International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 12 Issue 2 ǁ February 2024 ǁ PP. 156-166

Pruned VoRTX for lightweight 3D indoor scene reconstruction method

Wei Li [1] and Wenju Wang [1,*]
[1]*College of Communication and Art Design, University of Shanghai for Science and Technology Shanghai 200093, China; 213342987@st.usst.edu.cn (W.L.)*
[*1] *College of Communication and Art Design, University of Shanghai for Science and Technology Shanghai 200093, China ; wangwenju@usst.edu.cn (W.W.)*
Corresponding Author: Wenju Wang

Abstract
With the rapid development of deep learning technology, 3D indoor scene reconstruction has been widely applied in autonomous navigation, robotics and augmented/virtual reality, among others. However, current 3D indoor scene reconstruction methods have the best model resolution in terms of mesh representation while putting more pressure on the hardware in terms of computational resources. To this end, this paper proposes a lightweight deep learning network model Prued-Vortx for 3D indoor scene reconstruction. The network mainly consists of 2DCNN with global unstructured pruning, transformer fusion network with local unstructured pruning and 3DCNN module with local structured pruning. Global unstructured pruning of 2DCNN is mainly used to extract features from images. Local non-agency pruned transformer for fusion of 3D features obtained by inverse projection of extracted 2D features. The 3DCNN module for structured pruning is used to refine features and predict TSDF values to represent the 3D scene model. On the open source ScanNet dataset, the proposed method achieves 0.632 on the reconstruction performance index F1-score.The experimental results show that this method guarantees advanced accuracy performance in 3D indoor scene reconstruction while reducing computational pressure and improving reconstruction efficiency compared with existing deep learning methods.

Keywords: *3D Reconstruction, indoor scene, lightweight, deep learning*

Date of Submission: 08-02-2024 Date of acceptance: 23-02-2024

I. INTRODUCTION

Image-based 3D reconstruction i.e. inferring the 3D geometry of objects and scenes from one or more 2D images[1]. Compared with two-dimensional images, three-dimensional models, because of more one-dimensional information, so it is more able to the object's sense of reality and details of the texture of the performance, and this aspect of the research has been applied to many areas, such as virtual reality[2], robotics[3] and other areas. 3D reconstruction of images can be divided into 3D scene reconstruction and 3D object reconstruction according to the reconstructed objects. Among them, the reconstruction of outdoor scenes[4]and individual objects[5,6]has been widely studied[7]. However, there are fewer studies on indoor scene reconstruction, especially those based directly on color images. With the continuous development of deep learning in recent years, research in 3D reconstruction of indoor scenes based on RGB images has progressed accordingly. However, it is extremely challenging to recover the missing information in the RGB maps during the 3D reconstruction process, Therefore, the research on algorithmic models for reconstructing 3D indoor scenes based on RGB images has become a hot topic.

Based on the timeline,3D indoor scene reconstruction based on RGB images can be classified into traditional and deep learning methods.

Traditional methods for reconstructing 3D interior scenes[8-13],this type of method tends to estimate the depth of each view, and then fuses the estimated depths of the different views to reconstruct the 3D scene. However, due to the difficulty of matching features in untextured areas (e.g. floors, white walls) by hand with these methods. So much so that some later depth-based approaches[14-17] incorporate deep learning to extract global features, which is more robust for matching.

Deep learning methods can be broadly categorised into voxel-based reconstruction, point cloud-based reconstruction and mesh-based reconstruction according to the representation of the model.

Methods based on voxel representation , a semantic scene completion network based on 3DCNN first proposed by Song et al. SSCNet[18] enables complete scene reconstruction and semantic segmentation ,

however, this network has a high GPU. The network has since been improved, for example, ScanComplete[19] uses a coarse-to-fine full convolutional network (FCN) to extend the SSCNet to the point where it can handle a different range of scenarios. Zhang et al[20] applied a dense (conditional random field) CRF model and then used SSCNet to further improve accuracy. However, all of the above work is based on voxel representation of the work output 3D models with low resolution.

Methods based on point cloud representations: Wei Yin et al. unify depth estimation and 3D reconstruction by proposing a two-stage framework[21] that predicts the depth and camera focal length in a single RGB image separately and unifies them in the same model. The model can reconstruct a 3D point cloud from any single RGB image, but is somewhat impotent in cases such as camera radial distortion and images with rare viewing angles or extreme focal lengths. Guangkai Xu et al[22], in order to solve the problem of depth inconsistency in existing monocular continuous image estimation, proposed a locally weighted linear regression method to recover the scales and offsets of anchor points with very sparse. Specifically the method utilises locally weighted linear regression as a new metric depth alignment method and enhances spatial smoothing, recovering accurate 3D point cloud scene shapes but also ignoring the detail component. Although the visualisation of point cloud reconstruction is more realistic, data noise, data noise and data noise are still insurmountable drawbacks, and the geometric accuracy of the reconstructed point cloud is far less than that of the mesh model and is difficult to edit directly.

