3D Instance Segmentation of MVS Buildings

Jiazhou Chen, Yanghui Xu, Shufang Lu, Ronghua Liang*, and Liangliang Nan

***Abstract*—We present a novel 3D instance segmentation framework for Multi-View Stereo (MVS) buildings in urban scenes. Unlike existing works focusing on semantic segmentation of urban scenes, the emphasis of this work lies in detecting and segmenting 3D building instances even if they are attached and embedded in a large and imprecise 3D surface model. Multi-view RGB images are first enhanced to RGBH images by adding a heightmap and are segmented to obtain all roof instances using a fine-tuned 2D instance segmentation neural network. Instance masks from different multi-view images are then clustered into global masks. Our mask clustering accounts for spatial occlusion and overlapping, which can eliminate segmentation ambiguities among multi-view images. Based on these global masks, 3D roof instances are segmented out by mask back-projections and extended to the entire building instances through a Markov random field optimization. A new dataset that contains instance-level annotation for both 3D urban scenes (roofs and buildings) and drone images (roofs) is provided. To the best of our knowledge, it is the first outdoor dataset dedicated for 3D instance segmentation with much more annotations of attached 3D buildings than existing datasets[1]. Quantitative evaluations and ablation studies have shown the effectiveness of all major steps and the advantages of our multi-view framework over the orthophoto-based method.**

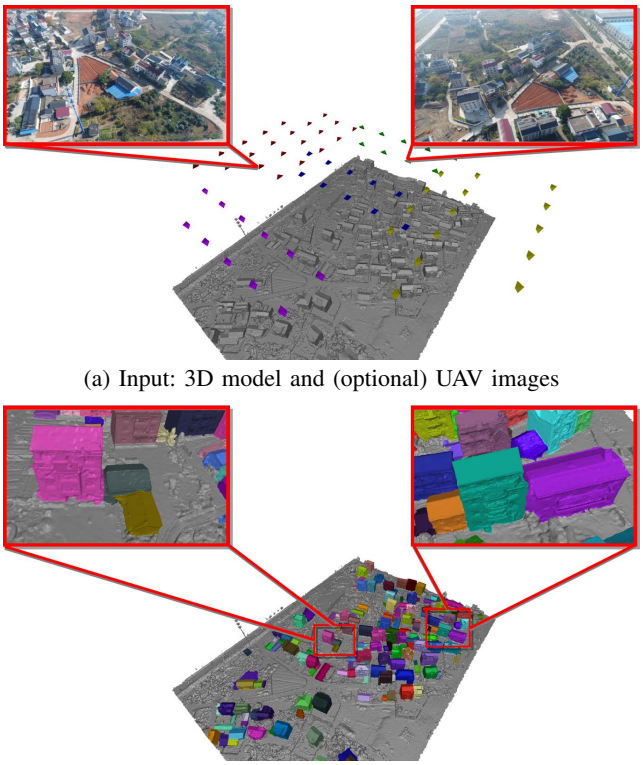
***Index Terms*—Instance segmentation, dataset, 3D urban scene, multi-view clustering.**

I. Introduction

In recent decades, the Multi-View Stereo (MVS) technique has been widely used in the GIS domain. Multi-view images are captured by Unmanned Aerial Vehicle (UAV) and used to automatically reconstruct dense 3D mesh models of large urban scenes [1], [2]. The reconstructed 3D mesh models provide a visually pleasing representation of urban scenes. However, due to the lack of semantic information, they can hardly be used directly in various real-world applications, such as urban planning, simulation, and solar potential estimation.

Buildings are the most important part of a city, its segmentation is the core of the semantic analysis of urban scenes. Rather than semantic segmentation, we focus on the instance segmentation of buildings, as it separates different building instances, even if they are attached. Thus, our goal is to segment all building instances in a large 3D urban scene precisely and automatically, as shown in Fig.1.

