Semantic Wireframe Detection

Yiming ZHOU [1], Ahmad OSMAN [1], Marc WILLMS [1], Albrecht KUNZ [2],
Selina PHILIPP [3], Janine BLATT [3], Simon EUL [3]
[1] Fraunhofer-Institut für Zerstörungsfreie Prüfverfahren IZFP, Saarbrücken
[2] htwsaar, Saarbrücken
[3] OBG Hochbau GmbH & Co.KG, Ottweiler

Kontakt E-Mail: yiming.zhou@izfp-extern.fraunhofer.de

Kurzfassung. We present a conceptually simple and effective algorithm to detect user specified wireframes in a given indoor room image. Previous deep learning-based methods can produce great line detection results, however, they contain many redundant information for some cases. Hence, our method integrates semantic segmentation algorithm to control which part of the imagine should be detected. Segmentation is the task of clustering parts of an image together which belong to the same object class. According to the class information the proposed algorithm can show the desired results e.g. wireframe between walls and ceilings. Our method can give texture information and prepare for the following reconstruction.

Introduction

Among all the high-level geometric features, straight lines and their junctions (together called a wireframe [1]) are the most fundamental that can be used to assemble the 3D structures of a scene. By providing a well-annotated dataset encourages the research of wireframe parsing. Zhou et al. [2] proposed an end-to-end trainable system called L-CNN, using a single and unified neural network.
The task of semantic segmentation can be referred to as classifying a certain class of image and separating it from the rest of the image classes by overlaying it with a segmentation mask. It is related to image classification, since it deals with per-pixel category prediction instead of image-level prediction. Fully convolutional networks (FCNs) became a baseline and inspired many follow-up works [3]. Since there is a strong relation between classification and semantic segmentation, many state-of-the-art semantic segmentation frameworks are based on the architecture of image classification on ImageNet. The CNN methods using VGGs or ResNet has dramatically pushed the performance boundary of semantic segmentation [4].

Showing the great success in natural language processing (NLP), there has been a recent work of trying to apply Transformers to vision tasks. Dosovitskiy et al. [5] proposed vision Transformer (ViT) for image classification. The authors replace text sequences with split image patches and feed them into a standard Transformer with positional embeddings (PE), lead to an outstanding performance on ImageNet. Zheng et al. [6] proposed SETR and Xie et al. [7] proposed SegFormer to demonstrate the feasibility of using Transformers in this task. SegFormer sets a new state-of-the-art of 51.8% mIoU on ADE20K dataset. Hence, we will use SegFormer as semantic algorithm and fine-tune on that.

1. Methods

1.1 L-CNN Network Architecture

Figure 1 illustrates the L-CNN architecture. It contains four modules: 1) a feature extraction backbone based on CNN to provide shared intermediate feature maps; 2) a junction proposal module; 3) a line sampling module that proposes line representation from the junction proposal module; 4) a line verification module that verifies the proposed lines. The total loss is the sum of loss on those modules.

In computer vision tasks, the backbone network is necessary to extract semantically meaningful features for the following module. We choose stacked hourglass network [8] as the backbone because of its efficiency. The hourglass network first downsamples the input images twice through convolutional layers. After the processing, the learned feature maps are gradually refined by the hourglass modules with intermediate supervision imposed on the output of each module.

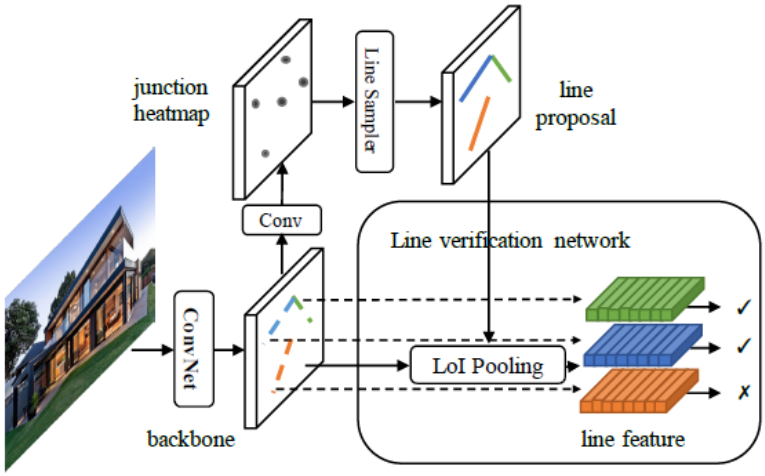


Figure 1: An overview of L-CNN network architecture [2]

1.2 Separate Module

Junction Prediction: An input image with resolution W x H is first divided into Wb x Hb bins. For each bin, the neural network predicts whether there exists a junction inside it, and if yes, it also predicts its relative location inside this bin. Mathematically, the neural network outputs a junction likelihood map J and an offset map O.

