MV-KPConv: Multi-view KPConv For Enhanced 3D Point Cloud Semantic Segmentation Using Multi-Modal Fusion With 2D Images

C. Du[1], M. A. Vega Torres[1], Y. Pan[1,2] & A. Borrmann[1,2]
[1]Technical University of Munich, Chair of Computational Modeling and Simulation, Munich, Germany
[2]Technical University of Munich , Institute for Advanced Study, Munich, Germany

ABSTRACT: Compared with unimodal deep learning algorithms that directly process 3D point clouds, multi-modal fusion algorithms that leverage 2D images as supplementary information have performance advantages. In this work, the performance of an open-source multimodal algorithm, MVPNet, is improved on the 3D semantic segmentation task by using KPConv as a more robust 3D backbone. Different modules of the two networks are meaningfully combined: the 2D-3D lifting method provided by MVPNet aggregates selected 2D image features into 3D point clouds, then KPConv is used to fuse these features with geometric information to make predictions. On a ScanNet sub dataset, the proposed network significantly outperforms the original MVPNet and KPConv regardless of the fusion structure. By integrating COLMAP into the workflow, we further extend the proposed method to a custom dataset. The results show the improved performance of our multimodal fusion algorithm in identifying relevant categories of objects in the 3D scene.

1 INTRODUCTION

In the construction industry and related research sectors, the concept of Digital Twin is becoming increasingly essential. It aims to bring the building model, object information, and data received from sensors and actuators together into one representation, generating a digital duplicate of the physical environment and processes (Wahbeh *et al.*, 2020). In order to capture the current physical state of the built environment, 3D point clouds can be utilized to depict the precise details of the as-is physical environment (Tan Qu and Wei Sun, 2015). More speficially, Scan-to-BIM describes approaches that interpret generated point clouds and create a valid as-built BIM model from them (Braun, 2020), which has been brought to the process of generating digital twins. However, original point clouds do not provide any semantic information. For any further automated assessment, the 3D point cloud needs to be processed by deep learning and/or with feature engineering techniques to obtain useful semantic data.

Traditional deep learning algorithms for 3D scene recognition usually use only a single modality such as a point cloud as input. Based on the representation of point cloud, the deep learning-based point cloud processing approaches can be briefly divided into three categories: voxel-based, point-based and multi-view-based. The voxel-based approaches partition the point cloud into fixed-resolution 3D grids with a discrete regular data structure. However, the voxel-based methods require high memory consumption due to the sparsity of the voxels. Some spatial resolution and fine-grained 3D geometry information can also be lost during voxelization (Cui *et al.*, 2021). Point-based approaches process point clouds directly without transforming them into an intermediate data representation (Xie, Tian and Zhu, 2020). A pioneering deep learning network in this direction is PointNet (Qi, Su, *et al.*, 2017). It uses a shared Multi-Layer Perceptron (MLP) to process individual points for per-point feature extraction. But applying the point-based methods directly on a massive point cloud can be time-consuming and memory-expensive. The traditional multi-view-based methods try to represent the 3D point cloud by multi-view 2D images, which can then be processed by standard 2D convolution (Chen *et al.*, 2017). However, the performance of this kind of method was not satisfactory. The main reason is that the approximate 2D projection leads to the loss of the geometric structure.

In summary, employing these unimodal algorithms to convert point clouds into other representations or directly process them, usually results in data loss or unsatisfactory performance. In addition, such algorithms understand 3D scenes mainly through the geometric information provided by the point cloud, but not all objects can be distinguished by their 3D shape, especially if they have flat surfaces such as windows or are richly textured.

To solve this problem, additional color and texture information needs to be exploited. This makes

multimodal fusion networks using 2D images as supplementary information more advantageous compared to unimodal networks. Generally speaking, multi-modal fusion gathers rich characteristics of complicated scenarios from various sensors and integrates them to gain more spatial and contextual information for robust, accurate, and fast scene understanding (Zhang *et al.*, 2020). Particularly, we want to employ mature 2D CNN to extract semantic features of 2D images to enrich the expression of point clouds. Multi-modal fusion methods derive from existing unimodal algorithms. For example, the voxel- or point-based unimodal approaches can be chosen as the backbone network for processing data in a holistic or segregated manner.

2 RELATED WORKS

In recent years, multiple advanced deep multi-modal fusion algorithms for 3D scene understanding were developed. Classic examples in the field of 3D object detection are Frustum-PointNet (Qi *et al.*, 2018), Frustum-ConvNet (Wang and Jia, 2019) and SIFR-Net (Zhao *et al.*, 2019). The idea behind this kind of result-level fusion algorithms is to limit the 3D search space for object detection by using the results of off-the-shelf 2D object detectors, thus reducing the computational effort and improving the runtime. However, the information from images is not fully leveraged. Unlike them, the EPNet (Huang *et al.*, 2020) processes point clouds and images with two separate networks and gradually fuses image semantics and point features with an adaptive fusion module. 3D-CVF (Yoo *et al.*, 2020) is proposed to combine two heterogeneous feature maps in the bird's eye view (BEV) domain and did a total of two fusions to further improve the accuracy.

