baseline than reconstruction from stereo pairs. It yields point clouds with less estimation artifacts such as continuous transitions between objects in depth map estimation (cf. Figure 4, bottom).

SnapNet-R: we labeled the point cloud with the previously trained model (from section 3.3.2).

4. Experiments and results

4.1. Datasets

We validate our approach on three datasets used in robotics which rely on different types of sensors to get 3D information.

Sun-RGBD is a dataset of images captured by low-cost RGB-D cameras [41]. It combines previously existing smaller RGB-D datasets (NYUDv2 [40], Berkeley B3DO [25] & SUN3D [46]) to reach a size of more than 10k images from 4 various RGB-D sensors. It is completed with several types of 2D and 3D annotations over the whole dataset which allows training and evaluating algorithms for various task such as scene classification, semantic segmentation or 3D object detection. All 37 classes are given in Fig. 7 but to name just a few there are: *table, bed, chairs, curtain, fridge, mirror, sink, floor, wall, etc.*. We also evaluate on **NYUDv2** [40] alone, considering the 40 classes and 13 macro-classes segmentation tasks as in [12, 32].

The **3DRMS Reconstruction Challenge** provides us with a dataset of image sequences captured by a robotic platform. This lawn-mower-looking ground robot is equipped with two stereo rigs for RGB and grayscale images and a Leica Lidar sensor for point cloud acquisition. The dataset contains 5 sequences of calibrated images with the corresponding camera pose: 4 for training (totalizing

Figure 6. Overview of the 3DRMS Challenge dataset: training sequence "boxhood row" with the 3D semantized point cloud and on right column 2D semantic annotations over image, front-left RGB image, front-right grayscale image.

108 views) and 1 for testing (with 125 views). For the training part only, there are also ground truth 2D semantic annotations and a semantically annotated 3D point cloud depicting the area of the training sequence (cf. Fig 6). Classes are *Unknown, Grass, Ground, Pavement, Hedge, Topiary, Rose, Obstacle, Tree, Background*.

4.2. Evaluation criteria

Let C be the set of classes (labels) and X be the input points to classify. Let $X_c \subset X$ be the points classified with label $c \in C$. Finally, we note X_c^* the objective classification (i.e. the ground truth) for label c.

Overall accuracy: (OA) is the proportion of well labeled points: $OA = \frac{1}{|X|} \sum_{c \in C} |X_c \cap X_c^*|$.

It does not take into account the unbalance between classes but gives a good view of the global behavior of the classifier.

Mean Accuracy (MA) is computed with the per class accuracies : $MA = \frac{1}{|C|} \sum_{c \in C} \frac{|X_c \cap X_c^*|}{|X_c^*|}$.

Intersection over union (IoU) penalise also false negative predictions : $IoU = \frac{1}{|C|} \sum_{c \in C} \frac{|X_c \cap X_c^*|}{|X_c \cup X_c^*|}$.

4.3. Semantic segmentation of RGB-D data with 3D-consistent multi-view

SUNRGBD: The semantic segmentation task of SUNRGBD is a hard task. The best method to our knowledge, the Context-CRF of Lin *et al.*[31], only achieves 42.3% of IoU. To this purpose they use a very rich and complex set of neural networks, united in a conditional random field framework. They first use a neural network to get a feature map of the input image, another pair of networks to predict patch-to-patch semantic relation and at last, after bilinear upsampling of the low resolution prediction, they apply a boundary refinement post-processing with a Dense CRF [28].

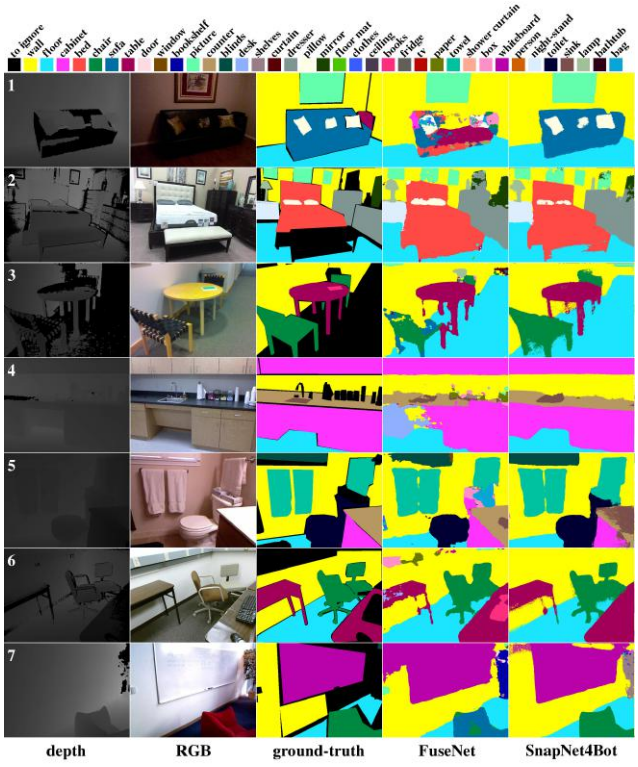


Figure 7. Qualitative segmentation results on SUNRGBD [41]. The first three columns contain depth, RGB and ground-truth images. Last two columns present the results obtained by Fusenet SF5 [23] then SnapNet-R.