Non-Maximum Suppression: In instance-level recognition like object recognition, non-maximum suppression (NMS) is applied to remove duplicate around correct predictions. NMS can remove blurred score map around correct predictions with the same mechanism.

Line Sampler: There are two different line samplers. Static line sampler returns both positive and negative samples that are directly derived from the ground truth labels. Positive line samples are uniformly sampled from all the ground truth lines with coordinate of the corresponding junctions. We first rasterize all the ground truth lines onto a 64x64 low-resolution bitmap. Then, we choose every not ground truth line and calculate the hardness score. For each imagine, negative samples are the top 2000 lines with the highest hardness scores. The other line sampler is a dynamic line sampler which samples the lines using the predicted junctions from the junction proposal module. The static line sampler helps cold-start the training because there are only few accurate positive samples from the sampler. The dynamic line sampler improves the performance of line detection by adapting the line endpoints to the predicted junction locations.

LOI Pooling: Inspired by the RoIPool and RoIAlign layers from the object detection algorithm Faster R-CNN [9, 10], LoI Pooling layer can extract line features to check whether a line segment exists in an imagine.

1.3 SegFormer Network Architecture

As depicted in Figure 2, SegFormer hast two main modules: (1) a hierarchical Transformer encoder to generate high-resolution coarse features and low-resolution fine features; and (2) a lightweight All-MLP decoder to fuse these multi-level features to produce the final semantic segmentation mask.

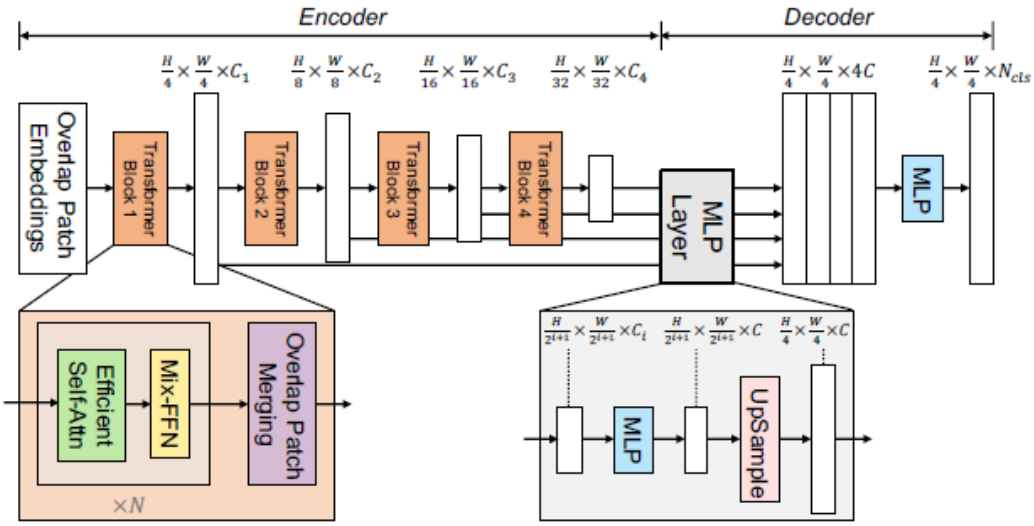


Figure 2: The proposed SegFormer framework [7]

1.3.1 Hierarchical Transformer Encoder

Hierarchical Feature Representation: ViT can only generate a single-resolution feature map, hence we need to adapt the structure to generate CNN-like multi-level features. High-resolution coarse features and low-resolution fine-grained features, which usually boost the performance of semantic segmentation.

Overlapped Patch Merging: The patch merging process used in ViT unifies an N x N x 3 patch into a 1 x 1 x C vector. We will apply this unification to hierarchical feature maps. With using an overlapping patch merging process, we can preserve the local continuity around those patches. We use the empirical definition of K=3, S=2 and P=1 from the original paper [7], where K is the patch size, S is the stride between two adjacent patches, and P is the padding size.

