Computer Vision Term Project Report

Jiazheng Li[1]
[1] Beijing Institute of Technology

Abstract

This paper presents a novel approach for 3D indoor scene reconstruction using the Manhattan-world assumption. It leverages Multi-Layer Perceptrons (MLPs) to represent scene geometry through signed distance functions, integrating planar constraints in floor and wall regions based on semantic segmentation. The method optimizes both scene geometry and semantics in 3D space, significantly enhancing reconstruction quality, especially in low-textured planar regions. The approach outperforms previous methods in terms of 3D reconstruction quality on ScanNet. The code and supplementary materials are available at https://github.com/foreverlasting1202/cv-term-pj.

1. Method

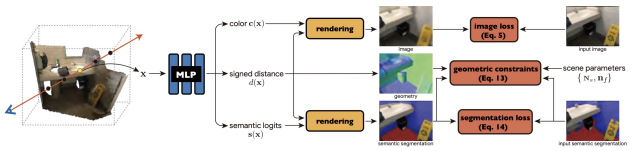

Figure 1. Overview of our method. [2]

1.1. The Manhattan and Atlanta World Assumption

In artificially defined environments, scenes typically have a structured form (for example, the layout of buildings and many indoor scenes like furniture), which can be represented by a set of parallel and orthogonal planes.

The Manhattan world assumption considers that the world is composed of a single horizontal plane and a single vertical plane, and the normals of these planes can easily describe this world.

In contrast, the Atlanta world assumption posits that artificially defined scenes can be composed of horizontal planes (e.g., the ground) and several vertical planes (e.g., buildings and walls), and then the normals of these planes (referred to as the world frame) can abstractly describe the scene.

In short, the difference between the two world assumptions lies in the number of normal vectors in a certain dimension.

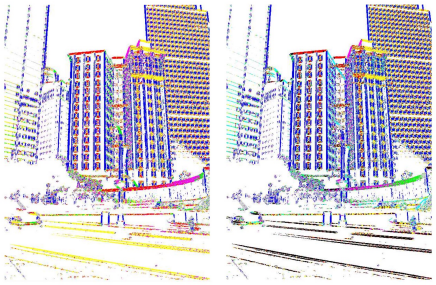

Figure 2. Left is Manhattan and while right is Atlanta. [5].

1.2. The implicit neural representation

Unlike Multi-View Stereo (MVS) methods [9], we model the scene as an implicit neural representation and learn it from images through a differentiable renderer. Inspired by [11], we represent the scene's geometry and appearance using signed distance and color fields. Specifically, for a given 3D point $\mathbf{x}$, the geometric model maps it to a signed distance $d(\mathbf{x})$, defined as follows.

For an input scene $\mathbf{x}$, three implicit fields are constructed:

$$\begin{aligned}&\text{color:}\mathbf{c}(\mathbf{x})\\&\text{signed distance:}d(\mathbf{x})\\&\text{semantic logits:}\mathbf{s}(\mathbf{x})\end{aligned}$$

We can use MLP to acquire them:

$$\begin{aligned}(d(\mathbf{x}),\mathbf{z}(\mathbf{x}))&=F_d(\mathbf{x})\\\mathbf{c}(\mathbf{x})&=F_{\mathbf{c}}(\mathbf{x},\mathbf{v},\mathbf{n}(\mathbf{x}),\mathbf{z}(\mathbf{x}))\\\mathbf{s}(\mathbf{x})&=F_{\mathbf{s}}(\mathbf{x})\end{aligned}$$

Here, $\mathbf{z}(\mathbf{x})$ is the geometry feature, $\mathbf{n}(\mathbf{x})$ is computed by the gradient of $d(\mathbf{x})$, and F is implemented as an MLP network.

1.3. NeRF Rendering

Using NeRF to render SDF, color, and semantics separately.