Methods based on grid representations: Truncated Signed Distance Function (TSDF[23]),the truncated distance (T) is proposed on the basis of Signed Distance Function (SDF[24]), has often been used to represent complete 3D surfaces that return directly to the entire scene in recent years. The introduction of TSDF for 3D indoor scene reconstruction represented by Atlas [25] opens up a new workflow, where independent features extracted from multiple consecutive images using 2D CNNs and back-projected into the voxel volume are accrued, the 3D codec refines the accrued voxels and predicts the TSDF values, and finally this prediction is extracted to the surface mesh via the marching cubes[26].Although this method achieves the reconstruction of the complete indoor scene, the computational process of averaging views is prone to reconstructing to disjoint areas. NeuralRecon[27] improved Atlas to use the gated recurrent unit model GRU[28] in recurrent neural networks (RNNs) to globally fuse local modules reconstructed from each image from coarse to refined grid models, but because of the temporal or structural recursive nature of the RNN the training time is long. Dejan Azinovi´ c et al.'s Neural RGB-D [29]uses the Neural Radiance Fields (NeRF[30]) formulation to learn Truncated Symbol Distance Fields (TSDFs) and to jointly optimise the scene representation network and the camera poses, which successfully improves on the current Neural Radiance Fields (NeRFs) technique based on implicit representations without reconstructing the actual surfaces, but with the loss of high-frequency local details in large scenes. To address the problem of high-frequency details, go-surf[31] improved Neural RGB-D by combining for the first time a learnable feature volume with SDF surface-based representation and rendering, and devised an SDF regularisation term to achieve faster and higher-fidelity reconstruction, but there is still the limitation of memory consumption. The NeuralRoom[7] system first obtains distance a priori and normal a priori to ensure the accuracy of the reconstruction details and limit the geometric features of the untextured region, respectively, and devises a perturbed residual-limited smoothing method to further improve the reconstruction quality of the planar region. Although this method guarantees a detailed and complete reconstruction, large positional and a priori errors can have a significant negative impact on the reconstruction results and require significant computational resources. TransformerFusion[32], as the first Transformer model[33] network architecture for monocular 3D scene reconstruction, uses the transformer to decode the features extracted from the input image into the scene volume after fusing them into a high-resolution 3D scene. Although the method achieves accurate surface reconstruction, the network architecture takes into account all relevant view information resulting in computational wastage. VoRTX[34] takes advantage of the natural mechanism of occlusion perception provided by the Transformer, where the attention to each input view varies with the 3D position, reducing the attention to the input image in the occluded regions of the view, with better reconstruction results. While mesh scene models can be used directly for subsequent editing, the 3D mesh scenes obtained by these methods are transformed from voxels making it difficult to balance computational resources with the resolution of the reconstructed model.

In order to reduce the pressure on computational resources by fully compressing the number of training model parameters while ensuring the accuracy of model reconstruction, this paper proposes a lightweight deep learning network model for 3D indoor scene reconstruction. The network uses 2DCNN with global unstructured pruning to extract image features, a transformer with local unstructured pruning to fuse the 3D features obtained by inverse projection of the extracted 2D features and 3DCNN with structured pruning to refine the 3D features, which achieves the compression of the network model and reduces the amount of computation while guaranteeing the existing high resolution reconstruction of indoor scenes.

II. RELATED WORK

2.1 FEATURE EXTRACTION METHODS

AlexNet[35] proposed by Alex Krizhevsky et al. showed excellent performance on the ImageNet Large Scale Visual Recognition Challenge[36] (ILSVRC) image recognition competition, and AI has entered a new period of development dominated by deep learning techniques. In computer vision, deep neural networks have also become a mainstream method for extracting image features how Kaiming et al. designed a residual network (ResNet)[37] to add the features obtained from certain layers of a neural network by skipping the next layer of neurons spaced apart, which solves the degradation problem due to the increase in the number of layers of the network so that the depth of the network can be up to 1202 layers. Gao Huang et al. further extended the residual network by proposing the DenseNet model[38], which densely connects all the previous layers with the later layers and splices the feature maps from different layers in the channel dimension to achieve feature reuse and improve the efficiency. Andrew Howard et al. developed the Mobilenet family of networks V1[39], V2[40] and V3[41] including innovations such as depth-separable convolution, inverted residual structure and the addition of the squeeze and excitation (SENet[42]) channel attention module to significantly improve feature extraction accuracy. Mingxing Tan et al.'s MNASNet[43] employs some of the similarities to Mobilenet, such as depth-separable convolution, and uses multi-objective optimisation and hierarchical NAS to simultaneously optimise both accuracy and latency, which allows for the exploration of the most appropriate network model on its own. Transformer networks[33] have long been shown to be very effective for natural language processing[44] and have since been migrated to a variety of visual tasks[45,46], which have also shown excellent performance in capturing richer information when processing images by virtue of self-attention and multi-attention mechanisms.

2.2 MODEL LIGHTWEIGHTING METHODS

As the depth of the network model increases resulting in a large number of parameters, the computational resources and speed of computation are increasingly demanding. Therefore, many lightweight modeling approaches have been developed, such as Hinton's Knowledge Distillation (KD[47]) idea that compresses informational knowledge from complex models (i.e., teacher models) into small, computationally efficient models (i.e., student models) to achieve the performance of a complex model while maintaining its fast computational speed. The practice of knowledge distillation has received significant attention in recent years[48-51], but often pre-trained large models are required for task-specific knowledge distillation, which may not always be available or readily accessible, and the quality and applicability of the large model can have a significant impact on the performance of the small model. Model Quantization[52] refers to the conversion of floating-point algorithms for neural networks into converting to low-bit-width numerical representations for the purpose of compressing and accelerating deep neural networks. The technique has achieved a more mature application in industry, with companies such as Qualcomm, NVIDIA, and Google releasing industry-related white papers[53] and model quantisation deployment frameworks[54,55], but when using quantised models for inference, it may add a certain amount of computational overhead due to the need for additional quantisation and inverse quantisation operations. In addition to knowledge distillation and model quantisation, two methods for lightweighting models, pruning has also been shown to be very effective and practical in many network compression paradigms[56-58].The goal of network pruning is to remove redundant parameters from a given network to reduce its size and potentially speed up inference. Mainstream pruning methods can be categorised into structural pruning where pruning is based on the structure of the model[59,60] and unstructural pruning where pruning is based on the weights of the model[61,62].