Efficient Self-Attention: Self-Attention [11] is widely used in NLP, and it is the main computation bottleneck. For the self-attention process, each of the heads Q, K, V have the same dimensions N x C, where N = H x W is the length of the sequence. The computational complexity is $O(N^2)$, which demands high calculation ability for large resolutions. Hence, we use the sequence reduction process introduced in [12]. This process uses a reduction ration R to reduce the length of the sequence. As a result, the complexity of the self-attention mechanism is reduced from $O(N^2)$ to $O(\frac{N^2}{R})$. In our expedients, we follow the default value and set R to [64, 16, 4, 1] from stage-1 to stage-4.

Mix-FFN: ViT uses positional encoding (PE) to introduce the location information. Instead, directly using a 3 x 3 Conv in the feed-forward network (FFN) can reach the similar effect. Mix-FFN mixed a 3 x 3 convolution and an MLP into each FFN.

1.3.2 Lightweight All-MLP Decoder

Since the hierarchical Transforer encoder has a larger effective receptive field (ERF) than traditional CNN encoders, the decoder consists only simple MLP. The proposed All-MLP decoder consists of four main steps. First, multi-level features Fi from the MiT encoder go through an MLP layer to unify the channel dimension. Then, features are up-sampled to 1/4th and concatenated together. Third, an MLP layer is adopted to fuse the tandem features F. Finally, another MLP layer takes the fused feature to predict the segmentation mask M with a H/4 x W/4 x Ncls resolution, where Ncls is the number of categories.

2. Experiments

1.2 L-CNN Implementation Details

The L-CNN is trained on the ShanghaiTech dataset [13] and also evaluated on the York Urban dataset [14]. The ShanghaiTech dataset contains 5,462 images, in which 5000 images as training set and 462 images as testing set. Author conducted this experiment on a single NVIDIA GTX 1080Ti GPU, which takes 8 hours with batch size of 6 and 16 epochs. We evaluate the well-trained model on our custom dataset and could achieve great result with 95% accuracy.

1.3 SegFormer Implementation Details

There are well-trained models on different available datasets: Cityscapes [15], ADE20K [16] and COCO-Stuff [17]. ADE20K is a scene parsing dataset of 150 fine-grained semantic concepts consisting of 20210 images. Hence, it fits our indoor room scene the best. Author trained the model on a server with 8 Tesla V100 (each 16 GB). The encoder is pre-trained on the Imagenet-1K dataset, then 160k iterations on ADE20K, Cityscapes, and 80K on COCO-Stuff iterations. Hence, the training is extreme sufficient for general tasks.
Then we need to fine-tune the well-trained model on our custom dataset. Firstly, we need to finish the annotations with an online useful tool called “segments-ai”. Our custom dataset consists of 40 training images and 15 test images. We firstly fine-tuned the model on our small custom dataset, however the loss went to inf easily or gradient explosion. Hence, for training the tiny dataset, we froze the first two Transformer Block in Figure 2. We fine-tune the high-level feature extractors and corresponding decoder with data augmentation through random resize with ration 0.5-2.0, random flipping, and random cropping. We used a batch size of 2 and epoch of 2k on a single NVIDIA GTX 3090Ti GPU. The learning rate was set to an initial value of 0.00006 and then used a “poly” LR schedule with factor 1.0 by default.

1.3 Integrat Segmentation and Line Detection

With the L-CNN we can get quite great results of our dataset but with much redundant information. We take the detection result and segmentation as input, convert the segmentation as label value and match the line detection results. We eliminate unnecessary detections inside specified class e.g. we only keep lines of board, rulers inside the wall, and remove other lines; we only keep the lines between ceiling and wall or between wall and floor/skirting. Through this pre-definition and coordination matching with segmentation, we can filter unneeded lines and keep the wireframes which show the fundamental structure of a scene.