Recent advances in deep learning have achieved great success in image instance segmentation, but applying these techniques to 3D mesh models is still challenging and has not been sufficiently explored, especially for 3D buildings in large-scale urban scenes. Feature extraction of 3D data is the key to applying deep learning to the 3D model segmentation. Volumetric-based methods [3], [4], and point cloud-based methods [5], [6] are limited by memory and computing power, thus are mainly used for the segmentation of relatively small indoor scenes [7], [8], [9], [10], [11], [12]. Besides, the lack of annotated 3D instance segmentation dataset for outdoor scenes also hinders the application of deep learning on instance segmentation of 3D urban scenes.



(a) Input: 3D model and (optional) UAV images

(b) Output: 3D building instances

Fig. 1. Instance segmentation of 3D buildings in a large urban scene. The pyramids above the 3D scene model in (a) indicate the position and orientation of the cameras.

Instead of directly segmenting 3D models, segmenting images first and projecting them to the 3D models is a potential alternative, as it can utilize powerful neural networks for image segmentation. Orthophoto maps could be the first candidate, due to their unified projection directions. However, buildings in orthophoto maps have severe self-occlusion, e.g. walls cannot be seen. Thus, its segmentation inaccuracy will be conducted to the 3D models during the 2D-3D projection. For this sake, we employ a multi-view 3D segmentation framework in this paper. It first employs existing learning-based 2D instance segmentation to segment roofs in drone images, back-projects roof instance masks to the 3D scenes considering the

J. Chen, Y. Xu, S. Lu and R. Liang are with the School of Computer Science and Technology, Zhejiang University of Technology, China

L. Nan is with the Deft University of Technology, The Netherlands

* Corresponding author

[1]The datasets are available at https://californiachen.github.io/datasets/Instance-Building

spatial occlusion, and finally constructs Markov random fields to segment out all buildings from 3D scenes.

Such a multi-view framework for 3D semantic segmentation tasks is straightforward, and a winner-take-all mechanism works well for most semantic segmentation tasks. However, a multi-view framework for 3D instance segmentation is much more challenging, as instances have to keep separating even if they are spatially connected. Projecting instance masks back to the 3D scene without correspondences will lead to instance ambiguities, as 2D instance masks from different views may be inconsistent. Buildings in multi-view images are often partially occluded by attached other buildings or tall trees, which increases ambiguities in mask correspondences.

In this paper, we propose a novel instance mask clustering method to build mask correspondences among multi-view UAV images. It solves the issue of instance ambiguities robustly, making our multi-view 3D instance segmentation method outperform the orthophoto-based method. To improve segmentation accuracy for diversely distributed buildings, we enhance multi-view RGB images to RGBH images by adding an extra channel encoding the height information. To ease the spatial occlusion challenge, we perform instance segmentation of roofs instead of entire buildings, since roofs have higher visibility than other parts of buildings.

Though our method incorporates existing image instance segmentation techniques, it includes the following core contributions:

- A multi-view instance segmentation framework that segments 3D buildings in large urban scenes efficiently and precisely;
- An occlusion-aware clustering method for instance masks, which robustly eliminates ambiguities in mask correspondences among multi-view images;
- A benchmark dataset *InstanceBuilding* for instance segmentation evaluation of 3D buildings in large urban scenes, which consists of pixel-level instance annotation for both UAV images and 3D urban models.

II. Related Work

There is a large volume of research in instance segmentation for various data sources. In this section, we review the existing work of instance segmentation for common images, aerial images, and 3D data.

A. Common image instance segmentation

Existing instance segmentation methods for natural images can be classified into two categories: object detection-based approaches and metric learning-based ones.

Object-detection based approaches. Object detection-based approaches work in a top-down manner and highly depend on object detection or proposal. R-CNN first introduces CNN in the field of object detection [13]. To improve computational efficiency, Fast R-CNN proposes an improved SPP (Spatial Pyramid Pooling) [14] structure named RoI pooling [15], and Faster R-CNN uses region proposal networks instead of selective searching to extract object candidates, which forms an efficient end-to-end network for object detection [16].

