LinkNet: 2D-3D linked multi-modal network for online semantic segmentation of RGB-D videos

Jun-Xiong Cai[a], Tai-Jiang Mu[a], Yu-Kun Lai[b], Shi-Min Hu[a]

[a]*Department of Computer Science and Technology, Tsinghua University, Fit 3?524, Beijing, China*
[b]*School of Computer Science and Informatics, Cardiff University, Wales, United Kingdom*

Abstract

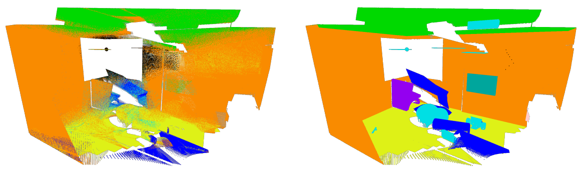
This paper proposes LinkNet, a 2D-3D linked multi-modal network served for online semantic segmentation of RGB-D videos, which is essential for real-time applications such as robot navigation. Existing methods for RGB-D semantic segmentation usually work in the regular image domain, which allows efficient processing using convolutional neural networks (CNNs). However, RGB-D videos are captured from a 3D scene, and different frames can contain useful information of the same local region from different views. Working solely in the image domain fails to utilize such crucial information. Our novel approach is based on joint 2D and 3D analysis. The online process is realized simultaneously with 3D scene reconstruction, from which we set up 2D-3D links between continuous RGB-D frames and 3D point cloud. We combine image color and view-insensitive geometric features generated from the 3D point cloud for multi-modal semantic feature learning. Our LinkNet further uses a recurrent neural network (RNN) module to dynamically maintain the hidden semantic states during 3D fusion, and refines the voxel-based labeling results. The experimental results on SceneNet [1] and ScanNet [2] demonstrate that the semantic segmentation results of our framework are stable and effective.

Keywords: scene understanding, RGB-D image segmentation, online semantic segmentation, multi-modal learning

1. Introduction

Online scene understanding of RGB-D videos, i.e., recognizing semantic objects when RGB-D frames are being received, is essential for intelligent robot and autonomous driving. At present, most works regard the online semantic understanding task as the semantic segmentation of individual image frames. There have been many semantic segmentation methods designed for 2D images based on deep convolutional neural networks (DCNNs) [3, 4, 5, 6]. However, recognition on single frame would be easily affected by environment changes, such as distance, texture and lighting, resulting in unstable semantic segmentation results during the movement. As shown in Fig. 1, directly fusing semantic segmentation results of RGB-D images into the 3D point cloud results in significant ambiguities and inconsistencies, leading to poor segmentation performance. This is because the color input keep changing during the movement of camera, resulting in inconsistent global features across frames.

In recent years, depth has become a common additional input for RGB images with the development of range sensors. This additional modality provides geometric details, which are beneficial to supplement the color information [7]. Directly regarding the depth as an extra input channel for the deep neural network in addition to the RGB has been proved to be less effective [8, 3]. Besides, various visual SLAM (Simultaneous Localization and Mapping) works [9, 10, 11] have been proposed for dense 3D reconstruction. Semantic segmentation directly for 3D scenes can satisfy spatial consistency. However, most semantic segmentation frameworks for point cloud [12, 13, 14, 15, 16, 17] are designed for offline use taking a complete reconstructed 3D point cloud as input,



(a) Fused by single-frames (b) Ground-truth

Figure 1: An example showing the instability of single-frame semantic segmentation. (a): fused output of frame-based semantic segmentation results generated by DeepLabV3+ [18] with voting strategy, (b): ground truth semantic segmentation. Semantic labels are indicated by different colors.

and cannot be directly adapted to online semantic segmentation.

In this paper, we introduce LinkNet, a 2D-3D linked multi-modal neural network framework for effective online semantic segmentation that tightly connects the fused 3D geometric information and RGB streams during online 3D reconstruction. The key observation is that, as the projection of the 3D world, although the information sensed in the image space can change due to the conditions of lighting, views, etc., these multi-view images should always be consistent with the same underlying 3D geometry. The main two issues are how to extract an effective feature from the reconstructing 3D scene and how to establish connections among consecutive frames to facilitate a temporally consistent feature representation.

According to the online 3D fusion, we can establish 2D-3D links between 2D images and the fused 3D point cloud to exchange information between the two domains. The benefits of linking 2D and 3D information are twofold. On the one hand, it allows to download the geometric features on the 3D point cloud and map them to the image domain, such that the multi-modal convolutional neural network (CNN) can be applied to improve the performance of image semantic segmentation. On the other hand, the point cloud reconstruction process will be accompanied by a large number of voxel fusion, allowing image domain information corresponding to the same 3D location to be effectively aggregated, which can provide features from different views to strengthen temporal consistency of the semantic segmentation.

More specifically, we convert the segmentation problem of multi-frame images into a multi-voxel classification problem, where each voxel receives continuous observations (i.e., features) from the live RGB-D streams. We thus exploit a recurrent neural network (RNN) to dynamically process such sequential information. We maintain the hidden semantic state of each voxel in the point cloud, and continue to download and upload with the support of 2D-3D links. RNN has certain memory ability, and can make the semantic segmentation results more stable and accurate. For 3D information input in LinkNet, we designed DHAC geometry descriptors, including distance from wall, height from ground, angle between normal and gravity, and curvature. These definitions all have semantic relevance or context relevance. The reason why we did not directly adopt the 3D coordinates as input is that the coordinate values are highly related to the starting position, and it is difficult to apply normalization in online system.

It is worth mentioning that LinkNet refines the semantic segmentation results through 3D reconstruction. At the same time, there are some works [19, 20, 21] that target at improving the quality of scene reconstruction with the help of semantics. These works can also output online semantic segmentation, but they essentially perform the semantic segmentation in the image domain, and do not take 3D information into account. The main contributions of this paper are as follows:

- We propose an online multi-modal semantic segmentation network, named LinkNet, for RGB-D streams, which combines the appearance information of the 2D image domain and the geometric descriptors extracted from the partially reconstructed 3D point cloud.

- We design a lightweight geometric feature, called DHAC (distance, height, angle and curvature), which is invariant to lighting and views, and can be calculated in real-time. This feature is demonstrated to be effective in our online semantic segmentation, and can also be useful for other applications.

- We establish a mechanism for pixel-level / voxel-level 2D-3D links that provides multi-view sequential features for voxels. We demonstrate its usefulness when feeding them to an RNN for stable and accurate online semantic segmentation.

2. Related Work

2.1. Image Semantic Segmentation

Semantic segmentation of images based on deep neural networks has made significant achievements. The iconic end-to-end work is the Fully Convolutional Network (FCN) proposed by Long et al. [3]. The design of FCN uses a well-known encoder-decoder architecture, which is also the basic architecture of most current image segmentation networks. Noh et al. [22] optimized semantic segmentation by designing a deconvolutional neural network. Oliveira et al. [23] applied the fully convolutional neural network to the field of human body part detection and achieved significant results. Following these, U-Net [24], SegNet [25], PSPNet [26] and the DeepLab series [4, 27, 6, 18] have continuously enriched the design of fully convolutional neural networks for image semantic segmentation.

Among them, ERFNet [28], AdapNet++ [29] and DeeplabV3+ [18] are the most advanced network frameworks. In addition to the image pyramid network mentioned above, HRNet [30] maintains high resolution representation during feature learning. The above methods only use the image color information that is easily affected by environment. Recently, Kundu et al. [31] proposed virtual MVFusion that has made progress in 2D image segmentation through smarter view selection and virtual rendering of reconstructed point clouds. However, this method is only suitable for *offline* environment and requires complete scene information. In this paper, we perform *online* multi-modal learning with extra geometric features to break through the limitations of color domain.