In the field of 3D semantic segmentation, Virtual MV-Fusion (Kundu *et al.*, 2020) employed synthetic images generated from virtual viewpoints of the 3D scene rather than processing raw photographic images captured by a physical camera, which allows them to freely choose the camera parameters that are most effective for the 2D semantic segmentation. The semantic features of the rendered images will then be fused in the 3D domain. The unified point-based framework proposed by Chiang et al. (2019) also uses 2D CNN to learn semantics from rendered images. The 2D features are projected onto two point sets that are sampled to varying resolutions. Then they use two different encoders to extract local geometric features and global context from the fused data, and finally employ a decoder to combine the features by interpolation. Unlike this type of approach using rendered images, MVPNet (Jaritz, Gu and Su, 2019) proposes a more flexible 2D-3D lifting method to transfer image semantics to point cloud. First, a small number of images that can accurately cover a given point cloud are selected in real-time by their proposed algorithm. The semantic features extracted from these images through a 2D network are then transferred to the 3D point cloud through an end-to-end feature aggregation module. Finally, PointNet++ (Qi, Yi, *et al.*, 2017) is applied to fuse the features in 3D canonical space to predict 3D semantic labels.

As one of the open-source multi-modal algorithms on the ScanNet benchmark, MVPNet has the potential to continuously improve and extend its performance. Firstly, Pointnet++ being a relatively old 3D network may limit the performance of the overall model. Since the final output of the 2D-3D lifting method of MVPNet is 3D point-like features, it is possible to process these features using a different and more powerful point-based 3D network. Secondly, the original MVPNet only provided a solution for processing standard RGB-D datasets, such as ScanNet, where images and 3D scenes were already aligned and each image had a corresponding depth map and camera parameters. In the Scan2BIM process, however, we usually use the 3D point cloud data captured by laser scanners for higher accuracy. Therefore, and in order to enhance the semantic segmentation of 3D laser-scanned point clouds in practical application scenarios, we need a flexible method to acquire and process not only 3D but also 2D data.

To address these issues, we employed the better-performing KPConv (Thomas *et al.*, 2019) to replace PointNet++ as a 3D network to further improve the performance of MVPNet. Also, by integrating COLMAP (Sch and Z, 2016) into the entire workflow, our proposed MV-KPConv can be applied in laser-scanned point clouds along with images, which fit the practical Scan2BIM application scenarios.

3 METHODOLOGY

3.1 *Overview*

An overview of the proposed approach is illustrated in Figure 1. The method starts with preprocessing stage. In an initial step, point clouds are subsampled to reduce the computational cost. Then the overlap of each image with the point cloud is computed in order to determine the coverage area of each image in the scene.

During the data loading phase, spherical subclouds are selected from the scene point cloud as input to the 3D network. Using the overlap information provided in the previous stage, a certain number of images that can cover the input sphere well will be selected instantly. The pixels with a valid depth of these images are projected into 3D space to form a