By comparison, the results we show in Table 1 do not make use of any refinement post-processing neither conditional random field inference. The complete processing pipeline of our strategy consists in wrapping several images from a single scene to 224x224 patches, infer a low resolution semantic map through a FuseNet SF5 for each of them, upsample each prediction map by nearest neighbor interpolation, reproject all information in the original point of view, and finally vote for the label to assign to every pixel.

Our fully implemented method corresponds to the experiment 9 in Table 1. We achieve a new state of the art result with 58.13% of mean accuracy on the SUNRGBD semantic segmentation task, compared to the 53.4% of the Context-CRF [31]. We also obtain 78.04% of overall accuracy against 78.4% for the Context-CRF and 39.61% IoU (vs 42.3%), which are the second best known results on this dataset.

Compared to the Fusenet SF5 framework alone, our multiview strategy leads to a pure gain of performance for all metrics, especially on mean accuracy going from experiment 1: 54.81% (52.61% in the original article [23]) to 58.13%.

If the Context-CRF method was to be used as the basic

	Training		Testing				
experiment	preproc.	augm.	preproc.	augm.	OA	MA	IoU
LSTM-CF [30] (RGB)	✗	✗	✗	✗	–	48.1	–
FCN 8s [32] (RGB)	✗	✗	✗	✗	68.2	38.4	27.4
Bayesian SegNet [27] (RGB)	✗	✗	✗	✗	71.2	45.9	30.7
Context-CRF [31] (RGBD)	✗	✗	✗	✗	**78.4**	53.4	**42.3**
*FuseNet SF5[23] (RGBD)	✗	✗	✗	✗	76.3	48.3	37.3
DFCN-DCRF [26] (RGBD)	✗	✗	✗	✗	76.6	50.6	39.3
*1 FuseNet SF5	✗	✗	✗	✗	76.88	52.61	39.17
1 FuseNet SF5	✗	✗	✗	✗	77.21	54.81	39.11
2	✗	✗	✓	✗	74.87	52.47	36.68
3	✗	✗	✓	✓	72.52	53.27	33.89
4	✓	✗	✗	✗	72.81	52.02	34.32
5	✓	✗	✓	✗	77.20	55.03	39.33
6	✓	✗	✓	✓	70.25	***56.87***	30.32
7	✓	✓	✗	✗	75.51	53.71	36.65
8	✓	✓	✓	✗	77.57	56.70	38.83
9 SnapNet-R	✓	✓	✓	✓	***78.04***	**58.13**	***39.61***
10** FusetNet SF5 (HD)	✗	✗	✗	✗	71.44	45.97	29.74
11** SnapNet-R(HD)	✓	✓	✓	✓	73.55	50.07	33.46

** Computed at low resolution (224x224) as in [23] on the contrary of all other results computed at native resolution.*
***We also test a High Definition strategy, cropping 224x244 patches at original resolution instead of warping image.*

Table 1. SUNRGBD[41] quantitative results. All numbered experiments are using a FuseNet SF5 trained on all SUNRGBD train data (on the contrary of [23] who removed RealSense images for depth quality reason). We replicated the result of FuseNet SF5 in experiment 1 and applied our full multi-view strategy in experiment 9. For each criterion, best values are emphasized in bold, second best values in bold italics. Results are discussed in section 4.3.

segmentation method of our SnapNet-R approach, the results would be probably even better for each accuracy metrics.

Because all the information is going through a 3D step, we had to preprocess the depth maps not to loose RGB information. Thus we present the results of this processing in Table 1. Experiment 5 uses pre-processing at training and testing time and leads to a little improvement over the FuseNet SF5 alone (experiment 1). But experiment 6 shows that multiview testing reduce IoU performance (-9 points). This implies that multiview generated images are slightly different from original view images and must be seen at training time.

Some qualitative results are shown in Fig. 7. SnapNet-R approach is less sensitive to occlusion situation (see row 5). Besides, looking at the chairs in row 6 shows a much better edge segmentation for SnapNet-R than for FuseNet alone. The poor results of FuseNet (row 1) can be explained by the depth map quality. In this specific situation SnapNet-R takes fully advantage of its inpainting processings.