Based on Faster R-CNN [16], Mask R-CNN combines with FPN [17] to detect objects with different sizes and uses RoIAlign instead of RoI pooling to form a simple, flexible, and effective instance segmentation network [18]. Recently, mask scoring R-CNN uses mask scores to improve the category scores used in Mask R-CNN [19]. PANet uses a bottom-up annotation structure to shorten the information path and enhances the feature pyramid with accurate localization signals existing at low levels [20]. HTC combines detection and segmentation with a multi-task and multi-stage hybrid cascade structure [21]. Swin Transformer [22] proposes a hierarchical Transformer whose representation is computed with shifted windows. Its hierarchical architecture has flexibility at various scales and linear computational complexity concerning image size, which helps it to achieve outstanding segmentation performance.

Metric learning-based approaches. Many other dense instance segmentation methods are based on metric learning [23]. These methods work in a bottom-up manner, generate embedding features [24], [25] for each pixel and use post-processing methods such as clustering [26], [27] or graph theory [28] to classify these pixels. Inspired by FCIS [29] and YOLACT [30], BlendMask uses a blender module to merge top-level coarse instance information with lower-level fine granularities [31].

B. Aerial image segmentation

In the last decade, instance segmentation methods for aerial images of urban scenes have also been proposed because of the wide applications of aerial images. Montoya et al. use an α-shape algorithm to calculate the boundary polygons of building objects, which are further optimized by CRF [32]. By combining the CNN backbone with FPN and RNN, Li et al. propose an end-to-end deep neural network to predict polygon outlines of buildings and road topology maps [33]. Conv MPN uses GNN (graph neural network) [34] to reconstruct the building plan from a single image [35]. DARNet employs a polar representation of contours to predict contours that are free of self-intersection and a loss function consisting of a data term, a curvature term, and a balloon term, which not only encourages the predicted contours to match ground truth building boundaries but also prefers low-curvature solutions [36].

Besides instance segmentation, many semantic segmentation methods for aerial images have been proposed recently in the remote sensing domain. They are also referred to as remote sensing image classification. Besides single-modality images, researchers in the remote sensing domain are also interested in employing deep learning techniques in the pixel-level classification of multi-modality images, including multispectral ones and hyperspectral ones, which are proven to overcome the challenge of information diversity [37]. For instance, Hong et al. introduce graph convolutional networks into hyperspectral image classification in a minibatch fashion [38], and also propose a new transform-based network that learns locally spectral representations from multiple neighboring bands instead of single bands [39]. Since multispectral and hyperspectral images require expensive and heavy spectrometers to acquire, RGB images are more common for UAVs. In this

paper, height maps are automatically generated and added to corresponding RGB images respectively. They can not provide as rich information as hyperspectral images, but this geometric information is a very important supplement that can significantly improve segmentation accuracy.

C. 3D instance segmentation

Unlike images that inherently have a grid structure, the vertices and faces in discrete surfaces (i.e., 3D meshes) do not have regular spatial structures to be directly convoluted. Volumetric methods ease this issue by using a 3D grid representation, which is notoriously expensive in terms of computational efficiency and memory consumption [4], [12], [40], [41], [42], [43].

Various strategies have been proposed to address the memory issue of volumetric methods. For example, OctNet uses an octree structure to avoid unnecessary cells [3], thus reducing memory consumption. PointNet uses T-net and max-pooling to achieve rotation invariance and the capability of handling unordered 3D point clouds. It fuses both local and global features, making it an efficient and effective feature extractor for point cloud data [5]. Through point grouping and multi-level feature extraction, PointNet++ can better extract discriminative features for point clouds with uneven density [6].

Based on features extracted by PointNet, PointNet++, and PointCNN [44], many 3D instance segmentation methods for point clouds have also been proposed. SGPN predicts the instances by learning the similarity matrix between point clouds. However, the size of its similarity matrix tends to explode as the number of points increases [8]. GSPN extends the structure of Mask R-CNN (that was originally developed for images) to process 3D data [7]. JSNet [45] and ASIS [10] both learn the instance embedding space and combine semantic features and instance features of the point clouds to jointly improve the accuracy of semantic segmentation and instance segmentation.