2.2. Multi-modal Network with Depth

Depth input is more resistant to interference caused by environment changes, which is an important feature in the study of semantic segmentation. With the increasing popularity of range sensors, some multi-modal networks have been proposed to improve semantic segmentation. Early works such as Couprie et al.'s [8] and Long et al.'s [3] directly treated the depth value as a new information channel and aligned with the color information for synchronous training, but the improvements were limited. Most of the recent works [7, 32, 33, 34] instead used multiple independent encoders for RGB and depth input to learn multi-modal features. Hazirbas et al. [35] designed FuseNet and Jiang et al. [36] proposed RedNet to integrate the features of the depth encoder into the color encoder from bottom up to achieve multi-modal training. Park et al. [37] designed RDFnet with top-down multi-level feature fusion through multi-scale and multi-modal feature blocks. Xiang and Fox [38] proposed DA-RNN that makes frame association through depth and Kinect-Fusion [9]. The SSMA framework designed by Valada et al. [29] is an adaptive method based on self-supervision. In this paper, we propose a better geometric feature descriptor, i.e., DHAC, which is generated from the point cloud and invariant to lighting and views. Moreover, our multi-modal fusion can take advantage of different modalities.

2.3. Deep learning on 3D point cloud

3D point cloud learning is a research hotspot in recent years. As the pioneer of point cloud learning, PointNet [12] uses global feature aggregation to realize point-wise point cloud feature learning. Then PointNet++ [39] uses spatial neighborhood information to enhance local features. DGCNN [40] uses the embedding feature domain to construct a dynamic graph, and proposes EdgeConv to implement an order-independent convolution. There are also many work to define the convolution operation for point clouds. PCNN [41] performs 3D convolution by constructing a local voxel domain. Cai et al. [42] used local depth mapping to project the point cloud onto the tangent plane to perform 2D convolution. PointCNN [13] specifies the input order of point cloud subsets by learning the arrangement matrix and uses 1D convolution for feature extraction. In addition, MCCNN [43] and PointConv [14] use Monte Carlo estimation to simulate the convolution operation. Recently, the Transformer [44], which is widely popular in the field of natural language learning, has begun to be extended to point cloud learning, thanks to the input order independence of the self-attention mechanism. PCT [45] is a classic migration work of Transformer. It directly applies the attention mechanism to global feature learning, and uses neighborhood embedding and Laplacian matrix-based offset-attention to optimize the performance. PointASNL [46] uses the attention mechanism to extract local features. PointGMM [47] proposes MLP splits and attentional splits to achieve shape completion. The above methods are all run in an offline manner, and special segmentation and resampling are required for large-scale 3D scenes. More comprehensive surveys on this topic can be found in [48, 49].

2.4. Online Semantic Segmentation

RGB-D videos have similar regular structure as ordinary videos. However, there is not much research on video-oriented deep neural networks for semantic segmentation, because multi-frame input will cause a burden to the design of the network. Zhang et al. [50] stacked the video frame data, then divided it into supervoxels, and finally trained to process the video with a 3D convolutional neural network in units of voxels. Shelhamer et al. [51] proposed the Clockwork network. This work assumes that the changes in the pixel domain caused by time changes are drastic, while the semantic changes are slight. Luc et al. [52] proposed the SegmPred model to predict the semantics of the upcoming frame through an adversarial network. These methods are based on the adaptation of improvement on 2D images, and no 3D geometric information is considered.

Another common way is 3D semantic reconstruction. SemanticFusion designed by McCormac et al. [20] uses semantic information as an aid to achieve more accurate scene reconstruction. Rünz et al. [21] proposed MaskFusion, in which instance segmentation results were used to track and reconstruct moving objects. Yang et al. [19] also used the semantic distribution of pixels to optimize the pose estimation. Zhang et al. [53] combined SSMA [29] on images and PointConv [14] on point clouds to optimize the voxel-wise semantic labeling. These methods can output scene semantic information online, but the semantic segmentation results are generated by related networks designed for the RGB image and the voxel in the reconstruction process. Their semantic segmentation results thus do not fully consider the 3D geometric and multi-view information. Our work aims to optimize semantic segmentation using 3D reconstruction.

3. Method

Fig. 2 shows the pipeline of our 2D-3D LinkNet. LinkNet takes live RGB-D video frames and camera poses as input, and outputs pixel-wise semantic predictions and semantic segmentation results of 3D point clouds online. First, we use point cloud fusion to establish the 2D-3D links between the 2D image and the 3D point cloud. Secondly, the geometric features generated from the 3D point cloud are downloaded to each frame, which are then used to output the semantic features via multi-modal learning. Finally, we refine the semantic features and achieve stable semantic segmentation predictions through a RNN module with the help of 2D-3D links.

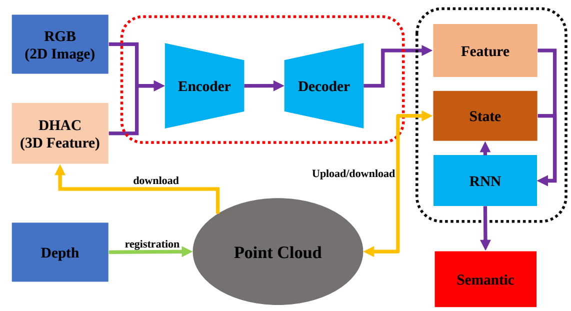


Figure 2: Pipeline of LinkNet. The red dashed box represents the multi-modal CNN, which takes 2D channels (RGB) and 3D channels (DHAC) as input and generates semantic features. The black dashed box represents an RNN module, which downloads/uploads hidden states through 2D-3D links between 2D pixels of RGB-D images and 3D voxels of the reconstructed point cloud.

3.1. Mapping between the RGB-D Image and Point Cloud

Before going deeper into the point cloud fusion, we briefly introduce the transformation between the image coordinates and camera coordinates. Given an aligned RGB-D image with the color channels C and depth channel $\mathcal{D}$ defined in domain $\mathcal{I} \subset \mathbb{R}^2$. Suppose the camera intrinsic matrix is $\mathbf{K} \in \mathbb{R}^{3\times 3}$, we can transform a pixel i: $\mathcal{I}(i) = (u_i, v_i)$ in the image space into a 3D point $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ in the camera space using homogeneous coordinates as follows:

$$\begin{aligned} p_i^T &= f_{\mathbf{K}}(i) \cdot (u_i, v_i, 1)^T, \\ f_{\mathbf{K}}(i) &= \mathcal{D}(i) \cdot \mathbf{K}^{-1}. \end{aligned} \tag{1}$$

Fig. 3 (a-b) show an example of converting an RGB-D image into a 3D point cloud.

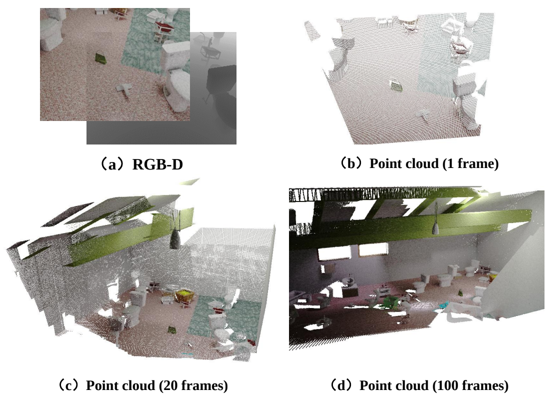


Figure 3: Point cloud fusion of depth images using camera poses. The scale of the scene and the density of the point cloud will increase as the number of registered frames increases.