We also tried to directly train at full resolution (experiment 10 and 11) by picking random crops of 224x224 pixels. Even if this approach is not able to achieve the same performances as the low resolution full frame experiments, we observe the same improvement tendency by using our multi-view data augmentation.

NYUDv2 labeled dataset is extracted from RGB-D video sequences. Our method only generates artificial new data based on a single frame on the contrary of MVCNet method [33] who uses more data with the adjacent video frames for training and testing. Results on NYUDv2 (see Table 2) show the same trend as for SUNRGBD. The SnapNet-R approach reacts well to more than 7 times fewer data at training time. About the 40 classes semantic segmentation task we outmatch the state of the art method MA by +8.77 points and reach the second best score for OA and IoU. Concerning the 13 classes task we achieve new state of the art results, reaching 81.95% OA (+2.82), $77,51\%$ MA (+6.92) and 61.78% IoU (+2.71).

4.4. 3DRMS challenge

As the segmentation task is part of the ongoing challenge, the test ground truth is not available yet. Besides, the small quantity of training data would lead to over-fitting

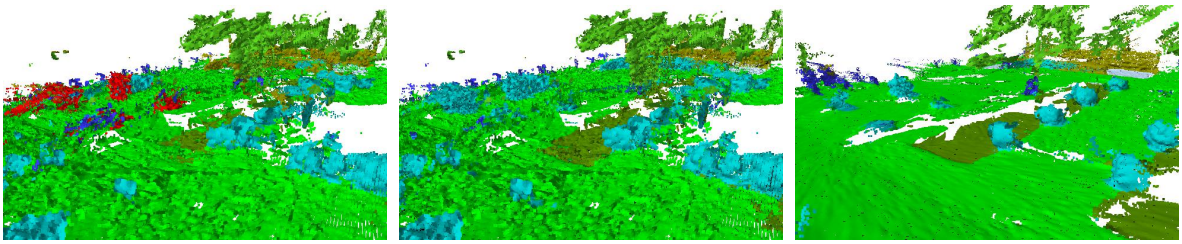


Figure 8. Results of labeled reconstruction for the 3DRMS challenge, **test set**. Left, labels obtained with direct labeling (**Classif 2D**), middle is SnapNet-R applied on stereo reconstruction (**Classif 2D-3D**) and right is SnapNet-R applied on multiview reconstruction (**Classif 3D**).

experiment	OA	MA	IoU
40 classes			
RCNN [17] (RGB-HHA)	60.3	35.1	28.6
FCN 16s [32] (RGB-HHA)	65.4	46.1	34.0
Eigen et al.[12](RGB-D-N)	65.6	45.1	34.1
Context-CRF [31] (RGB-D)	67.6	49.6	37.1
*FuseNet SF3[33] (RGB-D)	66.4	44.2	34.0
*MVCNet-MP [33](RGB-D)	**70.66**	***51.78***	**40.07**
FuseNet SF5 (RGB-D)	62.19	48.28	31.01
SnapNet-R (RGB-D)	***69.20***	**60.55**	***38.33***
13 classes			
Couprie et al.[10] (RGB-D)	52.4	36.2	–
Hermans et al.[24] (RGB-D)	54.2	48.0	–
SceneNet (DHA)[21] (DHA)	67.2	52.5	–
Eigen et al.[12] (RGB-D-N)	75.4	66.9	52.6
*FuseNet SF3 [33] (RGB-D)	75.8	66.2	54.2
*MVCNet-MP [33] (RGB-D)	***79.13***	70.59	***59.07***
Eigen-SF-CRF [35] (RGB-D)	63.6	66.9	–
FuseNet SF5 (RGB-D)	78.41	***72.07***	56.33
SnapNet-R (RGB-D)	**81.95**	**77.51**	**61.78**

** Computed at low resolution (320x240) on the contrary of all other results computed at native resolution.*

Table 2. NYUDv2[40] quantitative results. For each criterion, best values are emphasized in bold, second best values in bold italics. Results are discussed in section 4.3.

if we had practiced cross-validation on them only. We settle then for qualitative results on the testing set with models trained on the training sets.

Reconstruction. The figure 4 compares the point sets generated with stereo data without filters (up left), with outliers removal (up right) and using a global approach (bottom). We have to filter rough data before using SnapNet-R: small objects would not be seen otherwise. However stereo baseline is small and leads to depth estimation errors that are not corrected at accumulation. It results in multiple occurrences of small or thin objects such as tree trunks. In the case of 3DRMS challenge, global approaches provides smoother point cloud with less artifacts than incremental stereo. Thanks to normals estimation it is visible on Figure 8 through shadow which reflects local orientations.