3. Results Analysis

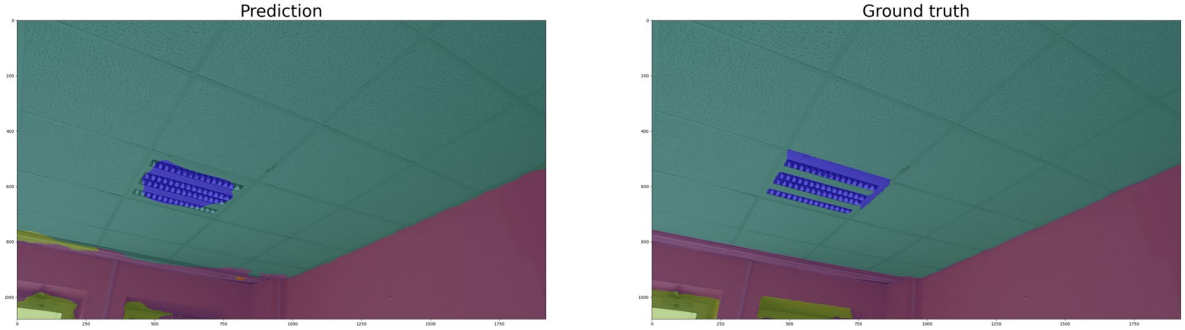
Because the wireframe detection is the pre-step of 3D indoor room reconstruction. For 3D reconstruction we need to detect these fundamental lines which can show the structure of a scene. Hence, keeping wireframes of walls, doors etc. and connection between them are critical.

Figure 3: The wirframe detection

From Figure 3 we can find the L-CNN works pretty well and can detect everything what we need with accuracy higher than 95%. Color of lines means different accuracy. During evaluation, we set accuracy as 95%, 97% and 99%. Because of outstanding performance of accuracy and speed, we do not need to fine-tune this network.

We have tested the Mask R-CNN [18], the performance is acceptable but still has some improvements space. Because of the huge success of applied Transformer in computer vision area, we considered semantic segmentation algorithm based on vision Transformer.

The well-trained semantic segmentation model can meet our most requirements because the training dataset ADE20K contains most scenario of indoor room. However, a few specified classes e.g. socket and skirting are now trained on the SegFormer, the model cannot finish the segmentation correctly. Segmentation is critical to control the final results, hence fin-tuning the SegFormer is necessary and meaningful.



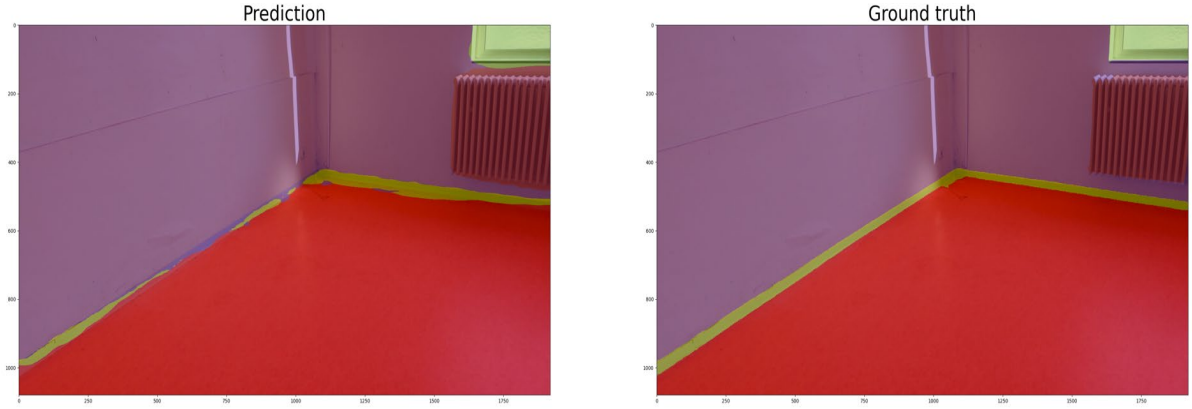
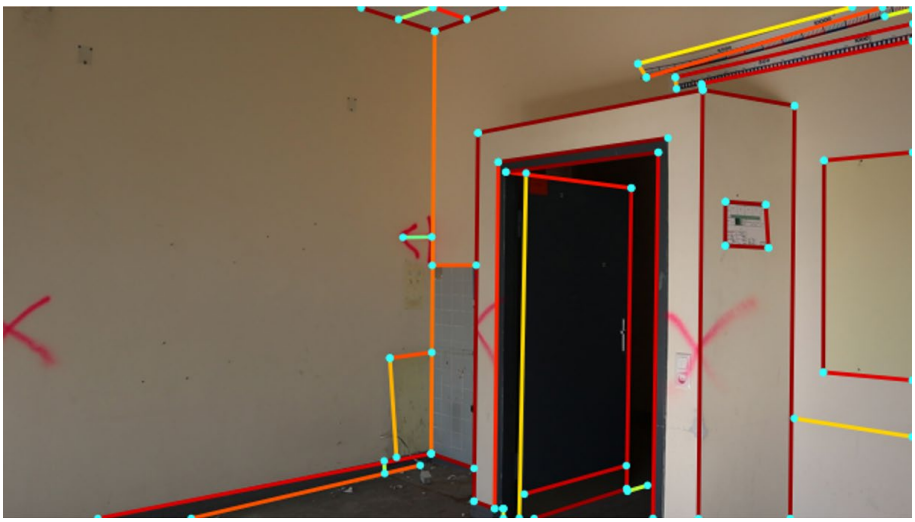
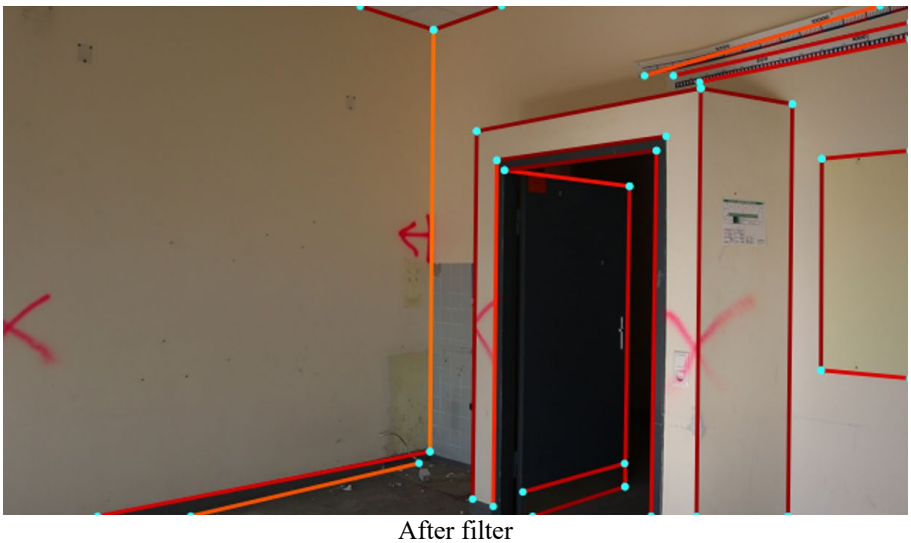