III. OUR METHOD

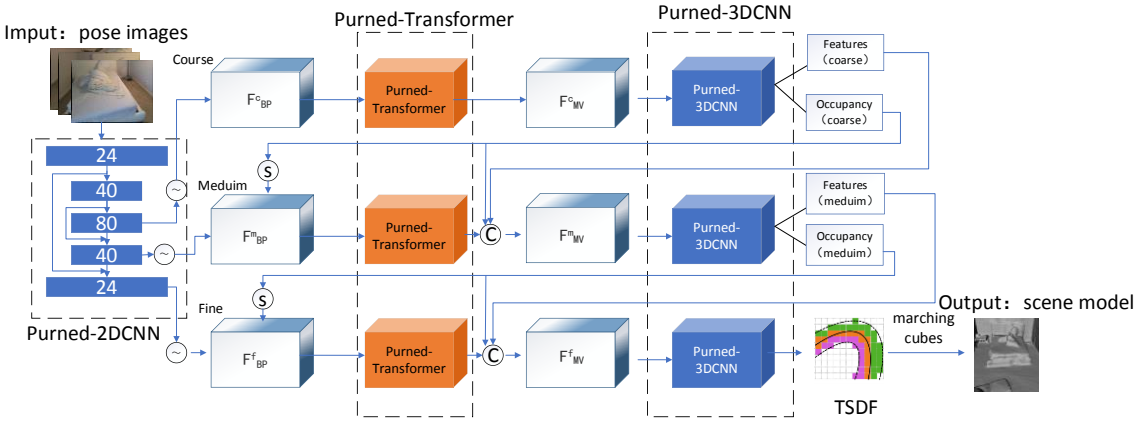

Figure 1: Model Structure of Prued-VoRTX Indoor Scene Reconstruction Approach

The lightweight indoor scene reconstruction network takes monocular RGB images and the corresponding positional information of each image as input, and reconstructs a complete high-precision 3D scene model with low hardware requirements. The network mainly consists of a globally unstructured pruned 2DCNN, a locally unstructured pruned transformer fusion network and a structured pruned 3DCNN(as shown in Figure 1, ~ denotes the reverse projection. Ⓢ denotes the predicted sparse mesh. Ⓒ denotes a splice operation. $F_{BP}^{(r)}$ denotes a batch of 3D voxel features obtained by backprojection. $F_{MV}^{(r)}$ denotes a voxel feature obtained by fusing the $F_{BP}^{(r)}$. (a) Global Unstructured Pruning 2DCNN: Unstructured pruning operation is applied to the 2DCNN network that extracts image features from the whole network, and the image features are efficiently and accurately extracted under small computational pressure and inversely projected onto 3D voxels to obtain 3D features. (b) Local unstructured pruning transformer fusion network: a transformer network after local unstructured pruning of the fully connected layer to fuse the voxel 3D features obtained by backprojection. (c) Structured Pruning 3DCNN: The structured pruned 3DCNN network directly reduces the number of parameters to compress the network model while outputting refined features and voxel block occupancy ratios for use by the next layer of the network or the last layer outputs TSDF[23] values for the Marching Cubes[26] technique to extract the scene surface and reconstruct the 3D indoor scene model.Detailed descriptions of the global unstructured pruned 2DCNN network, the local unstructured pruned transformer fusion network and the structured pruned 3DCNN network modules are given in Sections 3.1, 3.2 and 3.2 respectively.

3.1 GLOBAL UNSTRUCTURED PRUNING 2DCNN

Considering that the memory consumption of 2DCNN for extracting features in the image is relatively small compared to the rest of the entire network, this paper introduces the MNASNet[43] network and aligns it to do a global unstructured pruning operation, and sets the parameter of the least-influential convolution kernel in the entire 2DCNN network to 0 at a rate of 25% for extracting image features. Although the network is designed to autonomously explore the most appropriate network model, the large size of the network parameters still produces unnecessarily easy parameters. Therefore, we use a global unstructured pruning operation on the network to filter out many unimportant parameters to achieve efficient extraction of network features while reducing some of the memory consumption, as shown in Figure 2:

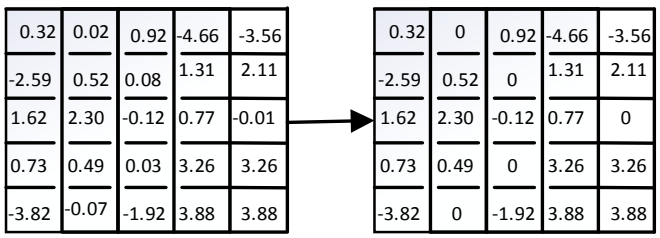

Figure 2: Unstructured Pruning 2D Convolution

For example, for a 5*5 convolution as shown in Figure 2, the weight of the convolution kernel with the smallest L1 paradigm is set to 0 according to a pruning rate of 0.25, and this pruning process can be characterized as in Eq. 1.Because this part of the parameter has less influence on the network to extract image features can be directly set to 0 to reduce the model to achieve efficient extraction of features.

$$Pruned\_MNASNet_{\min_{0.25}(|W_{kernel\_i}|)} = 0 \qquad (1)$$

W_{kernel_i} denotes the convolutional kernel weight, $\min_{0.25}$ denotes the number of parameters in the convolutional kernel with the smallest weight as a proportion of 0.25.

The total 2D feature extraction formula is shown in Equation 2 below:

$$\{F_I^c, F_I^m, F_I^f\} = Pruned\_MNASNet(I) \qquad (2)$$

$Pruned\_MNASNet$ denotes the pruned 2DCNN network, I denotes the input image, $\{F_I^c, F_I^m, F_I^f\}$ Indicates the extracted image features from coarse to fine, The feature is then back-projected into the world coordinate system to give a 3D voxel using internal and external camera parameters according to Equation 3.

$$[i,\ j,\ k]^T = [KtPt]^{-1}[u,v]^T \qquad (3)$$

In formula 3, $[u,v]$ Indicates pixel coordinates, Kt and Pt Indicates the internal and external parameters of the camera respectively, $[i,\ j,\ k]$ Indicates three-dimensional coordinates. So according to this formula the extracted 2D features can be assigned to the corresponding 3D voxels as is, as in Equation 4:

$$F_{BP}^{(r)}(:,i,j,k) = F_I^{(r)}(:,u,v) \qquad (4)$$

$F_I^{(r)}$ denotes the features extracted to the Ith image, $F_{BP}^{(r)}$ denotes a 3D voxel feature that projects a 2D feature onto a 3D voxel. r = c/m/f denotes all features from coarse to fine, It can be expressed by the formula $\{F_{BP}^c, F_{BP}^m, F_{BP}^f\}$.

