Supplementary Material
Fast Building Instance Proxy Reconstruction for Large Urban Scenes

Jianwei Guo, *Member, IEEE,* Haobo Qin, Yinchang Zhou, Xin Chen, Liangliang Nan,
Hui Huang, *Senior Member, IEEE*

In this document, we detail our 2D instance segmentation neural network, including an explanation of InstFormer and the training loss functions. Besides, we demonstrate the new 2D and 3D urban datasets and present more comparison and reconstruction results on these datasets. Our code and datasets will be released to facilitate future research.

1 2D Building Instance Segmentation

1.1 Network architecture of InstFormer

Fig. 1 summarizes the network architecture of InstFormer that predicts accurate instance masks for buildings at the pixel level. InstFormer adopts a 3-stage cascade structure comprising of three Box branches (*i.e.,* see the *Bounding Box Heads* in Fig. 1). The Box branches in the first two stages are responsible for gradually outputting coarse bounding boxes, and the counterpart in the last stage refines the box predictions and generates instance masks.

InstFormer is mainly composed of three essential components: *Backbone, Neck, and Head*, detailed as follows:

Backbone: The backbone of most current segmentation/detection methods is the feature pyramid (FP) structure based on convolution neural networks (CNNs). Though Vision Transformer (ViT) has shown superior performance in image classification, it performs poorly and has a high computational overhead when directly applied to dense prediction tasks, such as instance segmentation. Inspired by the FP structure of CNNs, we utilize Pyramid ViT [1] as a backbone to extract feature pyramids from images with dense instances and output high-resolution feature maps. The detailed structure of the PVT encoder is shown in Fig. 2. Because buildings in environments can be quite dense and have varying scales, high-resolution feature maps have to be processed, which is what PVT encoders are good at rather than ViT and CNNs. The key difference between PVT encoder and ViT encoder is that the former uses the spatial reduction attention (SRA) layer to replace the multi-head attention (MHA) layer in the latter. In addition, the computational/memory costs of the attention operation in SRA have been greatly reduced compared to those in MHA, which enables PVT encoder to handle larger input feature maps.

Neck: In existing feature pyramid networks (FPN) for segmentation, bi-linear interpolation or deconvolution is mostly used for upsampling. To increase the receptive field to aggregate contextual information, we integrate CARAFE [2] as a lightweight upsampler in the neck to efficiently perform content-aware handling calculations and reduce computational overhead. The detailed up-sampling process of CARAFE is shown in Fig. 1.

Moreover, to improve the capacity of feature expression in the detection branch, we add a dynamic detection head (DyHead) [3] combined with multiple self-attention mechanisms in the last part of the neck so that the model can better carry out spatial perception, scale perception, and task perception. As illustrated in Fig. 1, DyHead is composed of three attention modules, namely π_L, π_S, and π_C, which are concatenated together. Among them, π_L serves as a scale-aware attention function, dynamically fusing features from different scales of the feature pyramid based on semantic importance. Meanwhile, π_S, as a spatial-aware attention module, aims to discover discriminative regions that coexist consistently between spatial positions and feature levels based on the fused features. π_C, serving as a task-aware attention layer, is designed for joint learning and generalization of different object representations. Please refer to [3] for detailed constructions of these three attention modules.

Overall, the role of the neck is to improve the perception capacity of the model to secure more accurate instance masks for the subsequent multi-view instance fusion.

Head: To fully exploit the useful information in FPN, we employ a Generic RoI extractor with non-local building blocks and attention mechanisms to improve segmentation performance. At the same time, we use the Region Proposal Network (RPN) head to locate regions that may contain objects of interest. The RPN head consists of a series of convolutional layers followed by two sibling output layers: one for predicting the objectness score (foreground/background classification) and the other for regressing bounding box coordinates. These output layers enable the RPN to propose regions likely to contain objects of interest, which are subsequently refined and classified by downstream components

- *Jianwei Guo and Yinchang Zhou are with MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China.*
- *Haobo Qin is with University of Chinese Academy of Sciences, Beijing, China, and Shenzhen University, Shenzhen, China.*
- *Xin Chen is with Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China.*
- *Liangliang Nan is with Delft University of Technology, Netherlands.*
- *Hui Huang is the corresponding author (Email: hhzhiyan@gmail.com) with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.*