Labeling. Visual results provided in Figure 8 represent the predicted classes with artificial colors. Compared to **Classif 2D** (left), SnapNet-R leads to more consistent labeling (**Classif 2D-3D** middle). This is due to the synthetic cameras which provide views of the point cloud inaccessible to the ground robot. As an example, views looking at the ground with a small incidence angle allows to capture the global geometry of the pavement and ground areas. The experiments also show the robustness of the approach for 3D point clouds. We trained the SnapNet-R on the 2D classification training point clouds. The regularization induced by the CNN and the prediction smoothing with multiple snapshots reduce the imperfect labels due to reconstruction artifacts.

Perspectives. The main perspective concerns the global reconstruction. In this paper, we used the models learned from the **Classif 2D-3D** pipeline. The next step is to train directly on reconstructed data and labels using the 3D classification framework without intermediary steps. As presented in section 3.2 dealing with RGB-D data, the closer the data you train on are to the test data, the better the results are. This gives us a glimpse of improvement for the **Classif 3D** pipeline.

5. Conclusion

We presented in this paper an extension to SnapNet for robotics applications. We changed the core neural network architecture and proposed a snapshot strategy adapted to RGB-D single view data. Using the later as data augmentation improve significantly the segmentation performances on SUNRGBD and NYUDv2, leading to new state of the art results.

We also applied SnapNet-R on the 3D Reconstruction Meets Semantics challenge data. The three pipelines described in the this paper correspond to different use cases from robotic navigation to offline reconstruction. The qualitative results are promising and illustrate the potential of multiview strategy for 3D data labeling.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR*, Washington, DC, USA, 2005.

[2] N. Audebert, B. Le Saux, and S. Lefèvre. Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks. In *ACCV*, Taipei, Taiwan, Nov. 2016.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[4] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[5] A. Boulch, B. Le Saux, and N. Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *Eurographics/3DOR*, Lyon, France, April 2017.

[6] A. Boulch and R. Marlet. Fast and robust normal estimation for point clouds with sharp features. In *Computer graphics forum*, volume 31, pages 1765–1774. Wiley Online Library, 2012.

[7] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[8] A. P. Charaniya, R. Manduchi, and S. K. Lodha. Supervised parametric classification of aerial Lidar data. In *CVPR/W*, pages 30–30. IEEE, 2004.

[9] I. Cherabier, C. Häne, M. R. Oswald, and M. Pollefeys. Multi-label semantic 3d reconstruction using voxel blocks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 601–610. IEEE, 2016.

[10] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, Honolulu, Hawai, USA, July 2017.

[12] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[13] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[14] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2), June 2010.

[15] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

[18] 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*, 2014.

[19] N. Haala, C. Brenner, and K.-H. Anders. 3D urban GIS from laser altimeter and 2D map data. *International Archives Photogramm. Remote Sens.*, 32:339–346, 1998.

[20] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. Semantic3d.net: A new large-scale point cloud classification benchmark. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Science*, IV-1/W1, 2017.

[21] A. Handa, V. Pătrăucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding real world indoor scenes with synthetic data. In *CVPR*, Las Vegas, NV, USA, 2016.

[22] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In *ACCV*, 2016.

[23] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In *Proc. ACCV*, volume 2, 2016.

[24] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2631–2638. IEEE, 2014.

[25] 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 *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*. 2013.

[26] J. Jiang, Z. Zhang, Y. Huang, and L. Zheng. Incorporating depth into both cnn and crf for indoor semantic segmentation. *arXiv preprint arXiv:1705.07383*, 2017.

[27] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[28] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[29] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3D scene labeling. In *ICRA*, pages 3050–3057. IEEE, 2014.

[30] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *European Conference on Computer Vision*, pages 541–557. Springer, 2016.

[31] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Exploring context with deep structured models for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[33] L. Ma, J. Stueckler, C. Kerl, and D. Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. In *arXiv:1703.08866*, Mar 2017.

[34] D. Maturana and S. Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *IROS*, pages 922–928, Hamburg, Germany, 2015.

[35] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4628–4635. IEEE, 2017.

[36] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[37] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, Munich, 2015.

[38] R. B. Rusu, A. Holzbach, N. Blodow, and M. Beetz. Fast geometric point labeling using conditional random fields. In *IROS*, pages 7–12. IEEE, 2009.

[39] K. Sfikas, T. Theoharis, and I. Pratikakis. Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval. In *Eurographics Workshop on 3D Object Retrieval*, Lyon, France, 2017.

[40] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, Florence, Italy, 2012.

[41] S. Song, S. P. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, pages 567–576, Boston, USA, 2015.

[42] H. Su, , C. Qi, K. Mo, and L. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, Honolulu, Hawai, USA, July 2017.

[43] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *ICCV*, pages 945–953, 2015.

[44] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*, pages 356–369, Hersonissos, Crete, 2010. Springer.

[45] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, Boston, USA, 2015.

[46] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision*, pages 1625–1632, Sidney, Australia, 2013.