3.2 TRANSFORMER 3D FEATURE FUSION FOR LOCAL UNSTRUCTURED PRUNING

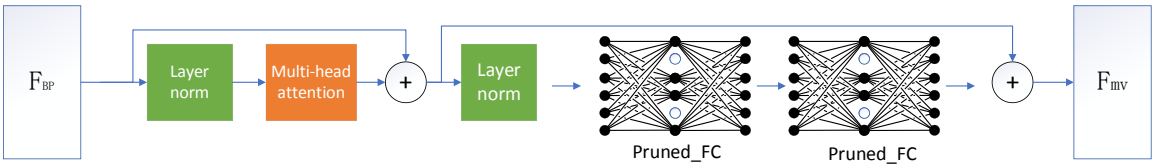

Figure 3: Transformer Fusion Network Architecture with Unstructured Pruning

In Section 3.1 Extracting 2D features are multiple monocular images together to extract the resulting features, and the features extracted from each image are assigned to the 3D voxels. We therefore use Transformer[34] as in VoRTX[34] to fuse these features into one voxel mesh feature. Considering that the Transformer network is already powerful enough, this paper only prunes the local fully-connected layers of the network to ensure that the network model is compressed without affecting the results.The Transformer fusion network is shown in Figure 3.

F_{BP} indicates the voxel characteristics to be integrated, Layer norm representation layer normalisation, Multi-head attention denotes the mechanism of multi-head attention, F_{MV} indicates fused features, Pruned_FC denotes the fully connected layer after pruning.

The unstructured pruned fully connected layer in Figure 3 is similar to dropout randomly inactivated neurons, but with the difference that dropout exists only during training and inactivates randomly. Unstructured pruning of the fully connected layer deactivates in a manner similar to the method of filtering parameters in Eq. (1), which can be characterised as Eq. (5).

$$Pruned\_Transformer_{\min_{0.33}(|FC_W|)} = 0 \qquad (5)$$

$Pruned\_Transformer$ denotes the pruned Transformer fusion network, FC_W denotes the weight of the fully connected layer in the Transformer network, $\min_{0.33}$ denotes the weight parameter with the smallest proportion of neuronal connections of 0.33.The overall integration equation can be characterised as equation (6):

$$(F_{MV}^{(r)}) = Pruned\_\text{Transformer}(F_{BP}^{(r)}) \quad (6)$$

$Pruned\_\text{Transformer}$ denotes the Transformer network after local unstructured pruning, $F_{BP}^{(r)} \in R^{N*H^{(r)}*W^{(r)}*C^{(r)}}$, $F_{MV}^{(r)} \in R^{H^{(r)}*W^{(r)}*C^{(r)}}$ is the voxel feature obtained by projection,N is the number of images input at once (bit-size),H and W are the resolution C is the number of channels of the extracted features, r = c m or f. So $F_{MV}^{(r)}$ has one less dimension than $F_{BP}^{(r)}$ in terms of number of images i.e. the features extracted individually from the input images are fused to a voxel feature.

3.2 STRUCTURED PRUNING 3DCNN REFINEMENT OF 3D FEATURES

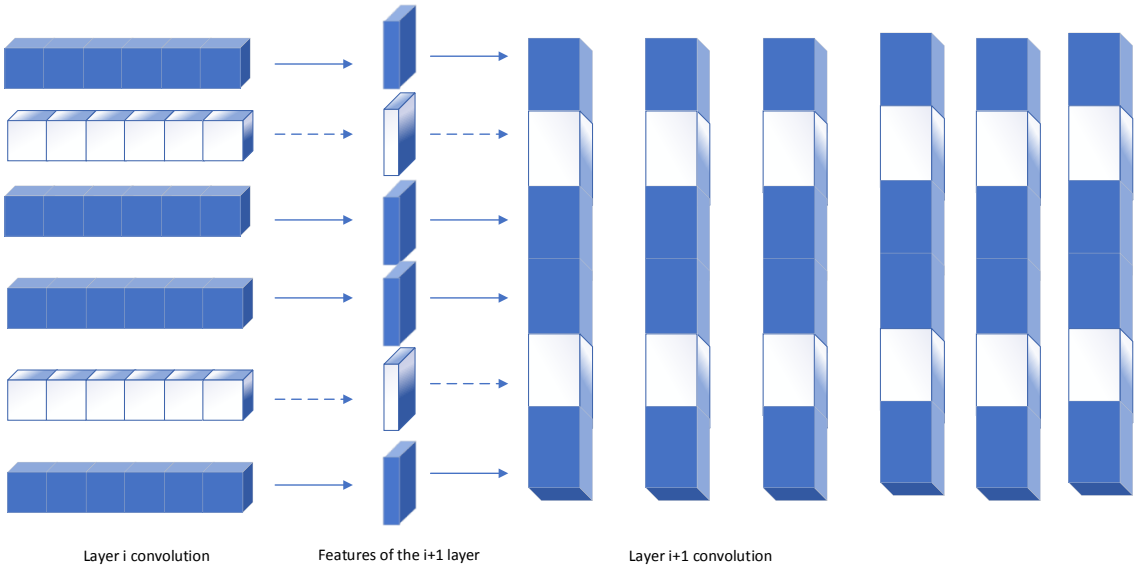

Figure 5: Schematic of 3DCNN Structured Pruning

In indoor scene reconstruction, the most memory consuming is the 3DCNN module, so for this module this paper uses structured pruning. Unlike unstructured pruning, structural pruning directly cuts away the convolutional/fully connected part of the network where the less influential parameters are located, which intuitively reduces the number of parameters and reduces computer memory consumption while keeping the sacrificed performance within acceptable limits.