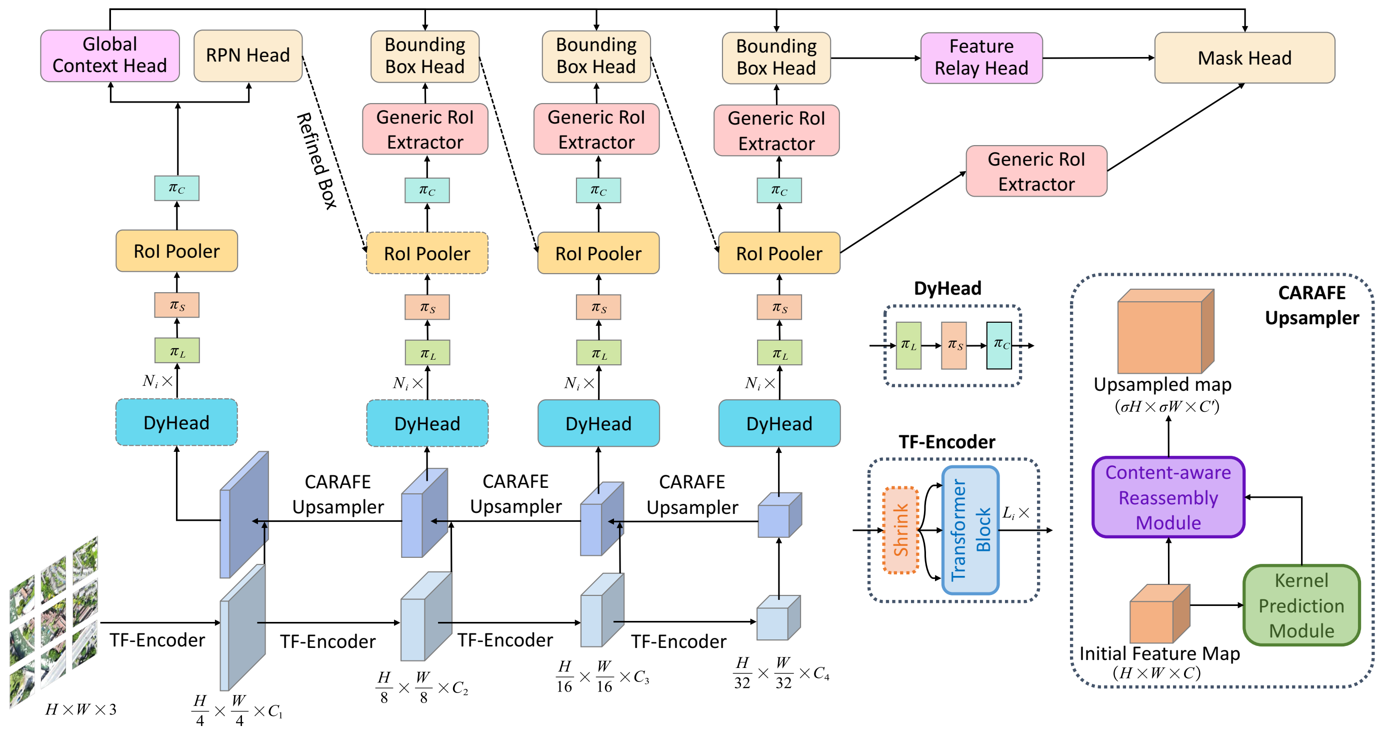


Fig. 1: **InstFormer**: Network architecture for dense and multi-scale building instance segmentation. The InstFormer adopts a hybrid task cascade (HTC) architecture. First, multiple tasks such as detection, mask prediction, and semantic segmentation are combined at each stage to form a joint multi-stage processing pipeline, allowing each stage to benefit from the other tasks. Second, contextual information goes through an extra branch for stuff segmentation, and a directional path is added to allow direct information flow across stages. Overall, the HTC architecture effectively improves the flow of information not only across stages but also between tasks.

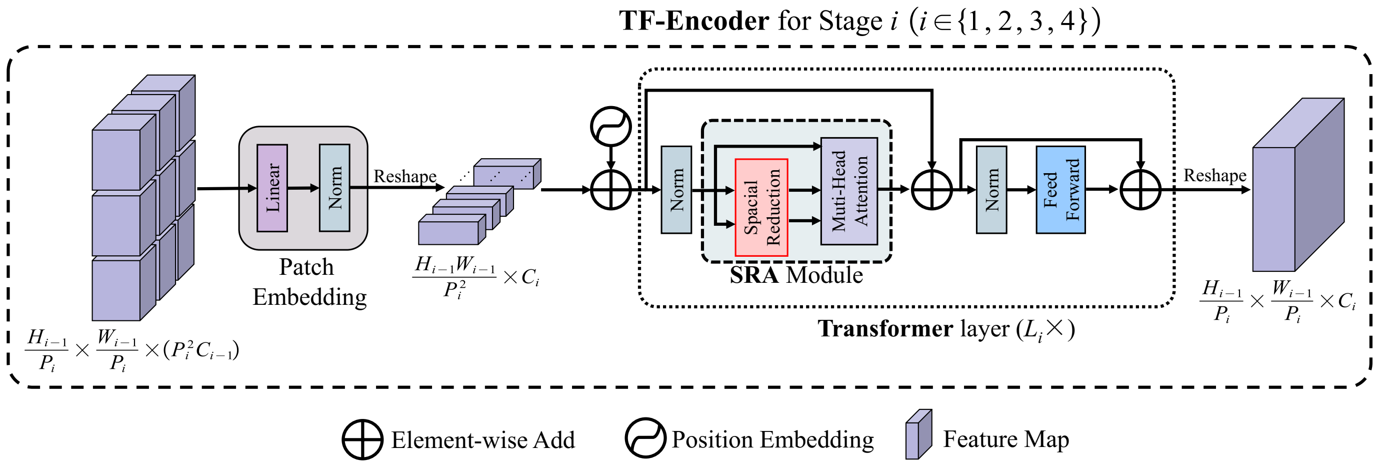
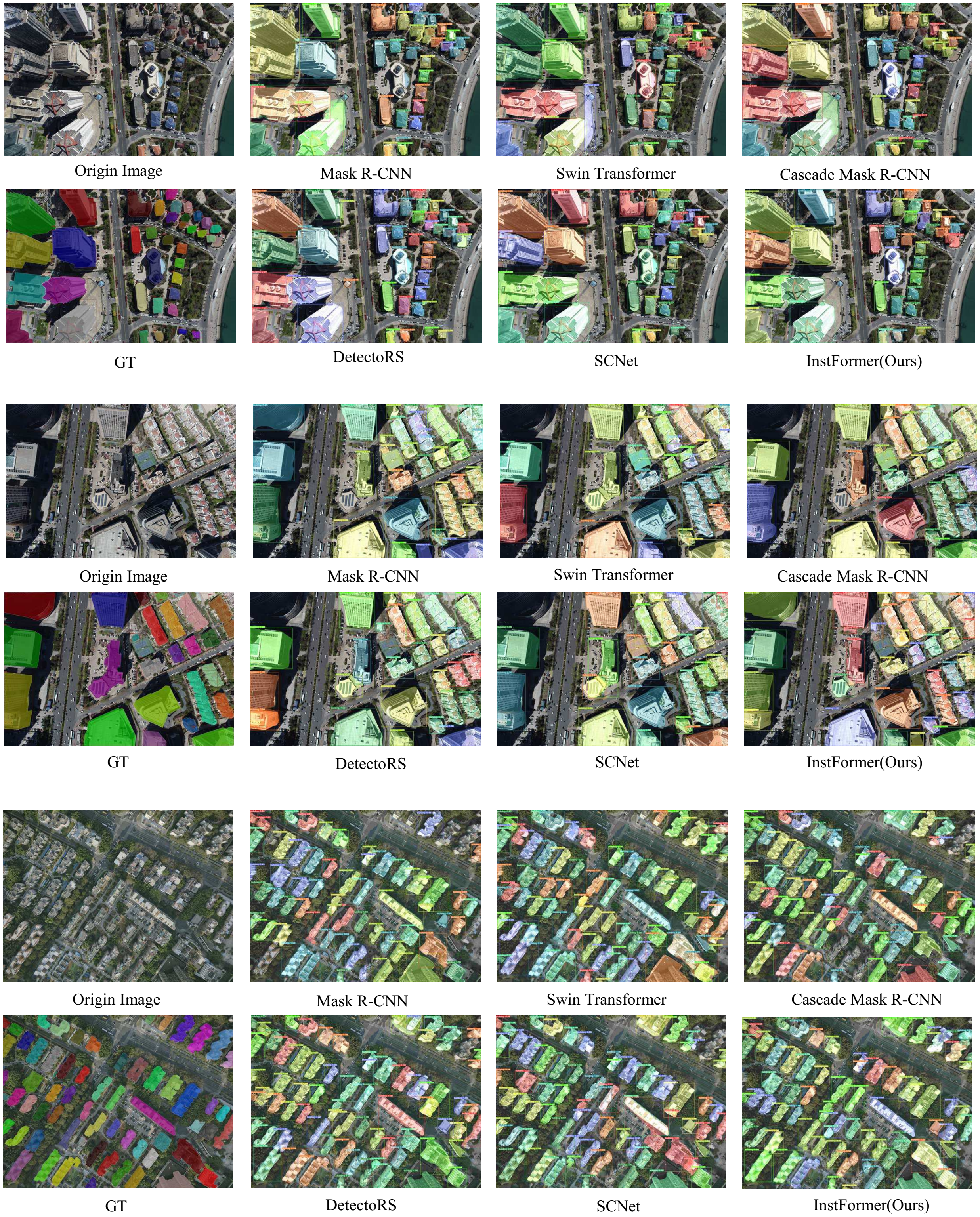