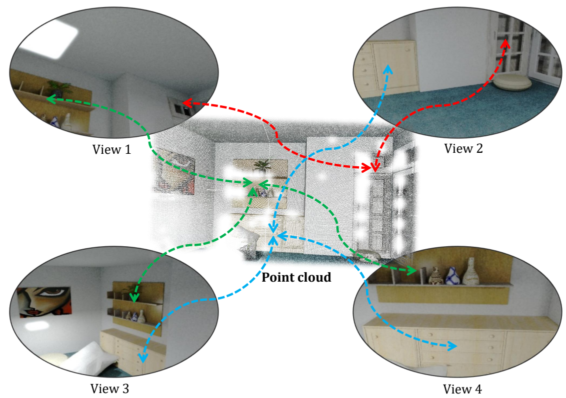


Figure 4: Example of 2D-3D Links. The colors of dotted arrows represent different categories of objects.

3.2. Point Cloud Fusion

By processing multi-frame data $\{\mathcal{I}^t\}$, where t is the frame (time) index, we can obtain the voxel set $\{\mathcal{V}^t\}$ corresponding to each RGB-D frame. However, the coordinate system of each frame is independent to each other. Here we need to use point cloud registration to estimate the relative pose between frames and fuse voxels from different views.

Assuming that the global camera pose of the frame data at time t is $\mathbf{T}^t \in \mathbb{SE}^3$, the converted point cloud data is $\mathcal{V}^t$. The specific relationship is as follows:

$$\begin{aligned} \mathcal{V}^t &= \{V_i = (x_i, y_i, z_i, t, f_i, s_i, l_i), i \in \mathcal{I}^t\}, \\ (x_i, y_i, z_i, 1)^T &= \mathbf{T}^t \cdot (p_i, 1)^T, \end{aligned} \tag{2}$$

where V_i represents the stored information for the voxel corresponding to the pixel i, (x_i, y_i, z_i) is the position of the voxel in the global space, t is the latest timestamp of the voxel, p_i is the 3D position in the camera space corresponding to pixel i, f_i is a geometric feature descriptors that will be introduced in Sec. 3.3, and s_i refers to the hidden semantic state stored on the point cloud to memorize the point cloud semantic label l_i at the voxel. There is no need to store colors in voxels, because each frame has its own color information, which will change due to different camera views or lighting conditions. Besides, the voxel already contains more reliable semantic information in s_i. It is worth noting that the camera pose can be solved by various SLAM or 3D reconstruction methods (as a byproduct of these algorithms), which is not the main focus of this paper. In most cases, we directly use the pose information provided by the 3D benchmark.

Assuming that the registered point cloud set before t is $\mathcal{S}^{t-1}$, the current frame point cloud is $\mathcal{V}^t$. We need to design fusion rules $\mathcal{S}^t = fuse(\mathcal{S}^{t-1}, \mathcal{V}^t)$ to produce the fused point cloud. Specifically, voxels V_a and V_b are to be fused into a single voxel V_c if the following conditions are satisfied:

$$\begin{aligned} V_a &\in \mathcal{S}^{t-1} \\ V_b &\in \mathcal{V}^t \\ Grid(x_a, y_a, z_a) &= Grid(x_b, y_b, z_b) \\ Grid(x, y, z) &= (\lfloor \frac{x}{\epsilon} \rfloor, \lfloor \frac{y}{\epsilon} \rfloor, \lfloor \frac{z}{\epsilon} \rfloor) \end{aligned} \tag{3}$$

where ϵ is the size of the voxel unit, and it is set to $\epsilon = 2cm$ in this work. We update the fused voxel V_c as follows:

$$\begin{aligned} V_c &= fuse(V_a, V_b) = (x_b, y_b, z_b, t_c, f_c, s_a, l_a) \\ (t_c, f_c) &= \begin{cases} (t_a, f_a), & (t_b - t_a) < 1sec. \\ (t_b, f_b), & \text{otherwise} \end{cases}. \end{aligned} \tag{4}$$

As above, during the voxel fusion process, we limit the update frequency of feature generation to improve efficiency (i.e., only recalculating geometric features when the time elapsed is over 1 second). Fig. 3 shows an example of the point cloud fusion. Obviously, the more frames we fuse, the more reliable and accurate geometric shape information and richer context are to be obtained.

Through point cloud fusion, we can obtain a series of 2D-3D links. These links specify a unique corresponding 3D voxel for each pixel. As shown in Fig. 4, we can establish the association among pixels of multi-views through the point cloud, and provide sequential data input for semantic prediction of voxels.

3.3. DHAC Geometric Descriptor

Color information is easily affected by the environment, such as lighting, weather or view-point, which induces instability for semantic segmentation. Besides, existing work [7] shows that encoding depth information through HHA features can improve performance. We thus propose DHAC, a 3D geometric descriptor satisfying spatial consistency. As an upgraded version of HHA, DHAC is more capable of describing scenes. Given a point $p_i = (x_i, y_i, z_i)$ in a point cloud $\mathcal{P}$, its DHAC descriptor f_i is calculated as:

$$
\begin{aligned}
f_i &= (d_i, h_i, a_i, c_i) \\
d_i &= \min\{\|p_i - p_j\|, p_j \in BB(\mathcal{P})\} \\
h_i &= z_i \cdot \vec{g} \\
a_i &= \| \arccos(\vec{n_i} \cdot \vec{g}) \|
\end{aligned} \tag{5}
$$

where d_i refers to the distance between p_i and walls, computed as the shortest distance between p_i and the bounding box (BB) of the 3D point cloud, h_i is the height along the direction of gravity $\vec{g}$, a_i is the angle between the normal $\vec{n_i}$ and gravity $\vec{g}$, and c_i is the curvature.

Normal $\vec{n_i}$ and curvature c_i can be estimated by the Principal Component Analysis (PCA) algorithm. Note that PCA normal estimation requires neighborhoods of a certain size that can be retrieved by a KD-tree. However, the KD-tree data structure is hard to build online, and its K-Nearest Neighbor (KNN) search algorithm is also time-consuming. Instead of maintaining a global KD-tree, we dynamically maintain the KNN for each voxel during the 3D reconstruction process, which is initialized and updated according to the 2D neighbors of the corresponding pixel. Specifically, we choose the 5×5 neighbors around each pixel as the candidates for voxel KNN. In this work, all the K value of KNN is set to 16.

Strictly speaking, in the start-up phase, d_i and h_i will gradually change with the update of the scene, so they do

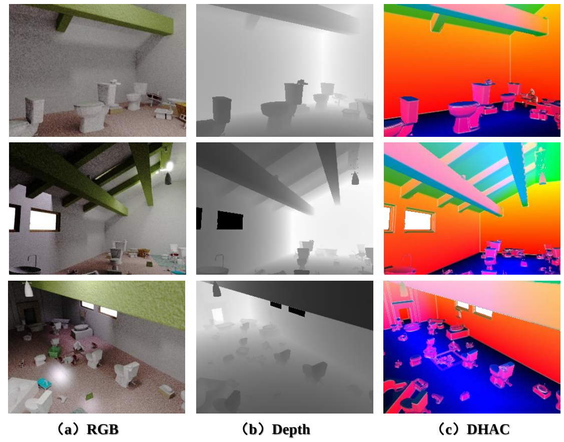


Figure 5: Examples of DHAC images. (a) (b) are the raw color and depth images. (c) DHAC images (distance, height, angle and curvature are mapped to RGBA channels).

not hold the view invariance completely. Nevertheless, they still have very good consistency. In the multi-modal learning process, we map f_i back into the 2D image domain to generate the DHAC images. As shown in Fig. 5, the DHAC descriptors can characterize the geometric features well and are almost consistent among different viewpoints. All these descriptors are highly semantic related or context related. Therefore, DHAC can effectively improve network performance.