The structured pruning introduced in this paper 3 The pruning of a layer structure of the DCNN is shown schematically in Figure 5.The white square on the left of the figure indicates that the entire convolutional kernel of layer i with a small parameter impact is clipped, and the corresponding output feature (the white part) is naturally absent, resulting in the convolutional kernel for extracting this absent feature in the convolution of layer i+1 also needing to be clipped, drastically reducing the number of parameters in the network. The structure of the screening unimportant convolutional kernel in this paper is shown in Equation (7):

$$Pruned\_3DCNN_{\min_{0.25}(|W|)} = 0 \quad (7)$$

$Pruned\_3DCNN$ denotes 3DCNN after pruning, W denotes the entire convolutional layer in 3DCNN, $\min_{0.25}$ denotes the convolutional kernel with the smallest second-paradigm number of the entire convolutional layer accounting for the 0.25 parameter.

The total 3D feature refinement is shown in equation (8)

$$(O^{(r)}) = Pruned\_3DCNN(F_{MV}^{(r)}) \quad (8)$$

$F_{MV}^{(r)}$ indicates voxel characteristics after fusion, $Pruned\_3DCNN$ represents a structured pruning operation, $O^{(r)}$ indicates the refined output features, when r = f, it is the value of the final regression tsdf.

In this paper, the predicted TSDF[23] values are shown in Figure1 by unstructured global pruning of 2DCNN, unstructured local pruning of Transformer and structured pruning of 3DCNN to predict the optimal computational results to be obtained.See Equation (9) for the calculation of TSDF[23] values and SDF[24]:

$$sdf_j = \| t_{pjw} - t_{cjw} \| - dep(I_{ij}) \quad (9)$$

Eq. t_{pjw} denotes the location information of the jth voxel under the world coordinate system, t_{ciw} represents the position information of the camera in the world coordinate system, $dep(I_{ij})$ denotes the depth of ij pixels in the I image. Truncate after completing the calculation of SDF values for each voxel with this equation: when the SDF value is greater than -1 is not more than 1 when set directly to TSDF value, less than -1 is set to -1, greater than 1 is set to 1.The closer the value of TSDF obtained by the voxel is to 0, the closer it can be approximated that is the surface of the model to be reconstructed, less than 0 means before the surface of the model, and the part greater than 0 means after the surface. The acquired TSDF[23]is combined with the Marching Cubes[26] algorithm to find equivalent surfaces in the voxels as reconstructed surfaces to be sliced out for the reconstruction of the indoor scene.

IV. Experiment

4.1 EXPERIMENT DATASET AND EXPERIMENTAL ENVIRONMENT

The ScanNet[63] dataset was chosen for training tests to validate the effectiveness of our proposed method, which is a large-scale 3D dataset of indoor scenes created and maintained by the Department of Computer Science at Stanford University, the Department of Computer Science at Princeton University, and the Stanford AI Laboratory. There are 1513 scenes in the dataset, containing more than 2.5 million indoor scene images with corresponding camera poses, mesh models, semantic labels, instance labels and CAD models, of which 1201 scenes are used for training and 312 scenes are used for testing.

The experimental environment is Ubuntu18.04 system which has Intel core i9-12900K CPU, 128GB RAM and NVIDIA GTX 3090 graphics processing unit (GPU). And we used pytorch1.11.0, PyTorch Lightning and cuda11.3 for deep network training. We initialised the learning rate to 0.001 and used the Adam optimiser[64] for gradient updating.

4.2 EVALUATION METRICS

For the evaluation metrics, we use Acc, Comp, Prec, Rec and F-score proposed in Atlas[25] as the evaluation metrics for reconstruction, as defined in Table 1.

Tabel 1: Definitions of Metrics

Metrics	Definitions
Acc↓	$mean_{p \in P}(mean_{p^* \in P^*} \| p - p^* \|)$
Comp↓	$mean_{p^* \in P^*}(mean_{p \in P} \| p - p^* \|)$
Prec↑	$min_{p \in P}(min_{p^* \in P^*} \| p - p^* \|) < .05$
Rec↑	$min_{p^* \in P^*}(min_{p \in P} \| p - p^* \|) < .05$
F-score↑	$F - score = \frac{2\,\mathrm{Pr}ec * \mathrm{Re}cal}{\mathrm{Pr}ec + \mathrm{Re}cal}$

where p denotes the position of the point in the reconstructed 3D space, p^* represents the position of a point in three-dimensional space,↓ Indicates that the lower the value of the evaluation indicator, the better the performance,↑ Indicates that higher values of evaluation indicators are better. Of these, the F-score is the most important reconstruction metric, as the accuracy and completeness of the reconstruction is taken into account.