TF-Encoder for Stage i $(i\in\{1,2,3,4\})$
$\frac{H_{i-1}}{P_i}\times\frac{W_{i-1}}{P_i}\times(P_i^2C_{i-1})$
Linear
Norm
Patch Embedding
Reshape
$\frac{H_{i-1}W_{i-1}}{P_i^2}\times C_i$
Norm
Spacial Reduction
Muti-Head Attention
SRA Module
Norm
Feed Forward
Transformer layer $(L_i\times)$
Reshape
$\frac{H_{i-1}}{P_i}\times\frac{W_{i-1}}{P_i}\times C_i$
Element-wise Add
Position Embedding
Feature Map

Fig. 2: **Pyramid Vision Transformer (PVT) Encoder**: The feature map (FM) extraction process of PVT can be divided into 4 stages. The input of stage i is the FM output from the previous stage (*i.e.*, stage $i-1$). The FM of the next stage is output through patch embedding and $L_i\times$ Transformer layers.

of the object detection pipeline. Moreover, we also use the global context head combined with the feature relay head to strengthen the relevance of classification, detection, and segmentation tasks.

Besides, to ensure the consistency of the sample IoU distribution of the model during training and inference, we adopt the interleaved execution [4] between the box branch and the mask branch, where we apply the direct information flow in the mask head. To sum up, the function of the head is to fully utilize the features in FPN and strengthen the correlation between different tasks and improve the generalization capability of the neural network.

Because of the advantages of the three components designed above, our transformer architecture is quite attractive for urban analysis.



Origin Image
Mask R-CNN
Swin Transformer
Cascade Mask R-CNN
GT
DetectoRS
SCNet
InstFormer(Ours)

Origin Image
Mask R-CNN
Swin Transformer
Cascade Mask R-CNN
GT
DetectoRS
SCNet
InstFormer(Ours)

Origin Image
Mask R-CNN
Swin Transformer
Cascade Mask R-CNN
GT
DetectoRS
SCNet
InstFormer(Ours)

Fig. 3: Visual comparison of the results of different 2D instance segmentation methods on our validation dataset.

1.2 Loss functions

The proposed InstFormer can be trained in an end-to-end manner using multi-task loss as follows:

$$\mathcal{L} = \sum_{t=1}^{3} \alpha_t \left(\mathcal{L}_t^{\text{cls}} + \mathcal{L}_t^{\text{reg}} \right) + \beta \mathcal{L}^{\text{mask}} + \gamma \mathcal{L}^{\text{glbctx}}. \quad (1)$$

Since we mainly focus on two categories (*i.e.*, buildings and background), the $\mathcal{L}_t^{\text{cls}}$ used for the binary classification adopts the cross-entropy (CE) loss. To make the Bounding Box more accurate, we use the CIoU [5] as the regression loss function $\mathcal{L}_t^{\text{reg}}$. The $\mathcal{L}^{\text{mask}}$ is the cross-entropy loss used to calibrate the instance mask output by the mask head

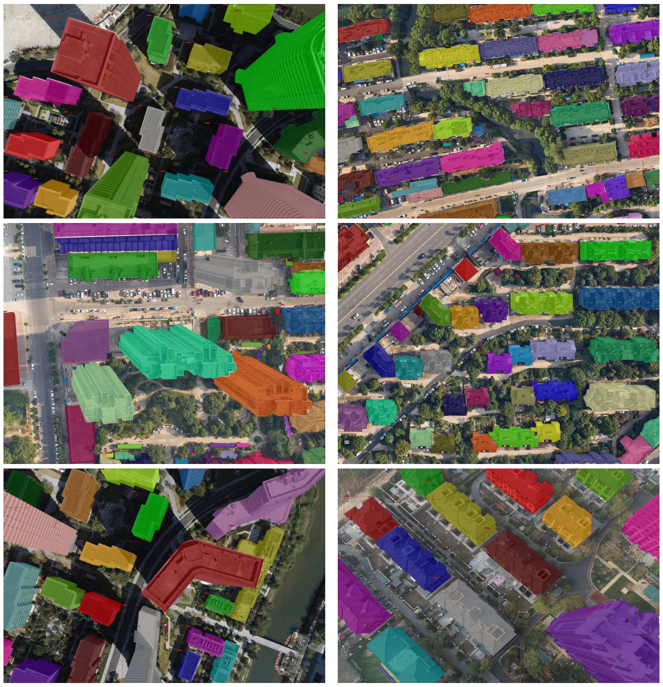


Fig. 4: Some examples of the annotated images, randomly chosen from our building instance segmentation dataset.

module. We also utilize the loss term $\mathcal{L}^{\mathrm{glbctx}}$ in SCNet [6] to obtain better global contextual features, which eventually improves the accuracy in instance fusion. The $\mathcal{L}^{\mathrm{glbctx}}$ is also implemented with the binary cross-entropy loss. In Eq. 1, the hyperparameter vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]$ are the weights of classification and regression losses corresponding to each stage. The hyperparameter β is the weight of the mask loss. Since the instance mask is the output after three stages of adjustment, it makes $\beta := \sum_{t=1}^{3} \alpha_t$ more reasonable, and [6] have pointed out that this setting of β can maintain the consistency of IoU distribution between training and inference samples, thus reducing over-fitting. The hyperparameter γ corresponds to the loss weight of the global contextual feature.

As $\mathcal{L}^{\mathrm{cls}}$, $\mathcal{L}^{\mathrm{glbctx}}$, and $\mathcal{L}^{\mathrm{mask}}$ are all computed as a binary CE loss, they share the same expression as:

$$\textit{BCE Loss} = -\frac{1}{N}\left[y^{(i)}\log\left(p_i\right) + \left(1 - y^{(i)}\right)\log\left(1 - p_i\right)\right], \tag{2}$$

where p_i denotes the prediction score of the i-th sample. $y^{(i)}$ is the indicator variable, *i.e.*,

$$y^{(i)} = \begin{cases} 1, \mathrm{Label}_i \ \textit{is positive} \\ 0, \mathrm{Label}_i \ \textit{is negative} \end{cases} \tag{3}$$

Moreover, the expression of regression loss $\mathcal{L}^{\mathrm{reg}}$ is:

$$\mathcal{L}^{\mathrm{reg}} = 1 - \mathrm{IoU} + \frac{d^2\left(\boldsymbol{b}, \boldsymbol{b}^{\mathrm{gt}}\right)}{c^2} + \alpha v, \ \ \mathrm{IoU} = \frac{\left|B \cap B^{\mathrm{gt}}\right|}{\left|B \cup B^{\mathrm{gt}}\right|}, \tag{4}$$

$$\alpha = \frac{v}{1 - \mathrm{IoU} + v}, \ \ v = \frac{4}{\pi^2}\left(\arctan\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}} - \arctan\frac{w}{h}\right)^2, \tag{5}$$

TABLE 1: Density comparison between SfM sparse point cloud and mainstream point cloud datasets.