3.4. LinkNet

The detailed architecture design of our LinkNet is shown in Fig. 6. Our LinkNet consists of two main modules: a multi-modal network and an RNN module.

The multi-modal network is intended to generate the multi-modal feature for the input color and depth data, which is developed from FuseNet [35]. Although any suitable multi-mode network can be used as the backbone of LinkNet, we adopt the FuseNet here by considering the trade-off between the performance and the efficiency. We extend the input channel of its depth encoder to support multi-modal learning of RGB and DHAC images via ‘RGB Encoder’ and ‘DHAC Encoder’, respectively. The 5-layer convolution design of the encoders is referenced from VGG16 [54]. Each output of ‘DHAC Encoder layer’ will be added to the output of the corresponding layer of ‘RGB Encoder’ to achieve multi-modal feature

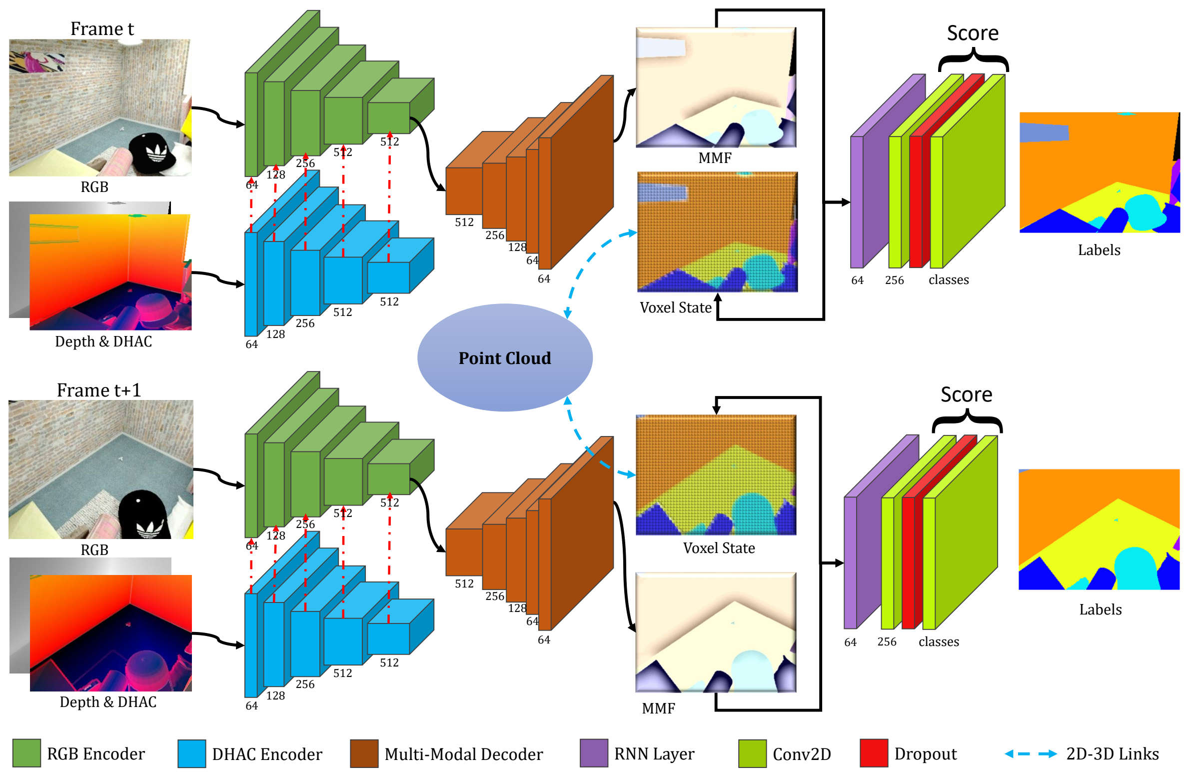


Figure 6: The architecture of LinkNet. The input RGB-D streams together with the proposed DHAC feature are fed into the RGB Encoder and DHAC Encoder, followed by a multi-modal decoder to generate the multi-modal feature. Before being sent to a Score layer for a temporally consistent semantic prediction, this multi-modal feature is refined by an RNN module with the help of the "voxel state" of the 3D point cloud that can be downloaded and uploaded via 2D-3D links (blue dotted arrows).

fusion (as illustrated by the red dotted arrow in Fig. 6). The final multi-modal feature Fm is decoded through a 5-layer 'Multi-Modal Decoder'. For more detailed network framework, please refer to [35].

Another core module of LinkNet is a 2D-3D linked RNN module. This module is designed to learn a temporally consistent feature representation for stable semantic prediction through the 2D-3D link between 2D images and the underlying 3D geometry. Specifically, for each pixel i of frame $\mathcal{I}^t$, we first find its linked voxel V_j using the method introduced in Sec. 3.2. We then feed the output feature of that pixel, Fm_i^t, from the previous multi-modal feature network and the voxel state s_j^{t-1} (including the hidden state and cell state), which is stored in the corresponding 3D voxel, into an RNN. The RNN generates the output feature o_i^t for pixel i and updates the voxel state as follows:

$$(o_i^t, s_j^t) = RNN(Fm_i^t, s_j^{t-1}). \tag{6}$$

If there is no pixels in frame t linked to voxel V_x, then s_x^t will be equal to s_x^{t-1}. Our RNN module is formed by two stacked standard Long Short-Term Memory(LSTM) modules [55] with the dimension of their hidden state and cell state set to 64. Their initial value is set to 0 and updated over time through valid 2D-3D links. The output feature from the RNN is further fed into a **Score** layer to predict the semantic label l_i^t online:

$$Labels = \{l_i^t\} = argmax\{\mathbf{Score}(\{o_i^t\})\} \quad (7)$$

This Score layer is composed of two convolution layers sandwiching a dropout layer. The kernel sizes of convolution layers are set as $[3 \times 3]$ and the probability of dropout is 0.2. Please note that the convolution layer here is not equivalent to the fully connected layer, because its kernel size is not $[1 \times 1]$.

4. Experiments and Results

Implementation Details. We trained the backbone network (composed of the RGB encoder, DHAC encoder and the Multi-Modal decoder), and the RNN module (i.e, the two stacked LSTMs and the Score layer), separately. Cross-entropy loss function is adopted during the training of both backbone network and the RNN. The initial learning rates of the backbone network and RNN module training are set to $2e-3$ and $5e-5$, respectively. They will decrease by 10% for every 500,000 iterations. The training batch size of the backbone network is set to 12, and of course, the batch size of RNN module is 1. For all input data, we resize it to a resolution of 320×240 pixels. This is because it is the resolution of depth maps for most range sensors, and a low resolution input can also speed up the inference. The number of epochs for training will be introduced later.

We evaluate LinkNet through both a synthetic dataset, i.e., SceneNet RGB-D [1], and a real scan dataset, i.e, ScanNet v2 [2]. Although our work can predict voxel-wise semantic labels, the quality of 3D reconstructed point cloud will be affected by the selected fusion algorithm. Therefore, we mainly evaluate the semantic segmentation of 2D images.

4.1. Timings

All experiments are performed on a computer with an Intel i7-8700K CPU, 64GB RAM and an Nvidia GeForce GTX 1080 Ti GPU (11GB on-board memory). Code written with Jittor [56] implementation will be available at: `https://github.com/archershot/linkNet`.

In the case of a single GPU, the average runtime per frame of our work is about 56ms (i.e., 18FPS), of which the LinkNet inference time is about 45ms per frame and the DHAC descriptor computation (including 2D-3D link generation) is about 11ms per frame. The system efficiency can be further increased to 23FPS using multi-GPU with streaming optimization. This efficiency is at the same level as most online 3D reconstruction algorithms and meets the requirements of online applications.