Figure 4: Semantic segmentation on custom dataset

Figure 4 shows two examples with fine-tuned dataset. Prediction means the result from SegFormer, group truth is our manually annotations. From this comparison, we can clearly see the result is quite outstanding. Only tiny area cannot be detected, especially the skirting, since it is really thin and less clear to observe. Next step is to combine both algorithm and control the wireframes.



Before filter



After filter

Figure 5: comparison between with and without using semantic algorithm as controller.

Figure 5 shows clearly the controller removes redundant wireframes and only keep the structural information.

1. Line detections in the ceiling area are completely removed except the boundary with wall.
2. We keep the wireframes of board and rulers inside the wall, others are removed.
3. Lines of boundary between wall and floor are important and not eliminated.

4. Summary

Through the semantic segmentation algorithm as filter, we can realize semantic wireframe detection and keep fundamental information which show the 3D structure of a scene. The semantic wireframe detection algorithm highly depends on the accuracy of segmentation result since, it controls the region of keeping or removing the lines. Hence, it is critical to realize the high accuracy segmentation. Because of less custom dataset, there are still improvement space for fine-tuning process.

The purpose of the project is to reconstruct the indoor room, realize 3D segmentation and detect the length of boundary between objects automatically. Hence, wireframe detection is one of the process of the whole project. We have combined the depth information and the wireframe detection, then we can get the length of detected lines automatically. As a result, manually measurement will not be necessary. Further research will be continued.

Referenzen

[1] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In CVPR, 2018.

[2] Yichao Zhou, Haozhi Qi, Yi Ma. End-to-End Wireframe Parsing. In ICCV, 2019.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In ICLR, 2015.

[5] Dosovitskiy, A., Beyer, L., Kolesnikov, A. I., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In ICLR, 2021.

[6] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. CVPR, 2021.

[7] Enze Xie, WenhaiWang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In NeuIPS, 2021.

[8] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[10] Ross Girshick. Fast R-CNN. In Proceedings of the IEEE international conference on computer vision, 2015.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. In NeuIPS, 2017.

[12] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv, 2021.

[13] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In CVPR, 2018.

[14] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In European conference on computer vision, pages 197–210. Springer, 2008.

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016.
[16] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In CVPR, 2017.
[17] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In CVPR, 2018.
[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. MASk R-CNN. In ICCV, 2017.