Datasets	Surface Density		Volume Density	
	Mean	Std	Mean	Std
ScanNet [7]	81122.09	10049.15	6731752.20	833908.68
S3DIS [8]	11931.49	4291.97	656057.25	235995.89
STPLS3D [9]	8.31	1.56	13.30	2.51
SFM Sparse	0.88	0.54	0.20	0.12

where $d^2\left(\boldsymbol{b}, \boldsymbol{b}^{\mathrm{gt}}\right)$ represents the square Euclidean distance between the centers of a predicted box B and ground truth box B^{gt}. c denotes the diagonal length of the minimum common circumscribed rectangle of B and B^{gt}, and IoU represents the area ratio of the intersection and union of B and B^{gt}. The purpose of the penalty term αv is to keep the aspect ratios of B and B^{gt} as consistent as possible, where (w, h) and $(w^{\mathrm{gt}}, h^{\mathrm{gt}})$ denote the frame sizes of B and B^{gt}, respectively.

2 Datasets

2.1 Instance segmentation dataset

For the training and evaluation of *InstFormer*, we have created a new dataset that consists of 720 nadir images from four cities captured with varying flight altitudes, and all building instances in these images have been manually annotated by eight students of computer science, using the annotation tool of LabelMe [10]. Fig. 4 shows a few annotated images from the building instance segmentation dataset. Fig. 3 shows the visual comparison of different 2D instance segmentation methods on our proposed validation dataset. It reveals that our InstFormer has better capability to localize the buildings accurately and can generate more complete instance masks.

2.2 3D synthetic benchmark dataset

To quantitatively evaluate the effect of the proxy geometry on the final reconstruction, we introduce three new customized synthetic urban scenes. Compared to the synthetic scenes proposed by previous work [11], [12], [13], [14], our dataset contains a larger number of buildings with different building styles and diverse distribution densities. In addition, we also generate rich ground textures and a variety of ground objects, such as trees, streetlights, garbage cans, benches, etc. Fig. 5 demonstrates our newly built virtual scenes with natural-looking colored textures and detailed close-ups of the underlying geometry.

2.3 Comparison between SfM sparse point cloud and dense point cloud datasets

As for 3D instance segmentation, it is difficult to extract sufficient point features from sparse point clouds, making direct 3D instance segmentation a challenging task. To show the characteristics of SFM sparse points, we compared the density of SfM points with several public popular point cloud datasets. Table 1 shows the quantitative analysis. We use the software of CloudCompare to estimate the density by counting for each point the number of neighbors*.

*. https://cloudcompare.org/doc/wiki/index.php?title=Density

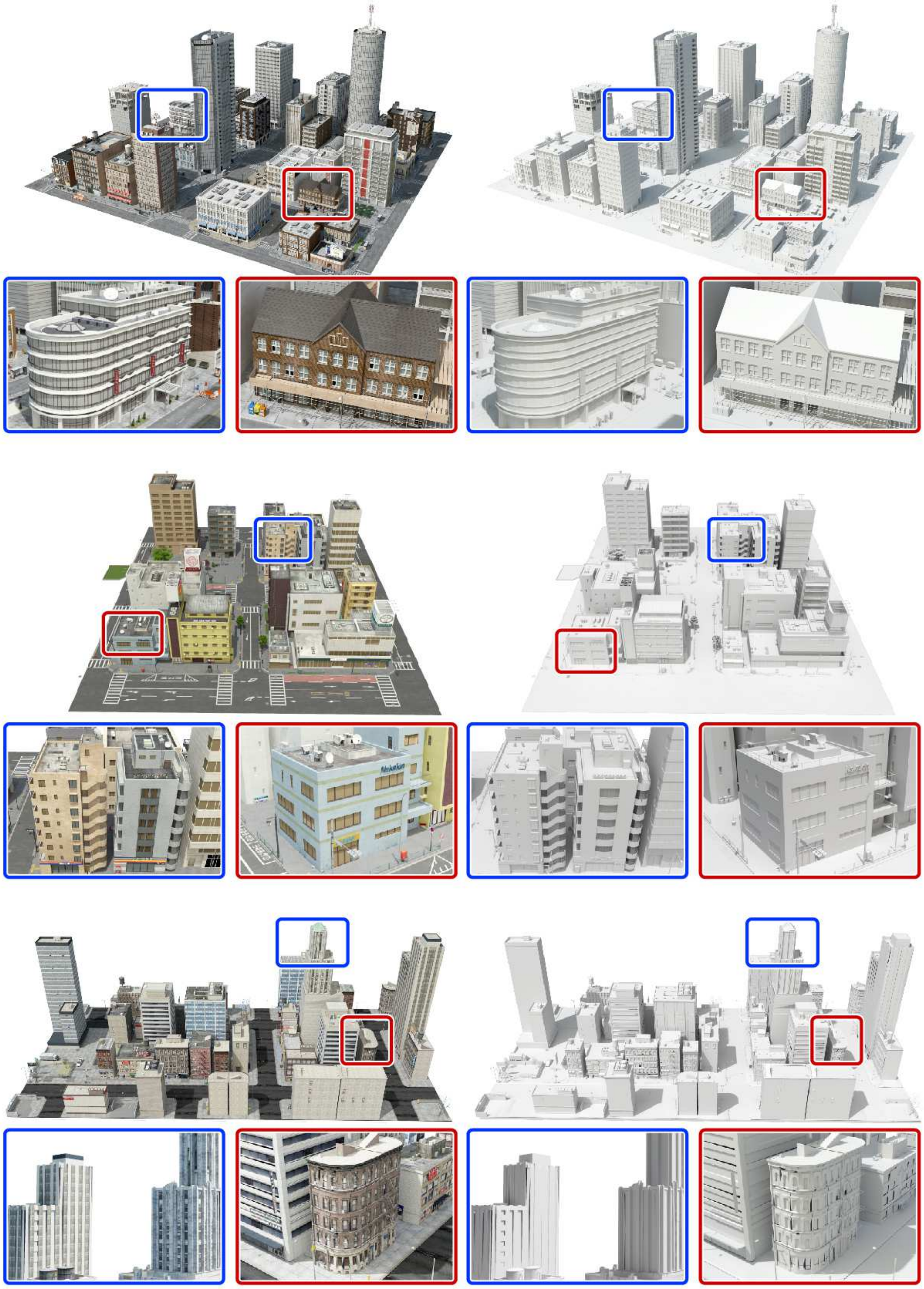


Fig. 5: The three newly built virtual scenes used for quantitative evaluation in our work. Top: *AK-1*; Middle: *JPN-1*; Bottom: *CT-1*. The left column shows the synthetic scenes with texture, and the right column shows the corresponding scenes without texture (to better reveal their geometry).