We utilize volume rendering to learn scene representation networks from images. Specifically, to render an image pixel, we sample N points $\{\mathbf{x_i}\}$ along its camera ray r. We then predict the signed distance and color for each point. To apply volume rendering techniques, we transform the signed distance $d(\mathbf{x})$ into a volume density $\sigma(\mathbf{x})$.

$$\sigma(\mathbf{x}) = \begin{cases} \frac{1}{\beta}(1 - \frac{1}{2}\exp(\frac{d(\mathbf{x})}{\beta})) & d(\mathbf{x}) < 0 \\ \frac{1}{2\beta}\exp(-\frac{d(\mathbf{x})}{\beta}) & d(\mathbf{x}) \geq 0 \end{cases}$$

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{K} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i$$

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{s}_i$$

Here, $\delta_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|$ is the distance between adjacent sampling points, $T_i = \exp(-\sum_{j=1}^{i-1}\sigma_j\delta_j)$ is the accumulated transparency, and β is a learnable parameter.

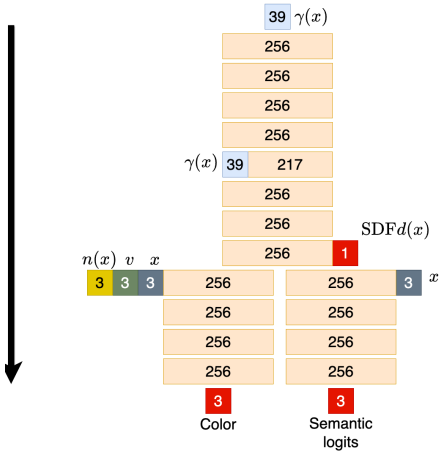

Figure 3. Network architecture.

1.4. Loss Functions

Based on the preceding preparation and analysis, we have designed the following loss function:

Color loss. $\mathcal{L}_{\text{img}} = \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|$. Here, $\mathbf{C}$ is the ground-truth pixel color, and $\mathbb{R}$ is the set of camera rays going through sampled pixels.

Eikonal loss. $\mathcal{L}_E = \sum_{\mathbf{y}\in\mathcal{Y}}(\|\nabla_{\mathbf{y}} d(\mathbf{y})\| - 1)^2$. Here, $\mathcal{Y}$ is the combination of points sampled from random uniform space and surface points for pixels.

Depth loss. $\mathcal{L}_d = \sum_{\mathbf{r}\in\mathcal{D}} \|\hat{\mathbf{D}}(\mathbf{r}) - \mathbf{D}(\mathbf{r})\|$. Here, $\mathcal{D}$ is the set of camera rays going through image pixels.

Semantic Segmentation loss. $\mathcal{L}_s = -\sum_{\mathbf{r}\in\mathcal{R}}\sum_{k\in\{f,w,b\}} p_k(\mathbf{r}) \log \hat{p_k}(\mathbf{r})$. Here, $\hat{p_k}(\mathbf{r})$ is the rendered probability for class k.

Geometry loss. $\mathcal{L}_{\text{joint}} = \sum_{\mathbf{r}\in\mathcal{F}} \hat{p_f}(\mathbf{r})\mathcal{L}_f(\mathbf{r}) + \sum_{\mathbf{r}\in\mathcal{W}} \hat{p_w}(\mathbf{r})\mathcal{L}_w(\mathbf{r})$.

Here, $\mathcal{L}_f(\mathbf{r}) = \|1 - \mathbf{n}(\mathbf{x_r}) \cdot \mathbf{n_f}\|$, $\mathbf{n_f} = (0, 0, 1)$, $\mathcal{L}_w(\mathbf{r}) = \min_{i\in\{-1,0,1\}} f_{\mathbf{n}_w\in\mathcal{N}_w}\|i - \mathbf{n}(\mathbf{x_r}) \cdot \mathbf{n_w}\|$, $\mathbf{N}_w$ is a set of horizontal baseline vectors of a certain size, and f is $\min$ /avg.

To get a certain f, we will use experiments to compare them. $\mathbf{N}_w$ is composed of a baseline vector rotated by several angles.

2. Implementation details

We implement our method with PyTorch and adopt DeepLabV3+ from Detectron2 for implementing 2D semantic segmentation network. The network training is performed on one NVIDIA 3090 GPU.

3. Experiments

Dataset. We perform the experiments on ScanNet (V2) [1]. ScanNet is an RGB-D video dataset that contains 1613 indoor scenes with 2.5 million views. It is annotated with ground-truth camera poses, surface reconstructions, and instance-level semantic segmentations.