Though great progress has been made for instance segmentation of indoor scenes. However, existing methods are designed for processing point cloud data. It is still a challenge to extend these methods to outdoor scenes without sufficient annotated data and generalize them to handle the fast accumulation of urban models in the form of meshes. In this work, we establish a 3D instance segmentation dataset for urban scenes and propose the first framework for 3D instance segmentation of buildings from urban MVS meshes.

III. Methodology

Compared to the lower parts of buildings that are more likely to be occluded by the nearby buildings and trees, building roofs usually have better visibility in aerial imaging. This observation motivates us to approach instance segmentation of entire buildings by looking into the segmentation of building roofs. In contrast to the previous work directly segmenting entire buildings [46], we perform roof segmentation by using a deep neural network. This strategy significantly improves the accuracy of the segmentation stage and simplifies the manual annotation in the data preparation stage.

One characteristic of our method is the hybrid process of 2D images and the corresponding 3D meshes, in which spatial occlusion is fully considered in processing the two distinctive data sources. Fig. 2 shows the proposed multi-view 3D instance segmentation framework that consists of three major steps:

1) **2D roof instance segmentation**. Roofs in multi-view images are automatically segmented by an instance segmentation neural network that is fine-tuned using our RGBH imagery dataset.
2) **Instance mask clustering**. An occlusion-aware clustering method for roof instance masks is exploited, which correlates instance masks from multi-view images to eliminate ambiguities. The mask clustering is the core of our method, which projects the 3D urban scene to the image space to measure the spatial overlap between arbitrary pairs of instance masks.
3) **3D building instance segmentation**. The clustered masks are projected back to the 3D space to segment 3D roof instances and the entire buildings are segmented in the end through an MRF optimization.

In the following part of this section, we introduce these three major steps, the benchmark dataset, and the implementation details.

A. 2D Roof instance segmentation

The challenge in building instance segmentation lies in that the roofs of adjacent (or even attached) buildings may have very similar appearances despite the difference in the height of the buildings. In this work, we take advantage of the complementary characteristics of the images and the 3D model of the scene by enhancing each RGB aerial image to an RGBH image that provides additional geometric cues. Specifically, we render a heightmap for each drone image using the 3D urban models reconstructed from the drone images and camera parameters. We add an additional channel encoding the height information to the drone images to obtain RGBH images, in which the height values are separately normalized for each image. With the RGBH images, we apply Swin Transformer [22] to segment roof instances automatically.

According to our quantitative evaluation on the benchmark dataset, the AP (average precision) [47] of the segmentation of roof instances on RGBH images reaches 0.582, which is significantly higher compared to 0.563 achieved on RGB aerial images. This demonstrates the advantage of height information on the roof instance segmentation. A visual comparison is shown in Fig. 3. Ground objects like trees and vegetable fields are successfully separated from buildings, even though some of them have visually indistinguishable textures from building roofs. The Swin Transformer neural network also computes a probability for each instance mask to represent its prediction confidence. To avoid low-confident masks, we only use roof instance masks whose prediction confidence is higher than 70%.

B. Instance mask clustering

After the roof instance segmentation, we obtain a set of roof instance masks from multi-view images, where multiple