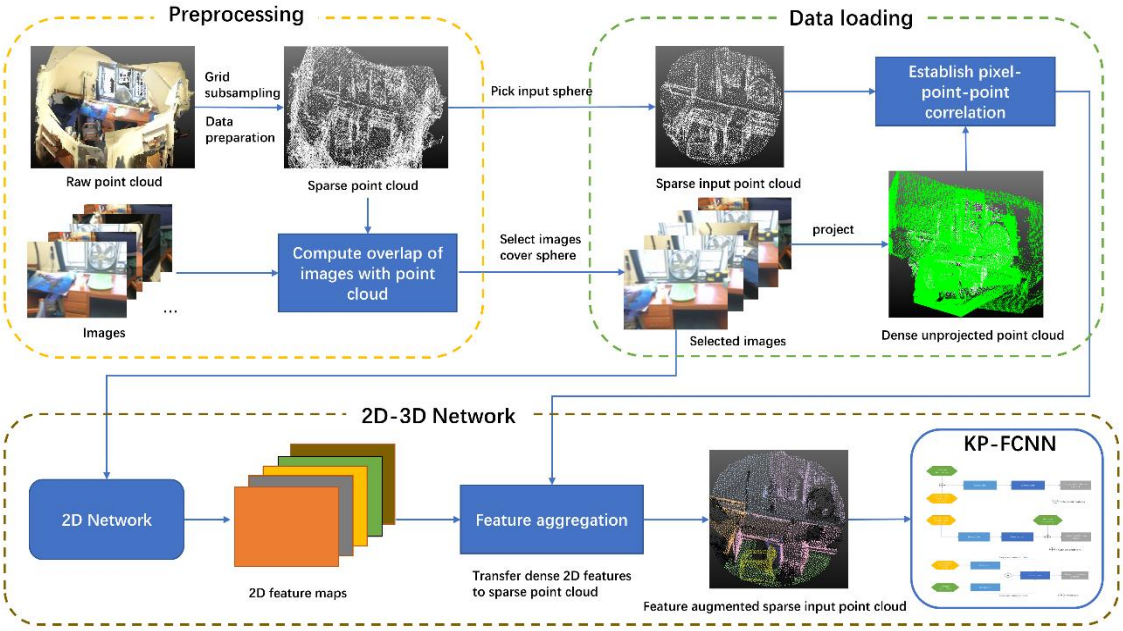


Figure 1 Overview of proposed method. For the 3D laser-scanned data, we use a camera to take 2D pictures in the corresponding 3D scene and employ COLMAP to predict the depth maps and camera parameters that are used to project the pixels into 3D domain. In KP-FCNN we designed three different fusion structures to find the optimal fusion timing.

dense point cloud. We then establish a correlation between the image pixels, the dense projected point cloud, and the sparse input point cloud using the K-Nearest Neighbors (KNN) algorithm.

Finally, the selected images are fed into a pre-trained 2D network to obtain feature maps of the same size as the input images. The feature aggregation module receives these feature maps associated with the dense point cloud, as well as the sparse spherical sub-clouds as inputs, and then transfers 2D semantic features from the dense point cloud to each input point through the previously established KNN correlations. Finally, the sparse point cloud augmented by the image features is fed into KP-FCNN to fuse with the geometric information. Using this fusion information, KP-FCNN will predict the 3D labels and perform the semantic segmentation task. Three different fusion architectures are designed and compared in KP-FCNN.

3.2 *Preprocessing*

3.2.1 *Point cloud sampling*

Point clouds of real scenes usually show different densities. Different densities affect the results and efficiency of point cloud processing methods. The subsampling method balances the point cloud density by ensuring a certain average data spacing. In addition, it is a first step towards reducing computational costs, as the number of points is greatly reduced. In our method, grid subsampling provided by KPConv is used as the sampling strategy. It projects the point cloud into a 3D grid, where each voxel retains only one point closest to the barycenter of the voxel. This method is fast and allows the sampling density to be easily controlled by the size of the voxels.

3.2.2 *Point cloud – Image overlap*

Although the point cloud has been subsampled before being fed into the network, the scene is still too large to be processed as a whole. Therefore, KPConv actually takes spherical sub-clouds in the scene and processes them. Due to this reason, we need to know which region of the scene is covered by each image, so that we can accurately select the images that maximize coverage of the input sphere during data loading.

Similarly as done by Jaritz et al. (2019) the following steps are performed to compute the coverage regions. Firstly, 6000 base points are randomly selected in the sampled point cloud. After that, a KD-tree is created for these base points for later neighborhood search. Then, all RGB images belonging to this scene are looped over. Each image will be projected into the 3D domain using its depth map and camera parameters. If using images captured by ourselves, a correction matrix will also be required here (see section 4.1). For each point in the projected point cloud, we find its nearest neighbor (only one and up to 1 cm away) in the base point set and save the mapping information. After these steps we can know which area of the original point cloud is covered by this image by knowing which base points the image can cover.

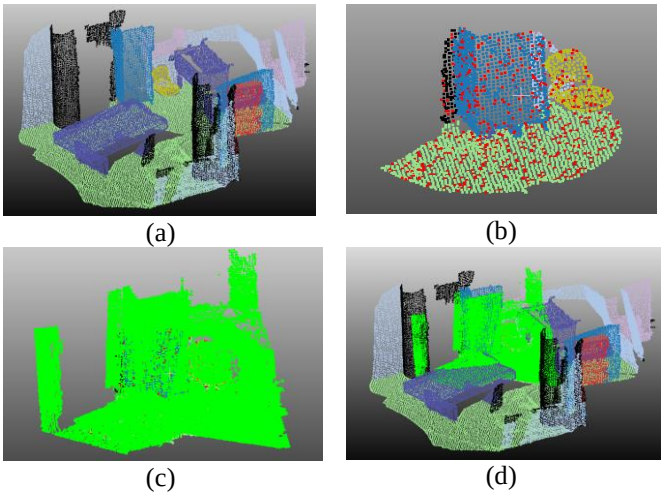
3.3 *Data loading*

3.3.1 *The picking strategy*

In order to have every region of the scene sampled evenly, we use a potential picking strategy provided by KPConv to pick the input spheres. In simple terms, this solution is to assign a potential value at each sphere center, which are obtained by continuously subsampling the point cloud. Whenever the network picks a sphere, the potential values of all coarse points (sphere centers) included in this sphere radius are increased so that we know this region has been selected. The potential value is increasing as a Gaussian function (the center increases the most and the increasing value decreases with distance). The next sphere will always be chosen in the region of lowest potential so that each part of the dataset will be selected approximately the same number of times. Similarly as done by Thomas (2020), the spatial regularity of the picking can be ensured by tracking the potential value of each coarse point in this way.