Specifically, we compute the surface density (the number of neighbors divided by the neighborhood surface) and the volume density (the number of neighbors divided by the neighborhood surface).

The inputs to previous learning-based methods are laser scanning (*e.g.,* ScanNet with an average volume density of 6731752) or MVS dense points (*e.g.,* STPLS3D [9] with an average volume density of 13.30). Not much work tried directly using the sparse data (with an average volume density of 0.20) for instance segmentation. Compared to sparse 3D points, 2D nadir images provide more useful information, especially the building roofs that have good

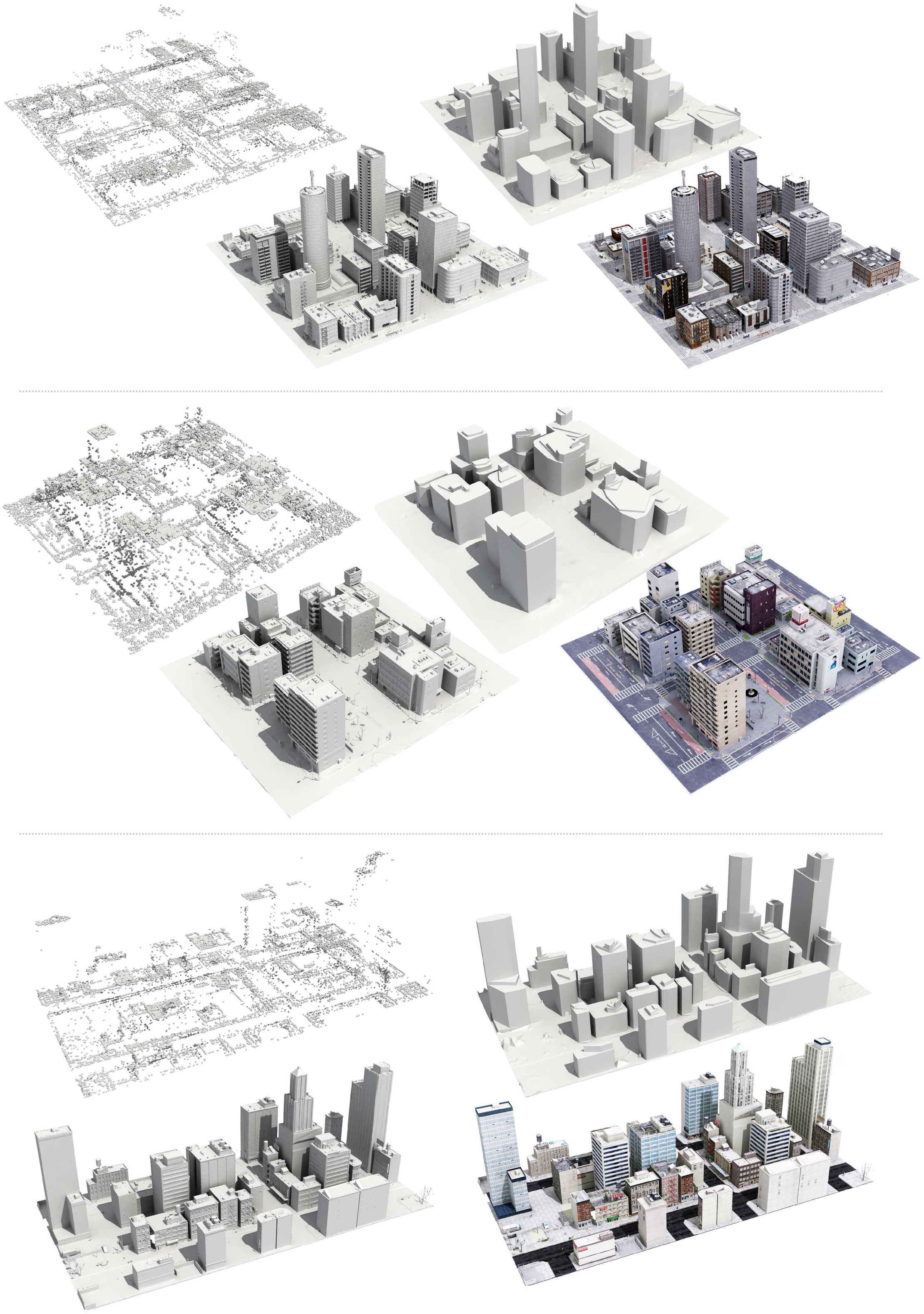


Fig. 6: Our proxy and final reconstruction results on the virtual scenes. For each scene, the SfM point cloud, proxy, final model without texture, and final model with texture are demonstrated.

visibility in aerial images. Thus, we adopt a voting-based instance fusion mechanism to effectively overcome sparsity and incompleteness in sparse points.

3 Additional results

For layer-based proxy reconstruction, in Fig. 6 and Fig. 7 we demonstrate our generated proxies of virtual and real scenes, as well as corresponding final reconstruction results.