4.3 EXPERIENTAL RESULTS AND ANALYSIS

4.3.1 COMPARISON OF VISUALIZATION RESULTS

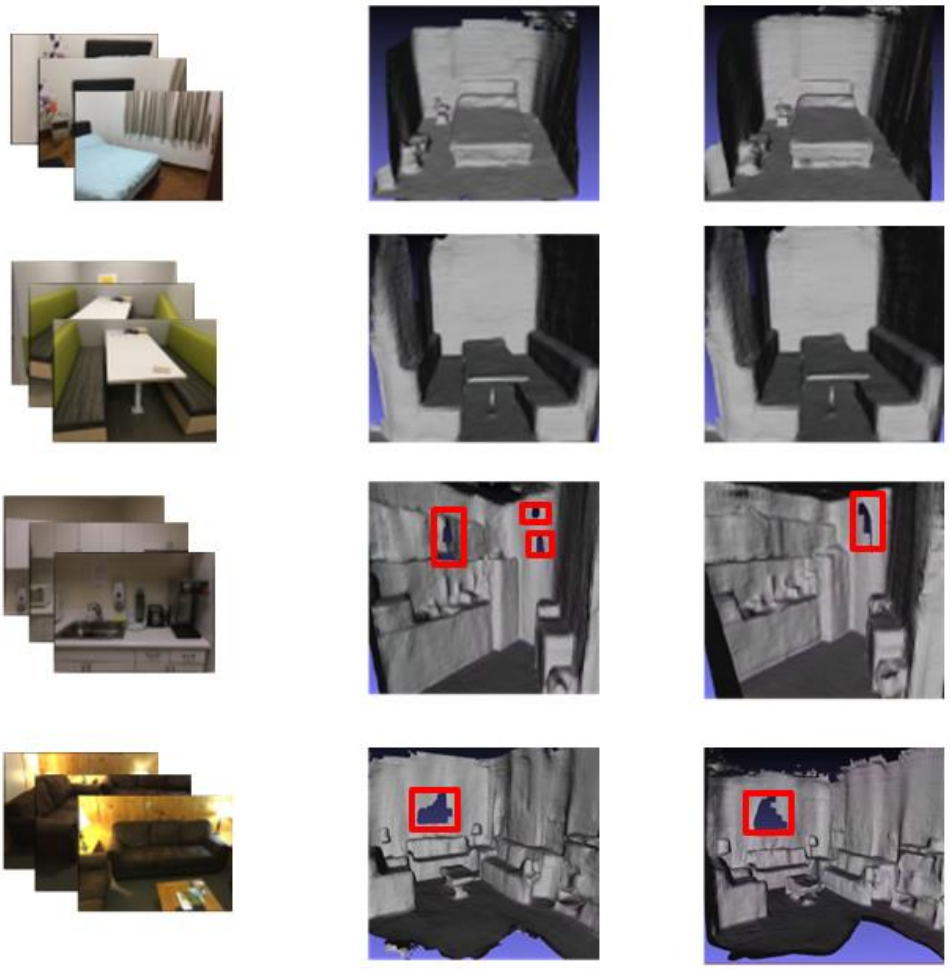

Figure 6: Comparison Of 3D Reconstruction Visualization Results

As shown in Figure 6, the left is the input monocular images, the center is the reconstructed model maps from the images by the VoRTX [34] method, and the right is the reconstructed model maps by our proposed method. In the third row, the model we reconstructed has relatively large holes while there are more small holes in the model reconstructed by the VoRTX method. We can see that the reconstruction results of our method and the VoRTX method are basically equal.

4.3.2 COMPARISON OF PERFORMANCE METRICS

Table 2: Reconstruction Accuracy Comparison (optimal results bolded)

Metrics	**Atls[25]**	**Recon[27]**	**Vortx[34]**	**Ours**
Acc↓	0.068	**0.049**	0.054	0.055
Comp↓	0.098	0.133	**0.090**	0.093
Prec↑	0.640	0.691	**0.708**	0.699
Rec ↑	0.539	0.461	**0.588**	0.586
F-core↑	0.583	0.551	**0.641**	0.632

As can be seen from the data in Table 2, the Acc↓ metric reaches 0.068 by Atls[25] method, 0.049 by Recon[27] method, and 0.054 by Vortx[34] method. It reaches 0.055 by our method, which is 0.013 lower than Atls, 0.007 higher than Recon, and 0.001 higher than Vortx. The Comp↓ metric reaches 0.098 by Atls[25] method, 0.133 by Recon[27] method, and 0.090 by Vortx[34] method. It reaches 0.093 by our method, which is 0.005 lower than Atls, 0.040 lower than Recon, and 0.003 higher than Vortx. The Prec↑ metric reaches 0.640 by Atls[25] method, 0.691 by Recon[27] method, and 0.708 by Vortx[34] method. It reaches 0.699 by our method, which is 0.059 higher than Atls, 0.008 higher than Recon method, and 0.009 lower than Vortx method. The Pec↑ metric reaches 0.539 by Atls[25] method, 0.461 by Recon[27] method, and 0.588 by Vortx[34] method. It reaches 0.586 by our method, which is 0.047 higher than Atls, 0.125 higher than Recon, and 0.003 lower than Vortx. The F-core↑ metric reaches 0.583 by Atls[25] method, 0.551 by Recon[27] method, and 0.641 by Vortx[34] method. It reaches 0.632 by our method, which is 0.049 higher than Atls, 0.081 higher than Recon, and 0.009 lower than Vortx.

Our methods have surpassed most of the metrics of the Atlas[25] and NeuralRecon [27] methods in terms of reconstruction accuracy especially for F-core scores, although most of them are slightly inferior to the optimal metrics.

4.3.3 COMPARISON OF RECONSTRUCTION SPEED

Table 3: Reconstruction Speed Comparison

Method	Time
VoRTX[34]	628s
Ours	**555s**

In this paper, 100 scenarios from the test portion of the database are used in a unified inference reconstruction experiment for comparison. The reconstruction time reaches 628s by VoRTX[34] method based on our equivalent hardware, while 555s by our method after pruning in this paper, which obtains an 11.6% improvement in speed.

4.3.4 COMPARISON OF RECONSTRUCTION SPEED

Table 4: Network Model Sizing Analysis

Method	Weight Parameter Quantity
VoRTX[34]	6.2M
Ours	**5.22M**

Compared the proposed method in this paper with VoRTX[34], the number of trainable weight parameters in the model of VoRTX[34] is 6.2M, while 5.22M after pruning in this paper, reducing the model size by 15.9%.

4.3.5 ABLATION EXPERIMENTS

Table 5: Ablation Experiments

Method	Reconstruction Time
VoRTX[34]	628s
P2dcnn-Vortx	609s
Ptrans-Vortx	618s
P3dcnn-Vortx	581s
Ours	555s

From the data in Table 5, the reconstruction time of the VoRTX [34] method is 628s for 100 scenes when it is not pruned. It reaches 618s after only pruning the fully-connected layer of the Transformer fusion network in the network, increasing speed by 1.5%. It reaches 609s after only pruning the 2D convolution in the network, increasing speed by 3%. It reaches 581s after only pruning the 3DCNN in the network, increasing speed by 7.4%. It reaches 555s after pruning all three modules in the network, increasing speed by 11.6%. It can be seen that each pruning module proposed in this paper plays the effect of lightening the network model and works together to maximize the effect of lightening the model.