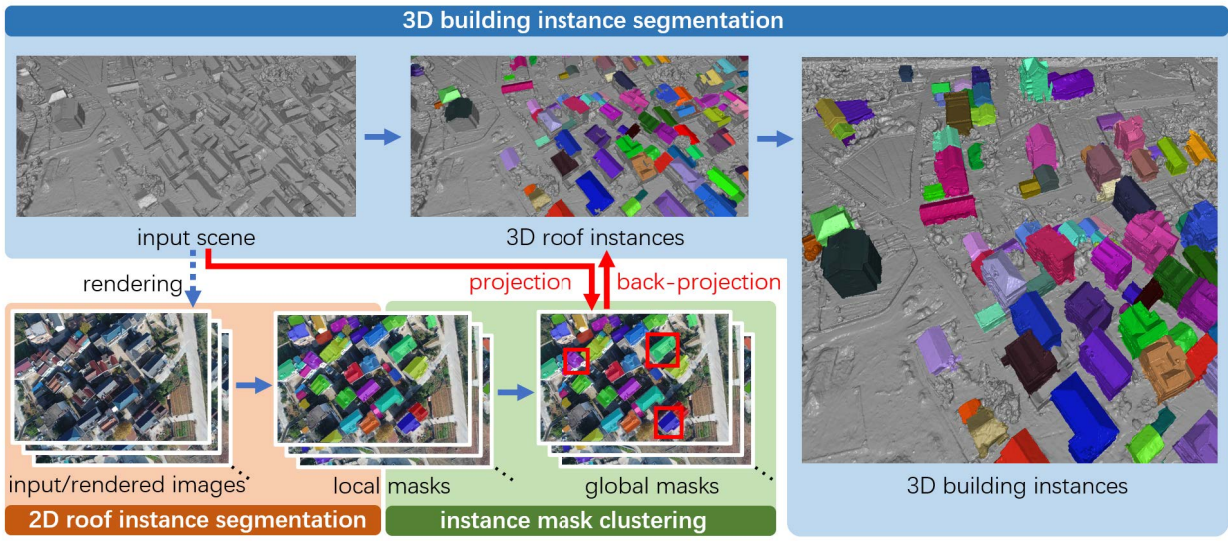
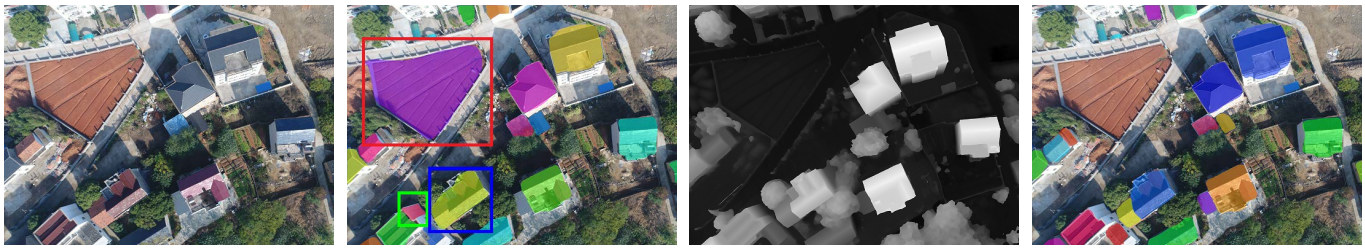


Fig. 2. An overview of the proposed method. Our method takes a 3D urban scene and optionally multi-view UAV images as input and segments all 3D building instances as results. It contains three major steps: 2D roof instance segmentation, instance mask clustering, and 3D building instance segmentation. The multi-view images are not obligatory, as they can also be generated by the rendering of the input 3D scene with textures (noted by the dotted arrow in the figure). The red rectangles highlight a few global masks selected by our clustering method. The projection and back-projection operations noted by the red arrows in the figure contribute to both instance mask clustering and the occlusion-aware 3D roof segmentation.



(a) (b) (c) (d)