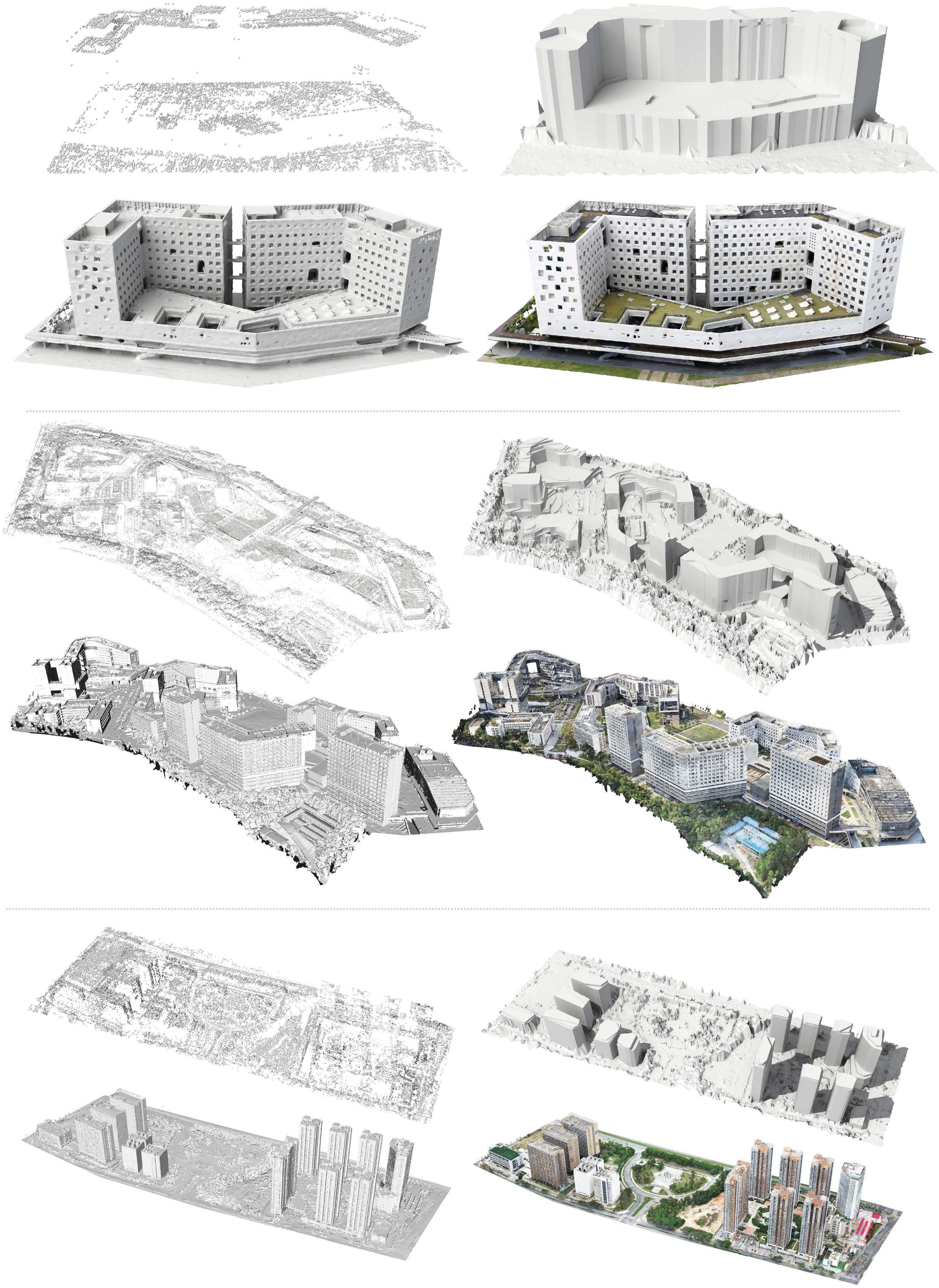


Fig. 7: Our proxy and final reconstruction results on real scenes. For each scene, the SfM point cloud, proxy, final model without texture, and final model with texture are demonstrated.

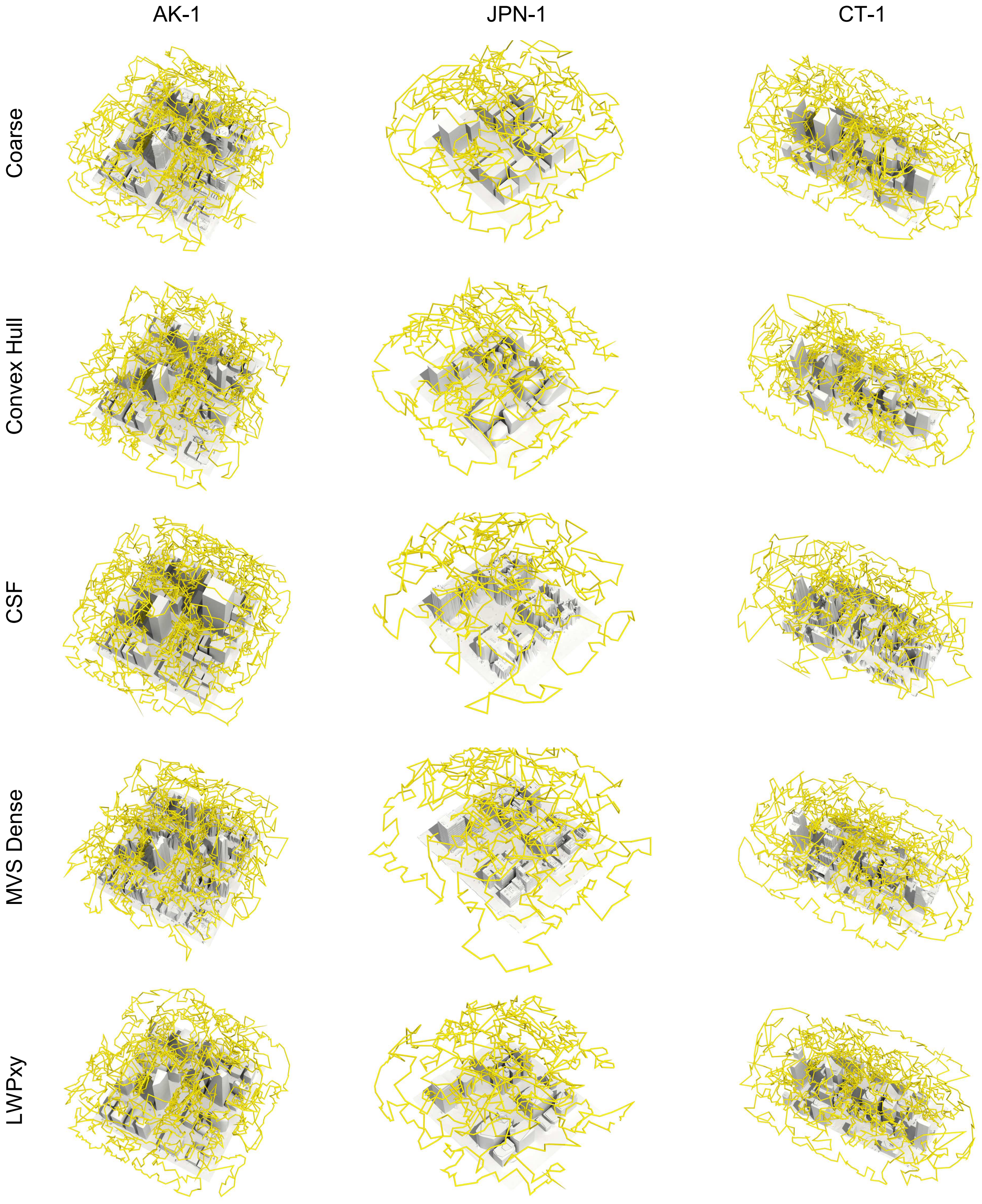


Fig. 8: In the virtual scene evaluation, we show the aerial path planning results based on different proxy models.

Fig. 8 presents the visualization results of path planning [14] for different proxy models using virtual scenes. In Fig. 9, Fig. 10, and Fig. 11, we present the visual comparisons on two virtual scenes (*i.e., JPN-1* and *CT-1*) and two real scenes of *SI-PARK* and *Polytech*.