V. CONCLUSION

In this paper, we propose a lightweight 3D indoor scene reconstruction network model, which mainly uses global unstructured pruned 2DCNN mainly for extracting the features in the image and then back-projecting them onto 3D voxels to get the 3D features, local non-structured pruned transformer is used for fusing the obtained 3D features, and finally structured pruned 3DCNN module is used for refining the features and predicting the values of TSDF to represent the 3D scene model. Compared with existing methods, the reconstruction performance of the network proposed in this paper is basically the same as that of existing state-of-the-art methods, but the number of parameters in the model is 15.9% less than that of existing methods, and the reconstruction speed is 12.5% higher than that of existing methods, which achieves the effect of lightweight indoor scene reconstruction network. In the future, we will further consider how to improve the reconstruction performance combined with lightweight models to achieve faster and better reconstruction of indoor scenes.

REFERENCES

[1]. Han X-F, Laga H, Bennamoun M J I T O P A, et al. Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era[J], 2019, 43(5): 1578-1604.

[2]. El Saer A, Stentoumis C, Kalisperakis I, et al. 3D RECONSTRUCTION and MESH OPTIMIZATION of UNDERWATER SPACES for VIRTUAL REALITY[C]. 2020 24th ISPRS Congress - Technical Commission II, August 31, 2020 - September 2, 2020, 2020: 949-956.

[3]. Khurana A, Nagla K S, Sharma R: 3D Scene Reconstruction of Vision Information for Mobile Robot Applications, Soft Computing: Theories and Applications, 2020: 127-135.

[4]. Leroy R, Trouve-Peloux P, Champagnat F, et al. Pix2Point: Learning Outdoor 3D Using Sparse Point Clouds and Optimal Transport[J], 2021.

[5]. Kemelmacher-Shlizerman I. Internet Based Morphable Model[C]. IEEE International Conference on Computer Vision, 2014.

[6]. Kar A, Tulsiani S, Carreira J, et al. Category-Specific Object Reconstruction from a Single Image[J], 2014.

[7]. Wang Y, Li Z, Jiang Y, et al. NeuralRoom: Geometry-Constrained Neural Implicit Surfaces for Indoor Scene Reconstruction: arXiv, 2022.

[8]. Schonberger J L, Frahm J-M. Structure-from-Motion Revisited[C]. 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, June 26, 2016 - July 1, 2016, 2016: 4104-4113.

[9]. Bleyer M, Rhemann C, Rother C. PatchMatch stereo - Stereo matching with slanted support windows[C]. 2011 22nd British Machine Vision Conference, BMVC 2011, August 29, 2011 - September 2, 2011, 2011.

[10]. Merrell P, Akbarzadeh A, Wang L, et al. Real-time visibility-based fusion of depth maps[C]. 2007 IEEE 11th International Conference on Computer Vision, ICCV, October 14, 2007 - October 21, 2007, 2007.

[11]. Schonberger J L, Zheng E, Frahm J-M, et al. Pixelwise view selection for unstructured multi-view stereo[C]. 14th European Conference on Computer Vision, ECCV 2016, October 8, 2016 - October 16, 2016, 2016: 501-518.

[12]. Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]. 2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, October 26, 2011 - October 29, 2011, 2011: 127-136.

[13]. Shotton J, Glocker B, Zach C, et al. Scene coordinate regression forests for camera relocalization in RGB-D images[C]. 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013, June 23, 2013 - June 28, 2013, 2013: 2930-2937.

[14]. Im S, Jeon H-G, Lin S, et al. Dpsnet: End-to-end deep plane sweep stereo: arXiv, 2019.

[15]. Yao Y, Luo Z, Li S, et al. MVSNet: Depth inference for unstructured multi-view stereo[C]. 15th European Conference on Computer Vision, ECCV 2018, September 8, 2018 - September 14, 2018, 2018: 785-801.

[16]. Yao Y, Luo Z, Li S, et al. Recurrent MVSnet for high-resolution multi-view stereo depth inference[C]. 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, June 16, 2019 - June 20, 2019, 2019: 5520-5529.

[17]. Cheng S, Xu Z, Zhu S, et al. Deep Stereo Using Adaptive Thin Volume Representation with Uncertainty Awareness[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, June 14, 2020 - June 19, 2020, 2020: 2521-2531.

[18]. Song S, Yu F, Zeng A, et al. Semantic scene completion from a single depth image[C]. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, July 21, 2017 - July 26, 2017, 2017: 190-198.

[19]. Dai A, Ritchie D, Bokeloh M, et al. ScanComplete: Large-scale scene completion and semantic segmentation for 3d scans: arXiv, 2017.

[20]. Zhang L, Wang L, Zhang X, et al. Semantic scene completion with dense CRF from a single depth image[J]. Neurocomputing, 2018, 318: 182-195.

[21]. Yin W, Zhang J, Wang O, et al. Learning to Recover 3D Scene Shape from a Single Image[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, June 19, 2021 - June 25, 2021, 2021: 204-213.

[22]. Xu G, Yin W, Chen H, et al. Towards 3D Scene Reconstruction from Locally Scale-Aligned Monocular Video Depth: arXiv, 2022.

[23]. Werner D, Al-Hamadi A, Werner P. Truncated signed distance function: Experiments on voxel size[C]. 11th International Conference on Image Analysis and Recognition, ICIAR 2014, October 22, 2014 - October 24, 2014, 2014: 357-364.

[24]. Curless B, Levoy M. A volumetric method for building complex models from range images[C]. 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, August 4, 1996 - August 9, 1996, 1996: 303-312.