3.3.2 *View selection*

During data loading, the overlap information obtained in chapter 3.2.2 is used to select the RGB images on-the-fly with a greedy algorithm (Jaritz et al., 2019). Knowing which base points are included in the input sphere, the overlap information can be further filtered. This filtering is needed to know by which images these base points can be covered. The image that covers the largest number of base points is selected first. After one selection is done, all the base points covered by this image are set to invalid and then the next round of image selection is started until 5 images are selected.



(a) (b) (c) (d)

Figure 2 View selection procedure (a) Complete point cloud scene. (b) Input spherical subcloud. The red dots are the base points inside it. (c) Dense point cloud formed by the projection of the 5 selected views. (d) Dense point cloud in the full scene view. It can be seen that the selected 2D views cover the input subcloud very well.

3.3.3 *The pixel-point-point correlation*

In order to use information from selected images, we also need to establish a correlation between pixels and points during data loading. The pixel-point-point association here can be understood as follows: "pixel-point" can be explained as the pixels in image having a mapping relationship to a point in the dense point cloud generated by its projection. This means that the semantic features of a pixel on the image feature map can be interpreted as features of a point on the dense point cloud, as long as the feature map and the image have the same size. The "point-point" represents the KNN correlation between the projected point cloud and input point cloud. For each point in the input point cloud, we find its K nearest neighbors in the dense point cloud and save their indices. This allows point features on the dense point cloud to be transferred to the input point cloud.

3.4 *The Network*

3.4.1 *The 2D network*

The 2D network architecture from the original MVPNet was adopted. The backbone of the encoder network is an ImageNet-pretrained ResNet34 (He *et al.*, 2016) with batch normalization and dropout. The decoder network is a lightweight variant of U-Net (Ronneberger, Fischer and Brox, 2015). Here, the convolution is used to fuse concatenated features from skip connections, and the transposed convolution is applied for upsampling. Batch normalization and ReLU are attached after each convolution layer. This 2D network was pretrained first on the task of 2D semantic segmentation on a ScanNet sub dataset, and then integrated into the whole pipeline with frozen weights. The feature map output by the 2D network has 64 channels of semantic features. Furthermore, the size of the output feature map H×W is equal to that of the input image. This makes subsequent 2D-3D feature lifting possible (Jaritz, Gu and Su, 2019).

3.4.2 *Feature aggregation module*

The feature aggregation module is the core module of the 2D-3D lifting method proposed by MVPNet. Using the pixel-point-point correlation, this module is able to transfer 2D semantic features to 3D points by

$$F_i = \sum_{j \in N_K(i)} MLP(concatenate[f_j, f_{dist}(x_i, x_j)]) \quad (1)$$

where f_j represents the 2D feature of a point x_j on dense point cloud. This point x_j is one of the K nearest neighbors of a point x_i in the input point cloud. f_j is first concatenate with a distance feature. The distance feature expresses the distance relationship between x_j and x_i which can be defined as:

$$f_{dist}(x_i, x_j) = concatenate[x_i - x_j, \| x_i - x_j \|^2] \quad (2)$$

After that, the connected feature is fed into a three-layered MLP. Jaritz et al. (2019) argued that the MLP can transform 2D image features to an embedding space more consistent with the 3D representation. Finally, the features of these K neighbor points (in our case K = 3) are aggregated by a sum operation and become 64-channel semantic features F_i, which will be fed into KP-FCNN together with points.

The feature aggregation module is differentiable and the weights inside the MLP need to be learned and updated by back propagation of the network. And since this module has no loss function of its own, its internal weights are adjusted by the loss function of the 3D network, i.e., KP-FCNN.

3.4.3 *KP-FCNN*

Kernel Point Convolution is a novel point convolution operator inspired by image-based convolution, but instead of kernel pixels, a set of kernel points are utilized to define the spherical area where each kernel weight is applied. The distribution of kernel points within the sphere can be defined in advance, i.e., the rigid kernel, or be learned by network to adapt local geometry of point cloud, i.e., the deformable kernel (Thomas *et al.*, 2019). Based on the KPConv operator, KP-FCNN designed for the segmentation tasks demonstrates strong performance, outperforming PointNet++ on multiple benchmarks, e.g., S3DIS (Armeni *et al.*, 2017), ScanNet etc.