Acknowledgments

This work was supported by NSFC (U2001206, U21B2023, 62172416, U22B2034), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), DEGP Innovation Team Program (2022KCXTD025), Shenzhen Science and Technology Program (KQTD20210811090044003, RCJC 20200714114435012, JCYJ20210324120213036), CAS Youth Innovation Promotion Association (2022131), and Scientific Development Funds of Shenzhen University. Jianwei Guo and Haobo Qin are with equal contributions.

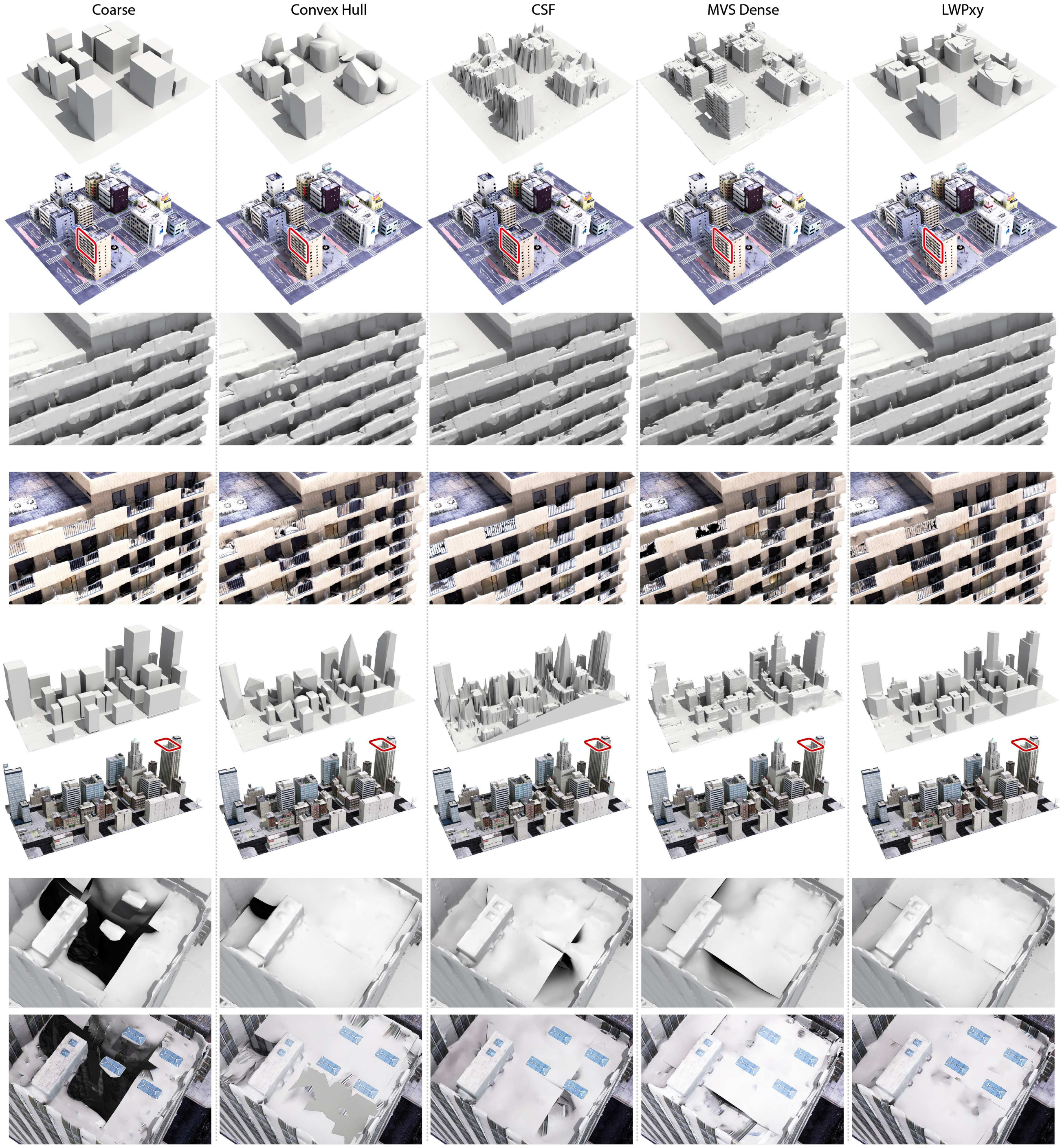


Fig. 9: Visual comparison on two virtual scenes (*i.e., JPN-1* and *CT-1*), for evaluating the effect of different proxy generation methods on the quality of the final reconstruction.

References

[1] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 568–578.

[2] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "Carafe: Content-aware reassembly of features," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3007–3016.

[3] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7373–7382.

[4] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Hybrid task cascade for instance segmentation," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4969–4978.

[5] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 12 993–13 000.

[6] T. Vu, K. Haeyong, and C. D. Yoo, "Scnet: Training inference sample consistency for instance segmentation," in *AAAI Conference on Artificial Intelligence*, 2021, pp. 2701–2709.