4.2. Results on the SceneNet RGB-D dataset

SceneNet RGB-D [1] is a synthetic dataset containing 16,865 indoor scans, and each scan contains 300 annotated RGB-D frames that are selected every 25 frames. The layout, texture and lighting of the objects in this dataset are all randomly generated. SceneNet RGB-D contains 258 instance labels that are divided into 14 semantic categories according to the NYU Depth V2 [57] standard.The experiment follows standard training/validation split reported in [1]. The number of training epoch for the backbone network is set to 20 with about 1×10^8 iterations and the one for the RNN module is set to 1 with about 5×10^6 iterations.

To demonstrate the advantages of our linked multi-modal network, we conduct extensive ablation studies: without the RNN module, and using single or combined modalities as inputs. Fig. 7 shows examples of single-modal semantic segmentation results. Among these modalities, HHA is a feature coding method based on depth and gravity estimation proposed by Gupta et al. [7]. This modality is more friendly to semantic segmentation than depth. It can be seen that the DHAC feature, benefiting from its good geometric properties, can resist the interference of lighting, texture and view-point, making it a suitable presentation for semantic segmentation in challenging conditions. It contains richer information than other modalities, leading to better performance. Fig. 8 shows examples of multi-modal experiments. It can be found that multi-modal input can be complementary to each other in the semantic segmentation. Especially in a dark lighting condition, modalities other than color are essential for prediction, and the DHAC feature clearly shows the best effect.

Table 1 lists the class-wise semantic segmentation results of different modal combinations. The results are evaluated with *OA*, *mAcc* and *mIoU* metrics. *OA* is the overall accuracy, *mAcc* is class-wise averaged recall, and *mIoU* is class-wise averaged *IoU*, which is defined as the

Table 1: Detailed comparison of various input modalities on the SceneNet RGB-D dataset [1].

Methods	Beds	Books	Ceiling	Chair	Floor	Furniture	Objects	Picture
RGB	22.0	-	77.8	29.6	77.2	36.0	35.8	69.4
Depth	53.7	-	72.8	40.2	67.9	24.4	54.6	24.6
HHA	47.1	-	67.8	35.2	66.6	14.3	55.9	17.5
DHAC	56.9	-	75.0	46.9	70.9	33.8	60.6	26.8
RGB+Depth (FuseNet)	46.2	-	79.3	53.7	75.1	36.9	54.5	51.0
RGB+Depth (SSMA)	19.3	-	74.5	21.5	69.3	17.1	35.5	29.4
RGB+HHA (FuseNet)	47.4	-	82.9	38.1	78.5	41.4	47.6	49.5
RGB+DHAC (FuseNet)	53.9	-	83.1	49.1	**84.8**	52.1	55.9	55.5
RGB+Depth (**LinkNet**)	51.3	-	83.3	50.6	82.2	38.0	56.2	51.2
RGB+DHAC (**LinkNet**)	**60.9**	-	**83.4**	**63.2**	83.2	**59.2**	**68.0**	**66.8**

Methods	Sofa	Table	TV	Wall	Window	**OA**	**mAcc**	**mIoU**
RGB	08.5	30.2	14.1	78.2	30.8	77.8	60.2	39.2
Depth	06.6	44.7	09.9	69.9	23.1	76.4	56.3	37.9
HHA	18.4	47.0	15.9	64.7	21.6	72.6	56.7	36.3
DHAC	21.0	57.0	25.6	70.2	24.6	78.0	65.3	43.8
RGB+Depth (FuseNet)	22.6	45.6	28.3	80.5	25.7	82.1	63.4	46.1
RGB+Depth (SSMA)	01.2	30.3	02.1	73.6	13.1	75.6	41.5	29.8
RGB+HHA (FuseNet)	18.0	54.3	41.9	81.4	**31.9**	82.5	66.3	47.1
RGB+DHAC (FuseNet)	18.8	58.0	49.1	82.1	29.1	84.4	69.8	51.7
RGB+Depth (**LinkNet**)	12.8	49.0	35.4	83.2	29.9	84.2	64.2	47.9
RGB+DHAC (**LinkNet**)	**29.7**	**66.5**	**61.5**	**83.3**	31.7	**86.6**	**73.3**	**58.3**

ratio of the intersection and union between the prediction and ground-truth. Although the occurrences of books are too low to be reliably classified, in most other categories, our LinkNet achieves a comprehensive improvement, which has a significant improvement of 12% in *mIoU* compared to the base model FuseNet. This shows that both the DHAC feature and our RNN module contribute to the improvement of semantic segmentation.

4.3. Comparisons on the ScanNet v2 dataset

The ScanNet v2 dataset [2] contains 1,513 scans of real indoor scenes with various object categories. The 2D semantic segmentation training/test set (ScanNet25k) provided by the benchmark contains 19,466 images for training, 5436 images for validation and 2,135 images for testing. The training epoch of the backbone network is set to 200 with about 4×10^6 iterations. And the training epoch of the RNN module is set to 10 with about 2×10^5 iterations.

Table 2 shows the semantic segmentation results on the ScanNet v2 test set. All the results of selected 21 classes are drawn from the ScanNet leaderboard [1]. We make comparisons with the representative works including Enet [58], PSPNet [59], MSeg [60], FuseNet [35], AdapNet++ [29] and SSMA [29]. Obviously, multimodal methods have clear advantages, among which our LinkNet performs quite well. Compared with FuseNet, LinkNet improves IoU by 3.1%. The improvement of

[1] http://kaldir.vc.in.tum.de/scannet_benchmark/semantic_label_2d

Table 2: Comparisons of LinkNet with bechmarking results on the ScanNet v2 test set.

Methods	**Mode**	**mIoU**	Bathtub	Bed	Book Shelf	Cabinet	Chair	Counter	Curtain	Desk	Door
Enet	single	37.6	26.4	45.2	45.2	36.5	18.1	14.3	45.6	40.9	34.6
PSPNet	single	47.5	49.0	58.1	28.9	50.7	06.7	37.9	61.0	41.7	43.5
MSeg	single	48.5	50.5	70.9	09.2	42.7	24.1	41.1	65.4	38.5	45.7
AdapNet++	single	50.3	61.3	72.2	41.8	35.8	33.7	37.0	47.9	44.3	36.8
FuseNet	multi	53.5	57.0	68.1	18.2	51.2	29.0	43.1	65.9	50.4	49.5
SSMA	multi	**57.7**	**69.5**	71.6	**43.9**	**56.3**	**31.4**	44.4	**71.9**	**55.1**	50.3
LinkNet	multi	56.6	65.6	**73.4**	18.0	54.4	29.4	**51.5**	67.7	51.4	**53.2**

Methods	Floor	Other Furniture	Picture	Refrig-erator	Shower Curtain	Sink	Sofa	Table	Toilet	Wall	Window
Enet	76.9	16.4	21.8	35.9	12.3	40.3	38.1	31.3	57.1	68.5	47.2
PSPNet	82.2	27.8	26.7	50.3	22.8	61.6	53.3	37.5	82.0	72.9	56.0
MSeg	86.1	05.3	27.9	50.3	48.1	64.5	62.6	36.5	74.8	72.5	52.9
AdapNet++	90.7	20.7	21.3	46.4	52.5	61.8	65.7	45.0	78.8	72.1	40.8
FuseNet	90.3	30.8	42.8	52.3	36.5	67.6	62.1	47.0	76.2	77.9	54.1
SSMA	88.7	**34.6**	34.8	**60.3**	**35.3**	70.9	60.0	45.7	**90.1**	78.6	**59.9**
LinkNet	**91.6**	33.0	**47.2**	56.3	32.0	**71.3**	**62.8**	**47.6**	84.4	**80.4**	59.8

LinkNet on ScanNet v2 is relatively limited. This is mainly because the ScanNet v2 test set just selects 1 frame every 100 frames. This reduces the number of available 2D-3D links, making it difficult to take full advantage of the RNN module of our LinkNet. At present, LinkNet outperforms SSMA [29] in about half of the categories, but the *mIoU* is slightly lower than that of SSMA, mainly because of the gap in the backbone network (i.e., FuseNet vs. SSMA, especially for the category of *book-shelf*). Although we can further improve the performance by choosing SSMA as the backbone network of LinkNet, it is difficult to meet the requirement of online 3D reconstruction, since the running time of each frame of SSMA is about 100*ms*.