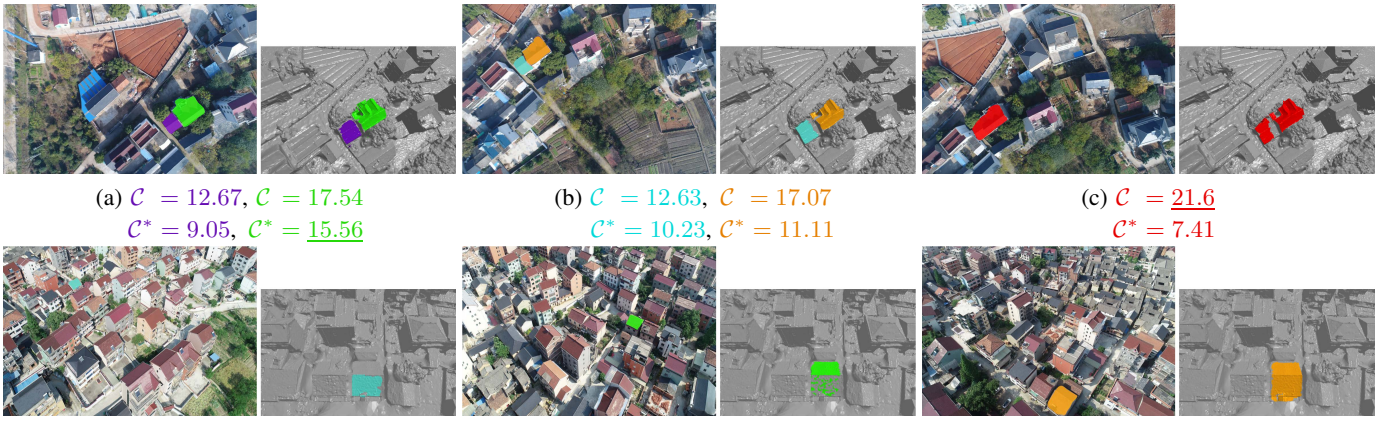
Fig. 3. Comparison of the segmentation results without and with height information. (a) An input test drone image. (b) The segmentation result using only the RGB image. The rectangles highlight three misclassified regions, i.e., a vegetable field (in red) and a wall (in green) of a building are segmented as roofs, and two roofs of attached buildings (in blue) are not separated. (c) The heightmap obtained by rendering the scene using the 3D model and the camera parameters. (d) The segmentation result using both the RGB image and the heightmap, where the vegetable field, wall, and roofs are all correctly separated.

masks may correspond to the same roof. Since roofs of the same building in multiple views have been segmented independently, the correspondences between roof instance masks are not known. This results in the number of instance masks being much larger than the number of roofs in the scene. To establish the correspondences between the masks from multi-view images, we refer to 3D roof instances by back-projecting the 2D roof masks onto the 3D model of the scene using the camera parameters. However, identifying masks that correspond to the same roof is challenging due to two main reasons: First, the 2D instance segmentation may have errors due to the limited capability of the neural network and the complex structure of the building roofs, as shown in the first row of Fig. 4. Second, the instance masks from different views are ambiguous due to different levels of spatial occlusion, as shown in the second row of Fig. 4.

To tackle these two challenges, we propose an instance mask clustering method that divides instance masks into different groups such that each group corresponds to a unique roof instance of an individual building. Representative masks are first selected from the segmented instance masks, and the remaining masks are merged with them according to mask similarity measures. For clarity, we refer to all roof instance masks in multi-view images as local masks, while the representative masks selected for clustering as global masks, as they represent unique building roofs across different images.

We first build a similarity matrix $\mathcal{M}$ to measure the spatial overlap for each pair of local masks. Based on $\mathcal{M}$, a mask with confidence value $\mathcal{C}$ to be selected as a global mask is computed for each local mask. Finally, all local masks are sorted in descending order to select reliable global masks and are clustered into groups according to their similarities with the global masks. Note that each mask group contains only one global mask. We establish the mapping between all local masks and global masks. In the following, we elaborate on these steps in detail.

Occlusion-aware mask similarity. To measure the spatial overlap of two masks, we project 3D mesh triangles to the image using the camera parameters. We render a depth map with the GPU acceleration for each multi-view image and



(a) $\mathcal{C} = 12.67$, $\mathcal{C} = 17.54$, $\mathcal{C}^* = 9.05$, $\mathcal{C}^* = \underline{15.56}$ (b) $\mathcal{C} = 12.63$, $\mathcal{C} = 17.07$, $\mathcal{C}^* = 10.23$, $\mathcal{C}^* = 11.11$ (c) $\mathcal{C} = \underline{21.6}$, $\mathcal{C}^* = 7.41$

(d) $\mathcal{C} = 17.43$, $\mathcal{C}^* = 12.15$ (e) $\mathcal{C} = 18$, $\mathcal{C}^* = 9.17$ (f) $\mathcal{C} = \underline{18.76}$, $\mathcal{C}^* = \underline{16.47}$