[7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of

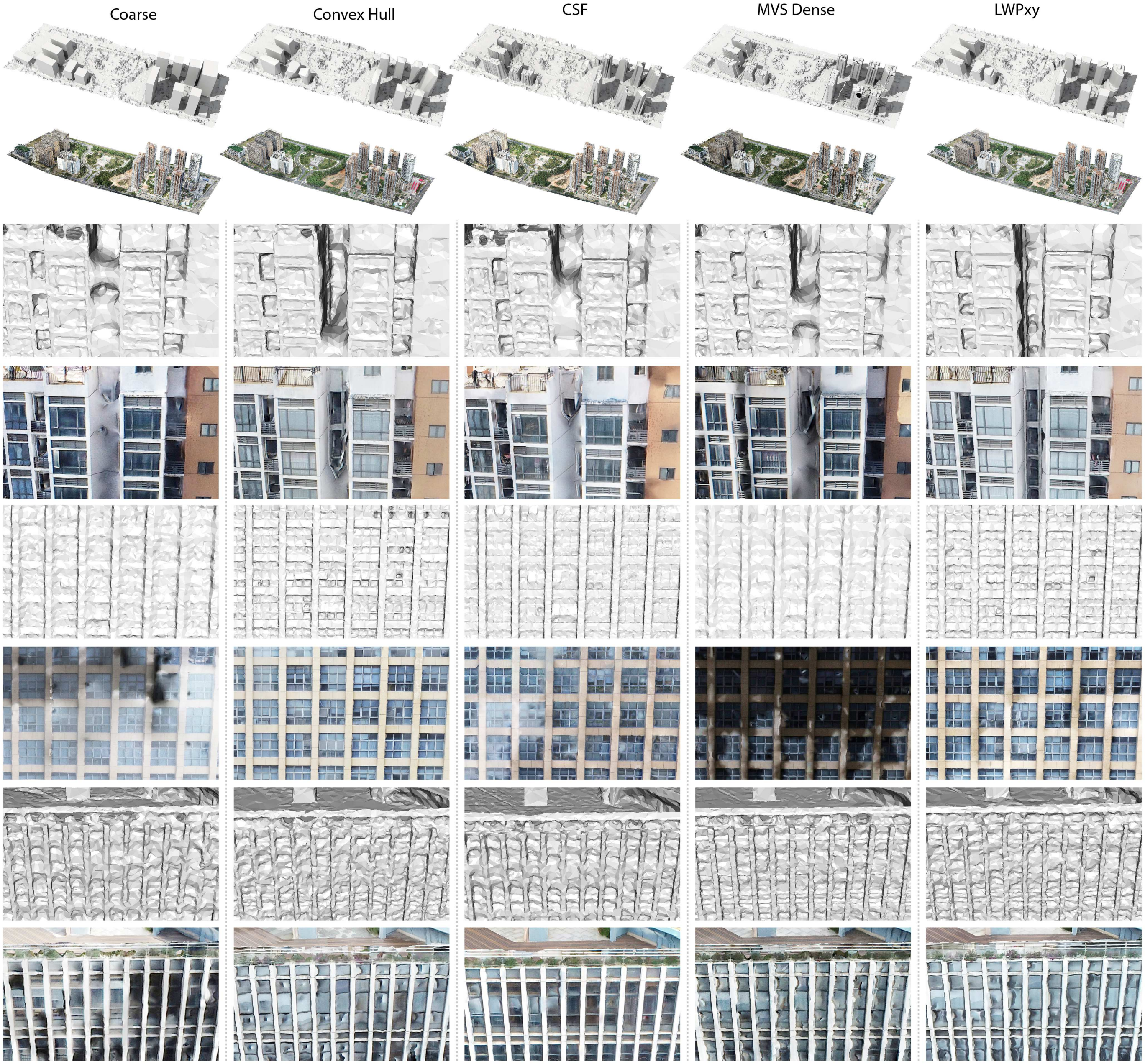


Fig. 10: Visual comparison on the real scene *SI-PARK*, which demonstrates the effect of different proxy generation methods on the quality of the final reconstruction.

indoor scenes," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5828–5839.

[8] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1534–1543.

[9] M. Chen, Q. Hu, Z. Yu, H. THOMAS, A. Feng, Y. Hou, K. McCullough, F. Ren, and L. Soibelman, "Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset," in *33rd British Machine Vision Conference BMVC*, 2022.

[10] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 157–173, 2008.

[11] B. Hepp, M. Nießner, and O. Hilliges, "Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Trans. Graph.*, vol. 38, no. 1, pp. 4:1–4:17, 2018.

[12] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, "Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 37, no. 6, pp. 183:1–183:15, 2018.

[13] T. Koch, M. Körner, and F. Fraundorfer, "Automatic and semantically-aware 3d uav flight planning for image-based 3d reconstruction," *Remote Sensing*, vol. 11, no. 13, p. 1550, 2019.

[14] X. Zhou, K. Xie, K. Huang, Y. Liu, Y. Zhou, M. Gong, and H. Huang, "Offsite aerial path planning for efficient urban scene reconstruction," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 39, no. 6, pp. 192:1–192:16, 2020.

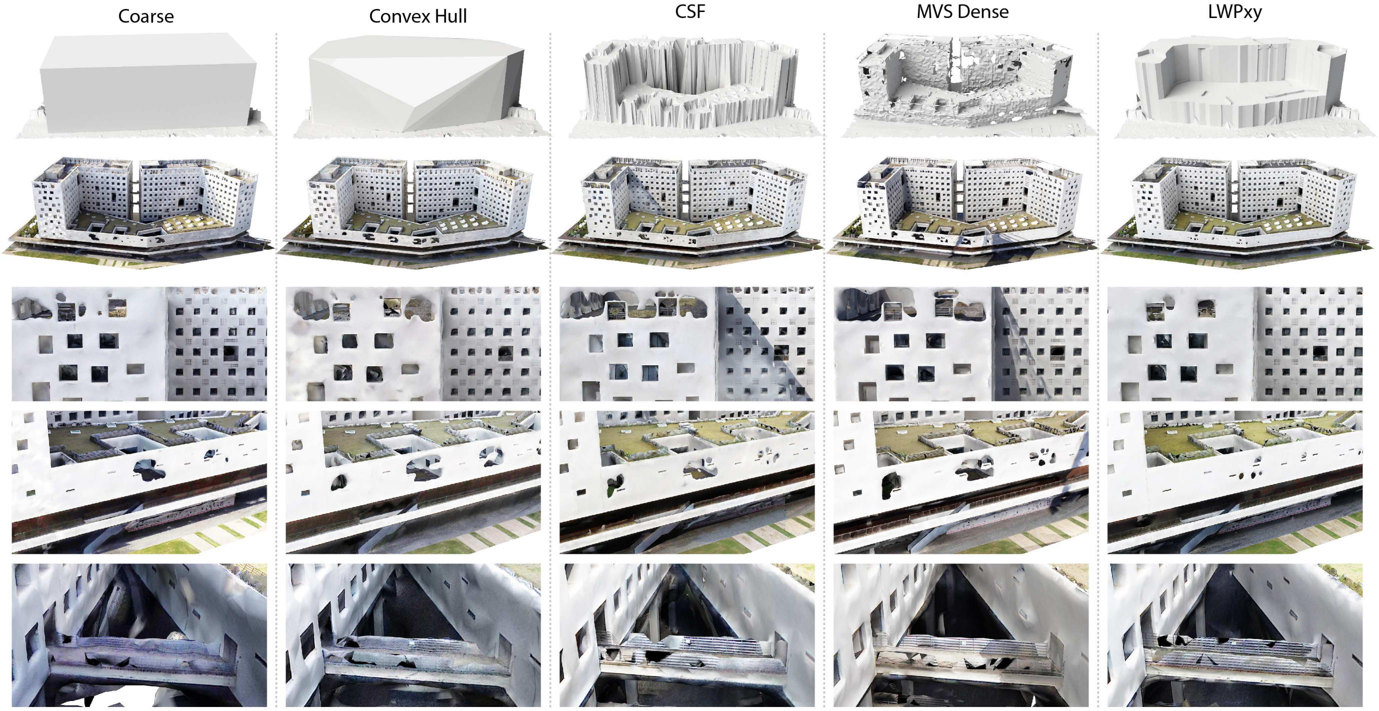


Fig. 11: Visual comparison on the real scene *Polytech*, which demonstrates the effect of different proxy generation methods on the quality of the final reconstruction.