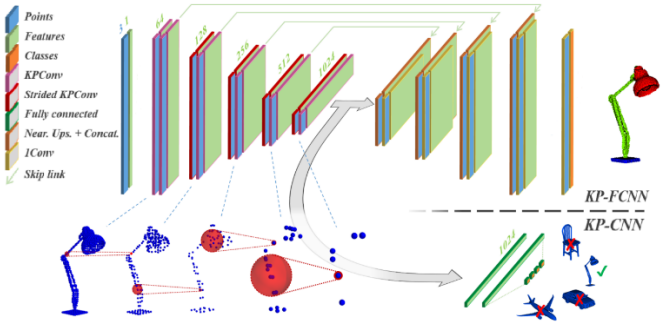


Figure 3 Demostration of two network architecture base on KPConv (Thomas *et al.*, 2019)

The architecture of the KP-FCNN in this study basically follows the standard structure presented in figure 3. The encoder part still consists of five layers, but except for the initial layer, a standard KPConv block has been added to the remaining layers, making the network structure deeper and thus better able to extract features. Based on this module, three fusion architectures in figure 4 were implemented to investigate the impact of fusion timing on the proposed network.

In the early fusion variant, the geometric features are concatenated with image features output by the feature aggregation module and fed into the KP-FCNN. In the late fusion, geometric features are passed through the encoder and decoder of KP-FCNN and then connected with image features in front of the segmentation head. In middle fusion, geometric features and image features are passed through sperate encoders and then averaged and fused before the decoder. Consistent with the standard architecture, the skip connection is also used to pass the features from the middle layers of the encoder to the decoder. The only difference is that the upsampled features in the decoder are concatenated with features from both encoders.

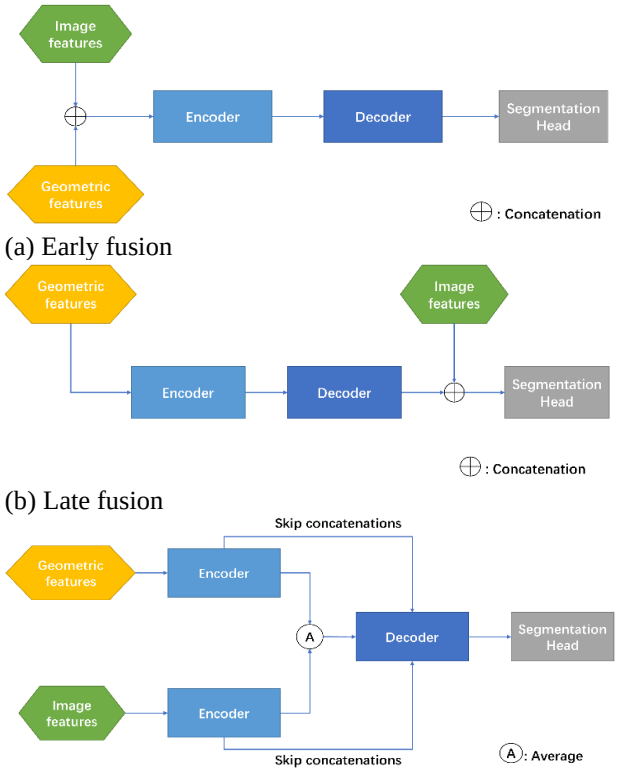


Figure 4 Three fusion architectures base on KP-FCNN

4 RESULTS AND ANALYSIS

4.1 *Datasets*

The ScanNet dataset (Dai *et al.*, 2017) covers various indoor scenes such as offices, meeting rooms, etc., with a total of 2.5 million frames collected using a depth sensor attached to iPad Air2. The complete dataset contains 1201 training scenes and 312 validation scenes. Each scene's data contains an RGB-D sequence, corresponding camera postures, and a whole scene mesh annotated with 20 semantic classes. Due to disk capacity limitations and because doors and windows are more generic and common objects for interior scenes, we only selected scenes containing these two classes from all training and validation sets. This sub-dataset containing 118 training scenes and 28 validation scenes was used in this study.

In order to test our method in real scenarios, we also constructed a custom dataset using scenes, captured in the Technical University of Munich. We used a laser scanner to scan and obtain 3D point clouds of two office rooms and captured an average of 200 photos of each room using a camera. These photos were

fed into COLMAP to obtain depth maps, camera intrinsic/extrinsic of these color images and a reconstructed point cloud. COLMAP is a general-purpose, open-source Structure of Motion (SfM) and Multi-View Stereo (MVS) pipeline. For the reconstruction of both ordered and unordered image collections, it provides a wide range of capabilities. By aligning the reconstructed point cloud with the laser-scanned point cloud, we obtain a correction matrix for the photos. This information is used later in the data loading session to ensure the accuracy of the pixel projection. We use the same semantic classes as ScanNet to label the point clouds.