Metrics. For 3D reconstruction, we use RGB-D fusion results as ground truth and evaluate our method using 5 standard metrics: accuracy, completeness, precision, recall, and **F-score**. Following the recommendation of [6], we consider the **F-score** as the overall metric.

Baselines. We use five baselines:

1. Classical MVS method: COLMAP [7].
2. NeRF [4].
3. NeuS [8].
4. VolSDF [10].
5. Manhattan [2].

Ablation study.

Figure 4. On the left is the framework, and on the right is the same with added color. It can be observed that the framework is reasonably processed, but the added color is not quite appropriate. A better method is needed to add color.

As can be seen from Table 1, our results perform better than the baseline in terms of Comp and F-score, but are not as effective as MVS in other aspects.

Method	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑
COLMAP	**0.047**	0.235	**0.711**	0.441	0.537
NeRF	0.735	0.177	0.131	0.290	0.176
NeuS	0.179	0.208	0.313	0.275	0.291
VolSDF	0.414	0.120	0.321	0.394	0.346
Manhattan	0.072	0.068	0.621	**0.586**	0.602
Ours(min Method)	0.069	0.049	0.631	0.581	0.579
Ours(avg Method)	0.075	**0.046**	0.642	0.583	**0.611**

Table 1. Average 3D reconstruction metrics on four datas.

The COLMAP algorithm based on MVS considers filtering points with inconsistent forms across multiple views during the fusion stage to achieve higher precision, but this leads to a decrease in recall. NeRF performs extremely poorly, mainly because its volume density lacks sufficient constraint on geometry, resulting in poor reconstruction of complex geometrical structures like toilet bowls. NeuS and VolSDF perform slightly better than NeRF due to SDF's excellent constraint on structural surfaces, but they still struggle with complex structures like toilet bowls. Reconstructions based on world assumptions yield better results for such structures, though there is still room for improvement in terms of precision.

4. Conclusion

In this paper, we adopted a 3D reconstruction method based on the Atlanta world assumption. We expressed a vector group by rotating the baseline vector of the horizontal plane multiple times and used its average value as the geometric loss. From the results, some metrics are better than the baseline, but since the dataset is still not very large, we cannot make a final judgment and need to spend time running more datasets.

Future Works. In the Atlanta world representation, there are actually some effective search algorithms [3] that can be used to update the vectors' conditions. In fact, we believe that a more promising direction for 3D reconstruction is based on GPT. Representing the geometric world entirely with triangles and using GPT to predict the next triangle for reconstructing the model is quite efficient. This approach may become mainstream.

5. Code

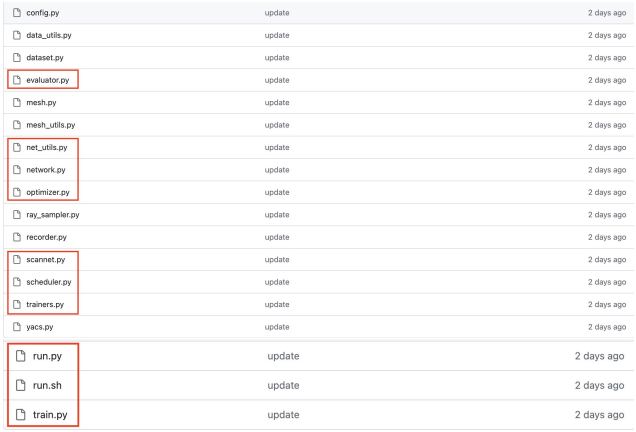

Figure 5. The code has been made publicly available on GitHub. The parts marked with a red circle are the sections I am responsible for.

References

[1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. 2

[2] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption, 2022. 1, 2

[3] Yinlong Liu, Guang Chen, and Alois Knoll. Globally optimal vertical direction estimation in atlanta world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020. 3

[4] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 2

[5] G. Schindler and F. Dellaert. Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004. 1

[6] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video, 2021. 2

[7] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited, 2021. 2

[8] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, 2023. 2

[9] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo, 2018. 1

[10] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces, 2021. 2

[11] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance, 2020. 1