4.4. Stability Analysis

To quantitatively evaluate how our LinkNet improves the temporal consistency of semantic segmentation for online streams, we compute the average semantic change ratio of pixels projected from the underlying 3D voxels among all consecutive frames on the SceneNet RGB-D validation set. We regard this metric as the *stability* of the online semantic segmentation: the lower the ratio is, the more stable the semantic segmentation is.

Table 3: Stability comparison on SceneNet RGB-D validation set.

Method	Stability
RGB+Depth (FuseNet)	8.73%
RGB+DHAC	7.12%
LinkNet	3.89%

We compare our LinkNet with FuseNet [35] as well as FuseNet with DHAC feature. As shown in Table 3, 8.73% of pixel labels are changed with FuseNet, while our LinkNet achieves more consistent semantic segmentation result with only 3.89% of label changes. In addition, DHAC also contributes to stable segmentation due to its insensitivity to the change of views.

4.5. Limitation

Our method also has some limitations. First, the feature refinement of LinkNet is preformed at the pixel level or voxel level, instead of the instance level. This may corrupt the semantic labeling results of the same instance, resulting in discontinuity in semantic segmentation. A progressive clustering [61] on voxels can be applied to alleviate this problem. Second, the RNN module would accumulate errors when a voxel is frequently linked to pixels with noise feature representation. A view selection strategy [31] would help to improve the quality of input frames.

5. Conclusion

In this paper, we propose LinkNet to perform stable and effective online semantic segmentation of RGB-D video. On the one hand, LinkNet incorporates the geometric features extracted from the fused 3D geometry into multi-modal learning in the image domain to improve feature robustness by taking advantage of the 2D-3D links offered by 3D reconstruction. On the other hand, LinkNet applies an RNN on the sequential features observed by each voxel to maintain the stability of semantic segmentation. Experiments on both synthetic and real scanned datasets demonstrate the effectiveness of our method.

In the future, we would like to consider more complex 3D features that are more suitable for semantic segmentation, such as voxel-based deep learning features. In addition, the backbone network can also be upgraded for 2D-3D multi-modal application.

References

[1] McCormac, J, Handa, A, Leutenegger, S, Davison, AJ. SceneNet RGB-D: can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation? In: IEEE International Conference on Computer Vision. IEEE Computer Society; 2017, p. 2697–2706.

[2] Dai, A, Chang, AX, Savva, M, Halber, M, Funkhouser, T, Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 2432–2443.

[3] Long, J, Shelhamer, E, Darrell, T. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2015, p. 3431–3440.

[4] Chen, L, Papandreou, G, Kokkinos, I, Murphy, K, Yuille, AL. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: Bengio, Y, LeCun, Y, editors. International Conference on Learning Representations. 2015,.

[5] Yu, F, Koltun, V. Multi-scale context aggregation by dilated convolutions. In: Bengio, Y, LeCun, Y, editors. International Conference on Learning Representations. 2016,.

[6] Chen, L, Papandreou, G, Schroff, F, Adam, H. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:170605587 2017;.

[7] Gupta, S, Girshick, RB, Arbeláez, PA, Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In: Fleet, DJ, Pajdla, T, Schiele, B, Tuytelaars, T, editors. European Conference on Computer Vision; vol. 8695 of *Lecture Notes in Computer Science*. Springer; 2014, p. 345–360.

[8] Couprie, C, Farabet, C, Najman, L, LeCun, Y. Indoor semantic segmentation using depth information. In: Bengio, Y, LeCun, Y, editors. International Conference on Learning Representations. 2013,.

[9] Izadi, S, Kim, D, Hilliges, O, Molyneaux, D, Newcombe, RA, Kohli, P, et al. KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Pierce, JS, Agrawala, M, Klemmer, SR, editors. ACM Symposium on User Interface Software and Technology. ACM; 2011, p. 559–568.

[10] Whelan, T, Salas-Moreno, RF, Glocker, B, Davison, AJ, Leutenegger, S. ElasticFusion: Real-time dense SLAM and light source estimation. International Journal of Robotics Research 2016;35(14):1697–1716.

[11] Dai, A, Nießner, M, Zollhöfer, M, Izadi, S, Theobalt, C. BundleFusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics 2017;36(3):24:1–24:18.

[12] Qi, CR, Su, H, Mo, K, Guibas, LJ. PointNet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2017, p. 77–85.

[13] Li, Y, Bu, R, Sun, M, Wu, W, Di, X, Chen, B. PointCNN: Convolution on X-Transformed points. In: Advances in Neural Information Processing Systems. 2018, p. 828–838.

[14] Wu, W, Qi, Z, Li, F. PointConv: Deep convolutional networks on 3d point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, p. 9621–9630.

[15] Hu, S, Cai, J, Lai, Y. Semantic labeling and instance segmentation of 3d point clouds using patch context analysis and multiscale processing. IEEE Transactions on Visualization and Computer Graphics 2020;26(7):2485–2498.

[16] Lu, Y, Zhen, M, Fang, T. Multi-view based neural network for semantic segmentation on 3d scenes. Sci China Inf Sci 2019;62(12):229101.

[17] Peng, H, Zhou, B, Yin, L, Guo, K, Zhao, Q. Semantic part segmentation of single-view point cloud. Sci China 2020;63(12):224101.

[18] Chen, L, Zhu, Y, Papandreou, G, Schroff, F, Adam, H. Encoder-Decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V, Hebert, M, Sminchisescu, C, Weiss, Y, editors. European Conference on Computer Vision; vol. 11211 of *Lecture Notes in Computer Science*. Springer; 2018, p. 833–851.

[19] Yang, S, Kuang, Z, Cao, Y, Lai, Y, Hu, S. Probabilistic projective association and semantic guided relocalization for dense reconstruction. In: IEEE International Conference on Robotics and Automation. IEEE; 2019, p. 7130–7136.

[20] McCormac, J, Handa, A, Davison, AJ, Leutenegger, S. SemanticFusion: Dense 3d semantic mapping with convolutional neural networks. In: IEEE International Conference on Robotics and Automation. IEEE; 2017, p. 4628–4635.

[21] Rünz, M, Agapito, L. MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects. IEEE International Symposium on Mixed and Augmented Reality 2018;:10–20.

[22] Noh, H, Hong, S, Han, B. Learning deconvolution network for semantic segmentation. In: IEEE International Conference on Computer Vision. IEEE Computer Society; 2015, p. 1520–1528.

[23] Oliveira, GL, Valada, A, Bollen, C, Burgard, W, Brox, T. Deep learning for human part discovery in images. In: Kragic, D, Bicchi, A, Luca, AD, editors. IEEE International Conference on Robotics and Automation. IEEE; 2016, p. 1634–1641.

[24] Ronneberger, O, Fischer, P, Brox, T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer Assisted Intervention. 2015, p. 234–241.

[25] Badrinarayanan, V, Kendall, A, Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017;39(12):2481–2495.