4.2 *Implementation details*

The network implementation is adapted from KPConv and MVPNet. All experiments were run on a workstation with a NVIDIA 5GB Quadro P2000 GPU and 32 GB RAM. The size of grid voxels in grid subsampling was set to 4 cm. The input sphere radius is chosen as 1.2 m. Multiple experiments showed that the network needed roughly 400 - 450 epochs to converge, so the maximum epoch was set to 500. The batch size of input point clouds was set to 5. Each epoch contains 500 steps, which means that 2500 spheres are processed per epoch. The remaining parameter settings follow the default settings of KPConv. As for the 2D part, 5 views are selected for each input sphere during training and validation. During the training of 2D network, an SGD optimizer with a momentum 0.9 and a weight decay of 0.0001 was used. The image batch size was set to 32 and the network was iterated 80,000 times.

4.3 *Results*

The table 1 summarizes the performance of several fusion structures with the rigid or deformable kernel. The proposed network is named as Multi-view-KPConv, i.e., MV-KPConv. The original MVPNet and KPConv are applied as the baseline models here. To be fair, the original MVPNet also uses 5 input images. The tests were done on a validation set consisting of 28 scenarios.

All fusion structures in the Table 1 fuse one channel geometric features (Z-coordinate of points) with 64-channel image features. Regardless of the fusion structure and kernel types, the proposed MV-KPConv's mIoU scores exceeded both baseline models. Comparing the different kernel types, we can find that rigid kernel generally performs better than deformable kernel on ScanNet dataset. This is largely consistent with what is reported in the KPConv. Comparing different fusion structures, we can see that the early fusion using rigid kernel has the best performance. This also confirms the advantage of early fusion, which allows the network to fully exploit the information of the raw data.

Table 2 shows the segmentation scores of the proposed method on 3D laser-scanned data and figure 5 visualizes the prediction results. We use the best models trained on the ScanNet sub-dataset to perform inference on our custom dataset. It can be seen that MV-KPConv, which uses 2D images as additional information, has an advantage over KPConv, which uses point cloud data alone, in recognizing objects with flat and glassy surfaces or rich textures, such as windows, bookshelves and pictures on the wall.

Table 1 Semantic segmentation IoU scores on ScanNet sub-dataset

Network	Kernel	mIoU	wall	floor	cabi-net	bed	chair	sofa	table	door	win-dow	book-shelf	pic-ture	coun-ter	desk	cur-tain	fridge	show er	toilet	sink	bath	other
MVPNet	-	71.2	83.2	92.8	59.8	84.2	87.8	84.0	74.4	81.1	79.6	93.7	11.4	**95.5**	72.2	40.6	85.6	20.7	86.7	52.0	**68.1**	81.0
KPConv*	rigid	52.6	73.1	92.1	46.1	71.2	81.7	53.2	57.5	38.0	53.9	63.9	3.8	60.8	62.2	15.3	5.5	20.3	88.1	46.6	74.3	44.0
MV-KPConv (Early fusion)	rigid	**74.4**	**86.0**	**93.5**	60.4	**91.2**	**90.2**	83.5	74.9	**83.1**	79.7	95.2	10.9	84.7	74.0	49.0	**88.7**	**44.0**	**90.5**	**56.5**	66.6	**85.6**
	deform	72.9	85.7	93.4	61.5	91.5	89.7	80.6	**77.2**	81.0	79.5	94.7	12.0	84.4	**75.2**	45.3	78.5	43.0	83.2	52.6	63.1	85.4
MV-KPConv (Middle fusion)	rigid	73.7	85.6	**93.5**	60.9	88.9	89.2	83.7	76.2	82.5	**81.1**	**95.2**	11.0	86.5	74.5	**51.6**	87.1	40.1	87.7	55.2	59.8	84.4
	deform	72.3	85.6	93.4	57.5	90.0	89.8	81.8	73.0	81.6	78.5	95.0	11.4	86.2	71.6	44.4	84.5	39.8	84.3	54.4	61.8	82.1
MV-KPConv (Late fusion)	rigid	72.2	85.3	93.1	**62.1**	88.3	90.1	**84.3**	72.4	79.8	80.5	92.9	11.7	85.4	74.6	39.4	84.6	35.3	85.9	55.7	59.7	82.5
	deform	71.5	84.6	**93.5**	57.7	89.7	89.1	83.3	75.5	78.6	78.6	93.1	**12.4**	80.5	72.7	41.4	86.0	34.1	83.1	54.8	60.6	80.8

* Only used Z as additional geometric feature

Table 2 Semantic segmentation IoU scores on our custom dataset

Network	mIoU	wall	floor	cabi-net	chair	sofa	table	door	win-dow	book-shelf	pic-ture	coun-ter	desk	other
MV-KPConv	**45.1**	77.9	**95.0**	13.4	**69.5**	0.0	32.7	34.8	**57.8**	**91.4**	**37.4**	0.0	25.3	**51.4**
KPConv	42.4	**79.3**	74.1	**19.4**	30.0	**93.4**	**52.5**	**36.6**	5.7	83.3	6.0	0.0	**55.1**	16.3