[25]. Murez Z, Van As T, Bartolozzi J, et al. Atlas: End-to-End 3D Scene Reconstruction from Posed Images[C]. 16th European Conference on Computer Vision, ECCV 2020, August 23, 2020 - August 28, 2020, 2020: 414-431.

[26]. Lorense W E J P S. A High Resolution 3D Surface Construction Algorithm[J], 1987.

[27]. Sun J, Xie Y, Chen L, et al. Neuralrecon: Real-time coherent 3D reconstruction from monocular video[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, June 19, 2021 - June 25, 2021, 2021: 15593-15602.

[28]. Cho K, Van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]. 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25, 2014 - October 29, 2014, 2014: 1724-1734.

[29]. Azinovi D, Martin-Brualla R, Goldman D B, et al. Neural RGB-D Surface Reconstruction: arXiv, 2021.

[30]. Mildenhall B, Srinivasan P P, Tancik M, et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis[C]. 16th European Conference on Computer Vision, ECCV 2020, August 23, 2020 - August 28, 2020, 2020: 405-421.

[31]. Wang J, Bleja T, Agapito L. GO-Surf: Neural Feature Grid Optimization for Fast, High-Fidelity RGB-D Surface Reconstruction: arXiv, 2022.

[32]. Boi A, Palafox P, Thies J, et al. TransformerFusion: Monocular RGB Scene Reconstruction using Transformers[C]. 35th Conference on Neural Information Processing Systems, NeurIPS 2021, December 6, 2021 - December 14, 2021, 2021: 1403-1414.
[33]. Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[C]. arXiv, 2017.
[34]. Stier N, Rich A, Sen P, et al. VoRTX: Volumetric 3D Reconstruction with Transformers for Voxelwise View Selection and Fusion[C]. 9th International Conference on 3D Vision, 3DV 2021, December 1, 2021 - December 3, 2021, 2021: 320-330.
[35]. Krizhevsky A, Sutskever I, Hinton G E J a I N I P S. Imagenet classification with deep convolutional neural networks[J], 2012, 25.
[36]. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J], 2014: 1-42.
[37]. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 770-778.
[38]. Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 4700-4708.
[39]. Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J], 2017.
[40]. Howard A, Zhmoginov A, Chen L-C, et al. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation[J], 2018.
[41]. Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]. Proceedings of the IEEE/CVF international conference on computer vision, 2019: 1314-1324.
[42]. Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018: 7132-7141.
[43]. Tan M, Chen B, Pang R, et al. Mnasnet: Platform-aware neural architecture search for mobile[C]. 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, June 16, 2019 - June 20, 2019, 2019: 2815-2823.
[44]. Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J], 2018.
[45]. Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows[J], 2021.
[46]. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[C]. International Conference on Learning Representations, 2021.
[47]. Hinton G, Vinyals O, Dean J J C S. Distilling the Knowledge in a Neural Network[J], 2015, 14(7): 38-39.
[48]. Tang R, Lu Y, Liu L, et al. Distilling task-specific knowledge from BERT into simple neural networks: arXiv, 2019.
[49]. Yim J, Joo D, Bae J, et al. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning[C]. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, July 21, 2017 - July 26, 2017, 2017: 7130-7138.
[50]. Wang X, Sun Y, Zhang R, et al. KDGAN: Knowledge distillation with generative adversarial networks[C]. 32nd Conference on Neural Information Processing Systems, NeurIPS 2018, December 2, 2018 - December 8, 2018, 2018: 775-786.
[51]. Shrivastava A, Qi Y, Ordonez V. Estimating and Maximizing Mutual Information for Knowledge Distillation[C]. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2023, June 18, 2023 - June 22, 2023, 2023: 48-57.
[52]. Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: A whitepaper, 2018.
[53]. Nagel M, Fournarakis M, Amjad R A, et al. A white paper on neural network quantization: arXiv, 2021.
[54]. Reed J K, Devito Z, He H, et al. TORCH.FX: PRACTICAL PROGRAM CAPTURE AND TRANSFORMATION FOR DEEP LEARNING IN PYTHON: arXiv, 2021.
[55]. Siddegowda S, Fournarakis M, Nagel M, et al. Neural Network Quantization with AI Model Efficiency Toolkit (AIMET): arXiv, 2022.
[56]. Ding X, Hao T, Tan J, et al. ResRep: Lossless CNN Pruning via Decoupling Remembering and Forgetting[C]. 18th IEEE/CVF International Conference on Computer Vision, ICCV 2021, October 11, 2021 - October 17, 2021, 2021: 4490-4500.
[57]. Gao S, Huang F, Cai W, et al. Network pruning via Performance Maximization[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, June 19, 2021 - June 25, 2021, 2021: 9266-9276.
[58]. Wang W, Chen M, Zhao S, et al. Accelerate CNNs from Three Dimensions: A Comprehensive Pruning Framework[C]. 38th International Conference on Machine Learning, ICML 2021, July 18, 2021 - July 24, 2021, 2021: 10717-10726.
[59]. Ding X, Ding G, Guo Y, et al. Centripetal SGD for pruning very deep convolutional networks with complicated structure[C]. 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, June 16, 2019 - June 20, 2019, 2019: 4938-4948.
[60]. You Z, Yan K, Ye J, et al. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks: arXiv, 2019.
[61]. Dong X, Chen S, Pan S J. Learning to Prune Deep Neural Networks via Layer-wise Optimal Brain Surgeon: arXiv, 2017.
[62]. Park S, Lee J, Mo S, et al. LOOKAHEAD: A FAR-SIGHTED ALTERNATIVE OF MAGNITUDE-BASED PRUNING[C]. 8th International Conference on Learning Representations, ICLR 2020, April 30, 2020, 2020.
[63]. Dai A, Chang A X, Savva M, et al. ScanNet: Richly-annotated 3D reconstructions of indoor scenes: arXiv, 2017.
[64]. Kingma D, Ba J J C S. Adam: A Method for Stochastic Optimization[J], 2014.