[26] Zhao, H, Shi, J, Qi, X, Wang, X, Jia, J. Pyramid scene parsing network. In: IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 6230–6239.

[27] Chen, L, Papandreou, G, Kokkinos, I, Murphy, K, Yuille, AL. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018;40(4):834–848.

[28] Romera, E, Alvarez, JM, Bergasa, LM, Arroyo, R. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. IEEE

Transactions on Intelligent Transportation Systems 2018;19(1):263–272.

[29] Valada, A, Mohan, R, Burgard, W. Self-supervised model adaptation for multimodal semantic segmentation. International Journal of Computer Vision 2020;128(5):1239–1285.

[30] Wang, J, Sun, K, Cheng, T, Jiang, B, Deng, C, Zhao, Y, et al. Deep high-resolution representation learning for visual recognition. arXiv preprint arXiv:190807919 2019;.

[31] Kundu, A, Yin, X, Fathi, A, Ross, D, Brewington, B, Funkhouser, T, et al. Virtual multi-view fusion for 3d semantic segmentation. In: European Conference on Computer Vision. 2020,.

[32] Cheng, Y, Cai, R, Li, Z, Zhao, X, Huang, K. Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2017, p. 1475–1483.

[33] Wang, J, Wang, Z, Tao, D, See, S, Wang, G. Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. In: Leibe, B, Matas, J, Sebe, N, Welling, M, editors. European Conference on Computer Vision; vol. 9909 of *Lecture Notes in Computer Science*. Springer; 2016, p. 664–679.

[34] Song, X, Herranz, L, Jiang, S. Depth cnns for RGB-D scene recognition: Learning from scratch better than transferring from rgb-cnns. In: Singh, SP, Markovitch, S, editors. AAAI Conference on Artificial Intelligence. AAAI Press; 2017, p. 4271–4277.

[35] Hazirbas, C, Ma, L, Domokos, C, Cremers, D. FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In: Lai, S, Lepetit, V, Nishino, K, Sato, Y, editors. Asian Conference on Computer Vision; vol. 10111 of *Lecture Notes in Computer Science*. Springer; 2016, p. 213–228.

[36] Jiang, J, Zheng, L, Luo, F, Zhang, Z. RedNet: Residual encoder-decoder network for indoor RGB-D semantic segmentation. arXiv preprint arXiv:180601054 2018;.

[37] Park, SJ, Hong, KS, Lee, S. RDFNet: RGB-D multi-level residual feature fusion for indoor semantic segmentation. In: IEEE International Conference on Computer Vision. 2017, p. 4980–4989.

[38] Xiang, Y, Fox, D. DA-RNN: semantic mapping with data associated recurrent neural networks. In: Amato, NM, Srinivasa, SS, Ayanian, N, Kuindersma, S, editors. Robotics: Science and Systems. 2017,.

[39] Qi, CR, Yi, L, Su, H, Guibas, LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. 2017, p. 5099–5108.

[40] Wang, Y, Sun, Y, Liu, Z, Sarma, SE, Bronstein, MM, Solomon, JM. Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics 2019;38(5):146:1–146:12.

[41] Atzmon, M, Maron, H, Lipman, Y. Point convolutional neural networks by extension operators. ACM Transactions on Graphics 2018;37(4):71:1–71:12.

[42] Cai, J, Mu, T, Lai, Y, Hu, S. Deep point-based scene labeling with depth mapping and geometric patch feature encoding. Graph Model 2019;104.

[43] Hermosilla, P, Ritschel, T, Vázquez, P, Vinacua, À, Ropinski, T. Monte carlo convolution for learning on non-uniformly sampled point clouds. ACM Transactions on Graphics 2018;37(6):235:1–235:12.

[44] Vaswani, A, Shazeer, N, Parmar, N, Uszkoreit, J, Jones, L, Gomez, AN, et al. Attention is all you need. In: Advances in Neural Information Processing Systems. 2017, p. 5998–6008.

[45] Guo, MH, Cai, JX, Liu, ZN, Mu, TJ, Martin, RR, Hu, SM. Pct: Point cloud transformer. Comput Vis Media 2021;.

[46] Yan, X, Zheng, C, Li, Z, Wang, S, Cui, S. PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE; 2020, p. 5588–5597.

[47] Hertz, A, Hanocka, R, Giryes, R, Cohen-Or, D. PointGMM: A neural GMM network for point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE; 2020, p. 12051–12060.

[48] Bronstein, MM, Bruna, J, LeCun, Y, Szlam, A, Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. IEEE Signal Process Mag 2017;34(4):18–42.

[49] Xiao, Y, Lai, Y, Zhang, F, Li, C, Gao, L. A survey on deep geometry learning: From a representation perspective. Comput Vis Media 2020;6(2):113–133.

[50] Zhang, H, Jiang, K, Zhang, Y, Li, Q, Xia, C, Chen, X. Discriminative feature learning for video semantic segmentation. In: International Conference on Virtual Reality and Visualization. 2014, p. 321–326.

[51] Shelhamer, E, Rakelly, K, Hoffman, J, Darrell, T. Clockwork convnets for video semantic segmentation. In: Hua, G, Jégou, H, editors. European Conference on Computer Vision; vol. 9915 of *Lecture Notes in Computer Science*. 2016, p. 852–868.

[52] Luc, P, Neverova, N, Couprie, C, Verbeek, J, LeCun, Y. Predicting deeper into the future of semantic segmentation. In: IEEE International Conference on Computer Vision. IEEE Computer Society; 2017, p. 648–657.

[53] Zhang, J, Zhu, C, Zheng, L, Xu, K. Fusion-aware point convolution for online semantic 3d scene segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2020, p. 4533–4542.

[54] Simonyan, K, Zisserman, A. Very deep convolutional networks for large-scale image recognition. In: Bengio, Y, LeCun, Y, editors. International Conference on Learning Representations. 2015,.

[55] Hochreiter, S, Schmidhuber, J. Long short-term memory. Neural Computation 1997;9(8):1735–1780.

[56] Hu, SM, Liang, D, Yang, GY, Yang, GW, Zhou, WY. Jittor: a novel deep learning framework with meta-operators and unified graph execution. Science China Information Science 2020;63(222103):1–21.

[57] Silberman, N, Hoiem, D, Kohli, P, Fergus, R. Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, AW, Lazebnik, S, Perona, P, Sato, Y, Schmid, C, editors. European Conference on Computer Vision; vol. 7576 of *Lecture Notes in Computer Science*. Springer; 2012, p. 746–760.

[58] Paszke, A, Chaurasia, A, Kim, S, Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:160602147 2016;.

[59] Zhao, H, Shi, J, Qi, X, Wang, X, Jia, J. Pyramid scene parsing network. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2017, p. 6230–6239.

[60] Lambert, J, Liu, Z, Sener, O, Hays, J, Koltun, V. Mseg: A composite dataset for multi-domain semantic segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE; 2020, p. 2876–2885.

[61] Lin, Y, Wang, C, Zhai, D, Li, W, Li, J. Toward better boundary preserved supervoxel segmentation for 3d point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 2018;143:39–47.