4.4 *Computational time*

Table 3 compares the computation times of different versions of MV-KPConv and MVPNet on ScanNet data. Both networks spend a lot of time in the preprocessing stage to calculate the overlap of images and point clouds. In the training phase, it can be seen that although MV-KPConv has better semantic segmentation capability, the time required and the volume of the model are much larger than that of MVPNet. This is in view of the fact that KPConv has a deeper and more complex structure than PointNet++. The late fusion takes slightly more time than early fusion. This is because an additional small linear transformation layer is added to the end of decoder in order to make the shape of the fused features acceptable to the segmented head. Moreover, due to the use of two encoders, the intermediate fusion structure has a much larger number of parameters than the other two structures, which makes the training time required much higher.

Table 3 Runtime and model size based on ScanNet data

Step	Network	Model size [MB]	Computational time [h]
Preprocessing	MVPNet	-	2.4
	MV-KPConv	-	3.3
Training	2D Network	180	16
	MVPNet	282	5
	KPConv	186	28
	MV-KPConv *rigid* (Early fusion)	457	47
	MV-KPConv *rigid* (Middle fusion)	500	56
	MV-KPConv *rigid* (Late fusion)	457	48
Inference	MVPNet	282	0.13
	MV-KPConv *rigid* (Early fusion)	457	0.25

4.5 *Discussion*

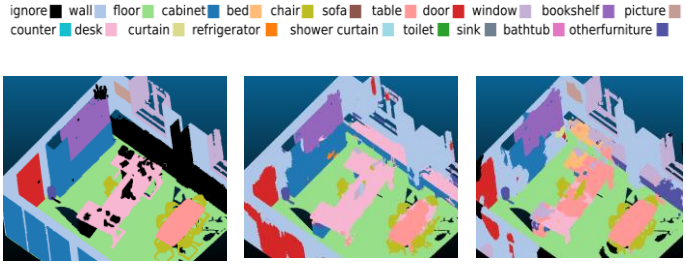
To analyze the design choices and to better understand the network characteristics, we further conducted some ablation experiments on the ScanNet dataset. To understand whether fusing 3D geometric features is really beneficial for the network, or whether 2D image features are sufficient, we tried fusing image features and geometric features completely (XYZ), partially (Z) and not at all. The results show that the MV-KPConv does benefit from the complementary information provided by the geometric features (height) compared to the use of image features alone. Interestingly, good results can be obtained using only Z coordinates, and XY coordinates seem to be useless to the network. This can be explained by the fact that X and Y do not make any sense in a dataset where the orientation of the objects in the scene can be in any direction. On the contrary, the Z value is the height of the point and has a very valuable geometric meaning.

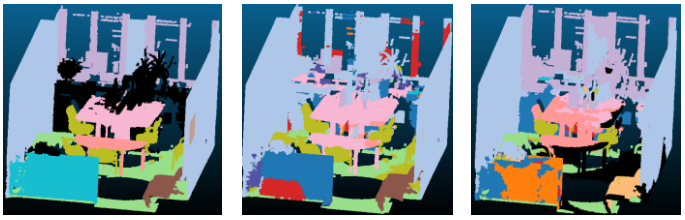
In addition, we also investigate the effect of point cloud color by fusing 3D RGB with input features. However, the use of point cloud colors has a negative effect on the network. This can be explained by the fact that the 2D network has already processed the color information, and it is confusing to pass the unprocessed color information to the 3D network again at this time. Moreover, the possible conflict between the image color and the point cloud color can lead to wrong predictions.

5 CONCLUSION AND FUTURE WORKS

In this paper, we successfully improve the performance of the multi-modal fusion algorithm MVPNet on 3D semantic segmentation tasks by using KPConv instead of PointNet++ as a better 3D backbone. We investigate the impact of different fusion structure designs on the network and further improve the performance of the network by choosing a suitable fusion structure. For practical Scan2BIM application scenarios, we propose a flexible scheme using a photogrammetric technique to obtain 2D supplementary information of 3D scenes. While it might be possible to use images that some laser scanners provide, using an external camera provides the flexibility of having different camera positions which contributes to avoid occlusions and capture more features of the environment. On our dataset, we demonstrate that multi-modal fusion algorithms have performance advantages over traditional unimodal algorithms for 3D scene understanding, particularly in recognizing the objects with flat and glassy surfaces or rich textures.

In addition, we found in our study that since SfM estimates depth by geometric constraints and feature correspondence between image sequences, the depth maps it predicts are usually sparse. This may result in some useful pixel features on color images not being projected onto the point cloud, thus affecting the prediction performance. A possible approach is to use deep neural networks to perform end-to-end dense depth map estimation on individual images. Project page: https://github.com/dcy0577/Enhancing-3D-Point-Cloud-Segmentation-Using-Multi-Modal-Fusion-with-2D-Images





Ground truth KPConv MV-KPConv

Figure 5 Qualitative results on our custom dataset. Our method has a great advantage in identifying windows.