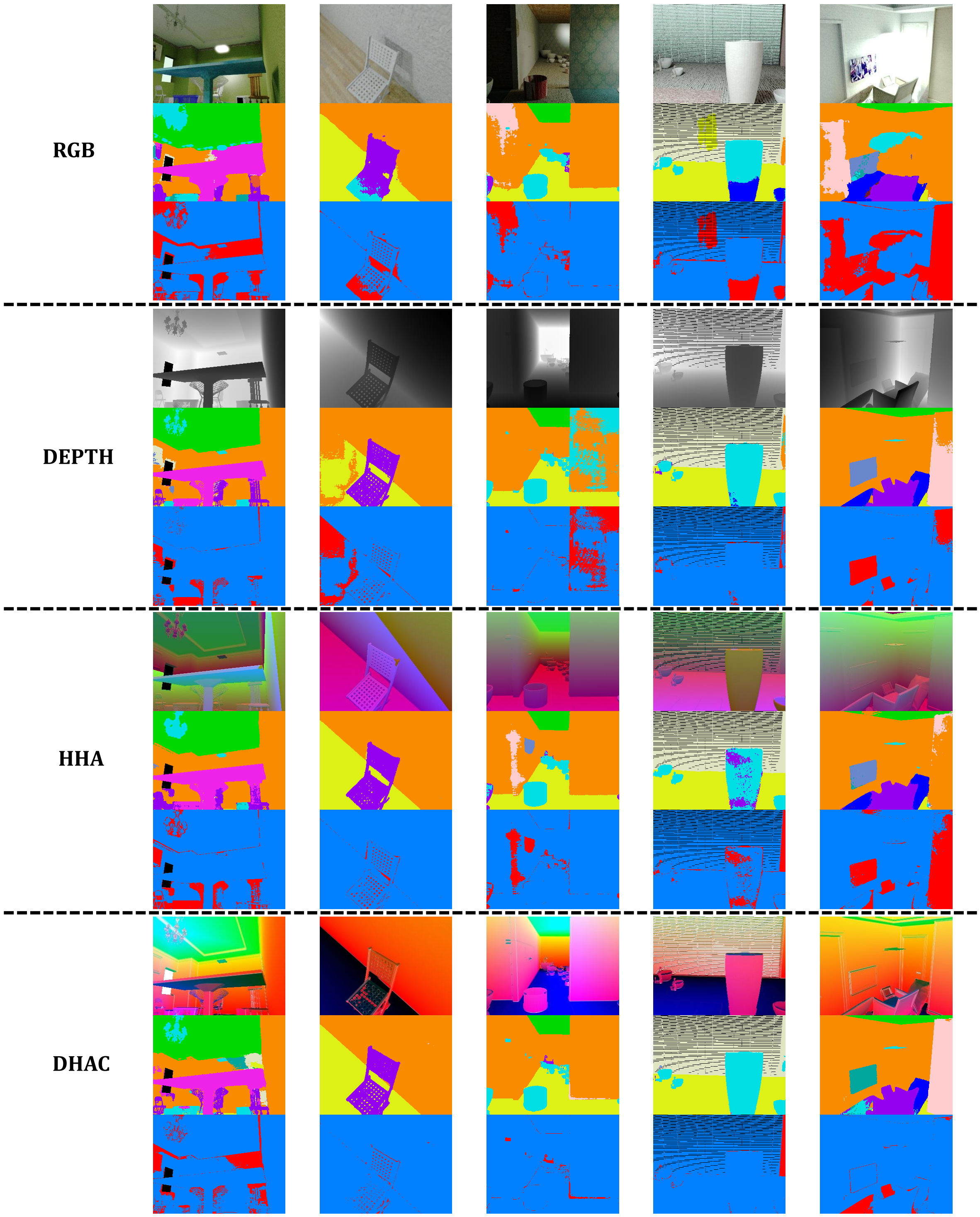


Figure 7: Examples of semantic segmentation on SceneNet RGB-D dataset with single modalities including RGB, Depth, HHA and DHAC. For each modality, the first row shows the input, the second row presents the semantic segmentation results, and the third row shows the error maps, where blue represents the correct predictions and red represents the wrong ones.

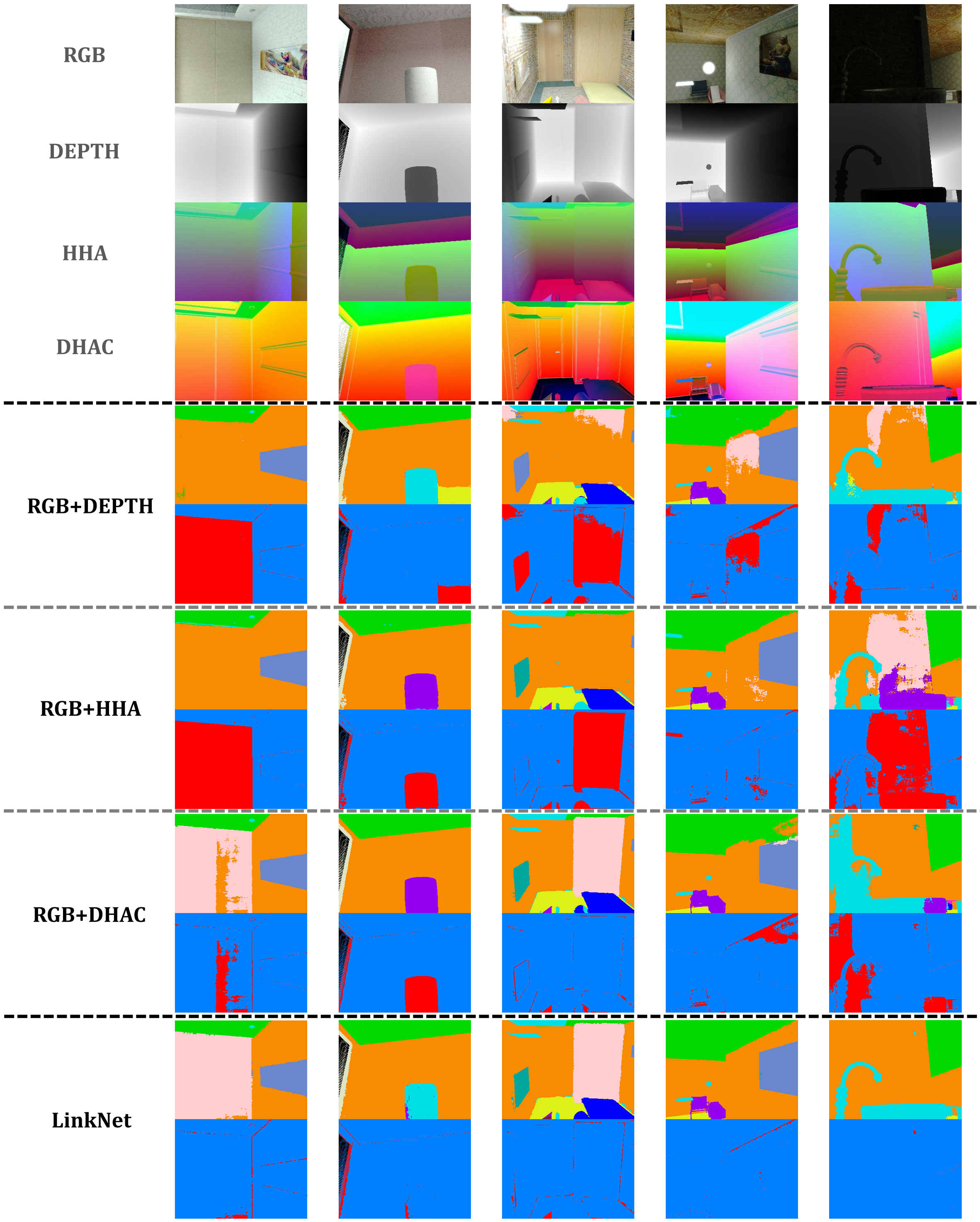


Figure 8: Examples of semantic segmentation on SceneNet RGB-D dataset with multi-modal inputs. The first block containing four rows shows different modalities, and remaining blocks are multi-modal comparisons, where within each block the first row is the result shows semantic segmentation results, and the second row gives the error maps (blue represents the correct predictions and red represents the wrong ones).