6 REFERENCES

Armeni, I. *et al.* (2017) 'Joint 2D-3D-Semantic Data for Indoor Scene Understanding'. Available at: http://arxiv.org/abs/1702.01105.

Braun, A. (2020) 'Automated BIM-based construction progress monitoring by processing and matching semantic and geometric data', (November), p. 150.

Chen, X. *et al.* (2017) 'Multi-view 3D object detection network for autonomous driving', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 6526–6534. doi: 10.1109/CVPR.2017.691.

Chiang, H. Y. *et al.* (2019) 'A Unified Point-Based Framework for 3D Segmentation', *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019*, (Table 2), pp. 155–163. doi: 10.1109/3DV.2019.00026.

Cui, Y. *et al.* (2021) 'Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review', *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19. doi: 10.1109/TITS.2020.3023541.

Dai, A. *et al.* (2017) 'ScanNet: Richly-annotated 3D reconstructions of indoor scenes', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 2432–2443. doi: 10.1109/CVPR.2017.261.

He, K. *et al.* (2016) 'Deep residual learning for image recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 770–778. doi: 10.1109/CVPR.2016.90.

Huang, T. *et al.* (2020) 'EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12360 LNCS, pp. 35–52. doi: 10.1007/978-3-030-58555-6_3.

Jaritz, M., Gu, J. and Su, H. (2019) 'Multi-view pointnet for 3D scene understanding', *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pp. 3995–4003. doi: 10.1109/ICCVW.2019.00494.

Kundu, A. *et al.* (2020) 'Virtual Multi-view Fusion for 3D Semantic Segmentation', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12369 LNCS, pp. 518–535. doi: 10.1007/978-3-030-58586-0_31.

Qi, C. R., Su, H., *et al.* (2017) 'PointNet: Deep learning on point sets for 3D classification and segmentation', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 77–85. doi: 10.1109/CVPR.2017.16.

Qi, C. R., Yi, L., *et al.* (2017) 'PointNet++: Deep hierarchical feature learning on point sets in a metric space', *Advances in Neural Information Processing Systems*, 2017-Decem, pp. 5100–5109.

Qi, C. R. *et al.* (2018) 'Frustum PointNets for 3D Object Detection from RGB-D Data', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 918–927. doi: 10.1109/CVPR.2018.00102.

Ronneberger, O., Fischer, P. and Brox, T. (2015) 'U-Net: Convolutional Networks for Biomedical Image Segmentation', in Navab, N. et al. (eds) *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*. Cham: Springer International Publishing, pp. 234–241.

Sch, J. L. and Z, T. H. (no date) 'Structure-from-Motion Revisited'.

Tan Qu and Wei Sun (2015) 'Usage of 3D Point Cloud Data in BIM (Building Information Modelling): Current Applications and Challenges', *Journal of Civil Engineering and Architecture*, 9(11), pp. 1269–1278. doi: 10.17265/1934-7359/2015.11.001.

Thomas, H. *et al.* (2019) 'KPConv: Flexible and deformable convolution for point clouds', *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob, pp. 6410–6419. doi: 10.1109/ICCV.2019.00651.

Thomas, H. (2020) 'Learning new representations for 3D point cloud semantic segmentation To cite this version : HAL Id : tel-02458455 Apprentissage de nouvelles représentations pour la sémantisation de nuages de points 3D Learning new representations for 3D point cloud sema'.

Wahbeh, W. *et al.* (2020) 'Digital twinning of the built environment-an interdisciplinary topic for innovation in didactics', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5(4), pp. 231–237. doi: 10.5194/isprs-Annals-V-4-2020-231-2020.

Wang, Z. and Jia, K. (2019) 'Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal', *IEEE International Conference on Intelligent Robots and Systems*, pp. 1742–1749. doi: 10.1109/IROS40897.2019.8968513.

Xie, Y., Tian, J. and Zhu, X. X. (2020) 'Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation', *IEEE Geoscience and Remote Sensing Magazine*, 8(4), pp. 38–59. doi: 10.1109/MGRS.2019.2937630.

Yoo, J. H. *et al.* (2020) '3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-view Spatial Feature Fusion for 3D Object Detection', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12372 LNCS, pp. 720–736. doi: 10.1007/978-3-030-58583-9_43.

Zhang, Y. *et al.* (2020) 'Deep multimodal fusion for semantic image segmentation : A survey To cite this version : HAL Id : hal-02963619 Deep Multimodal Fusion for Semantic Image Segmentation : A Survey'.

Zhao, X. *et al.* (2019) '3D object detection using scale invariant and feature reweighting networks', *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 9267–9274. doi: 10.1609/aaai.v33i01.33019267.