Fig. 4. Ambiguities in 2D roof instance segmentation. In each row, instance masks are shown in both the drone images (in different views) and the corresponding 3D mesh models (in an identical view). Top row: two roofs are separated in (a) and (b) but incorrectly mixed together in (c). Bottom row: (d) and (e) only cover a small part of the same roof, while (f) covers the entire roof. Original mask confidence values (denoted by $\mathcal{C}$) and improved mask confidence values (denoted by $\mathcal{C}^*$) are given in the sub-captions. The underlined numbers are the highest confidence values in each row.

employ a depth test to check the visibility for all the vertices. For the ith local mask, we record a set of triangles S_i whose centers are projected within this local mask region. A similarity matrix $\mathcal{M}_{n\times n}$ is then computed to quantify the spatial overlap between every pair of local masks, where n is the number of all local masks. The similarity element m_{ij} measures the intersection over union (i.e., *IoU*) between the ith and the jth local masks, i.e.,

$$m_{ij} = A(S_i \cap S_j)/A(S_i \cup S_j), \tag{1}$$

where $A(S)$ is the surface area of the triangles in the set S. $\mathcal{M}_{n\times n}$ is a symmetric matrix as $m_{ij} = m_{ji}$.

Mask confidence. Generally, an ideal global mask should have the most overlap with local masks that correspond to the same roof and have the least overlap with local masks that correspond to roofs of different buildings. To select such global masks, we estimate a confidence value $\mathcal{C}$ for each local mask to evaluate the overall overlap with all other local masks in the scene. It is calculated as the sum of similarity elements on the ith row of the similarity matrix $\mathcal{M}$:

$$\mathcal{C}_i = \mathcal{P}_i \cdot \sum_{j=1}^{n} \mathcal{P}_j \cdot m_{ij}, \tag{2}$$

where $\mathcal{P}_i$ is the probability value produced by the Swin Transformer neural network, $\mathcal{C}_i$ sums up the probability-weighted similarity elements of local masks. It makes sense because local masks with higher prediction confidences are more likely to be global masks.

However, there is still one drawback in Equation (2): local masks with large areas may suppress smaller ones because they likely overlap more with other local masks, and thus they obtain larger mask confidence values. Local masks with larger areas are not always the ideal global masks. The top row of Fig.4 shows such a counter-example. To solve this issue, we define a binary term Δ_{ij} to avoid such unexpected suppression:

$$\Delta_{ij} = \delta(m_{ij} - \beta), \tag{3}$$

where $\delta(\cdot)$ is the delta function:

$$\delta(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}. \tag{4}$$

With this binary term, the mask confidence is updated to:

$$\mathcal{C}_i^* = \mathcal{P}_i \cdot \sum_{j=1}^{n} \Delta_{ij} \cdot \mathcal{P}_j \cdot m_{ij}. \tag{5}$$

$\mathcal{C}_i^*$ sums up the similarity elements m_{ij} whose values are higher than a threshold $\beta \in [0, 1]$ for each local mask. When β is close to 0, $\Delta_{ij} = 1$ for most cases and thus $\mathcal{C}_i^*$ degenerates to $\mathcal{C}_i$. When β is close to 1, Δ_{ij} and $\mathcal{C}_i^*$ are both close to 0, which means that the confidence values of all local masks to be selected as global masks are close to 0. In such a case, the clustering cannot distinguish global masks from local masks, resulting in false clustering in the end. In this work, we set $\beta = 0.5$ in all our experiments, indicating that local masks with similarities (i.e., the ratio of the overlapping area) higher than β are considered in the computation of mask confidence values. More details about the evaluation of the parameter β can be found in the implementation details in Subsection IV-C.

Mask clustering. One key observation of this work is that local masks with higher confidence values are consistent with other masks and thus should have higher priority to be selected as global masks, as shown in Fig. 4.

Based on the mask confidence, we employ a simple yet efficient order-based mask clustering. We first sort all local masks according to their confidence values $\mathcal{C}^*$ and then traverse them in descending order to select global masks. In the traversing loop, if a local mask has not been marked, we mark it as a new global mask, and other non-marked local masks whose similarities with this global mask are higher than β are considered consistent with this global mask, i.e., $\delta(m_{lg} - \beta) = 1$ where l and g are the indices of the local

mask and this global mask, respectively. If a local mask has been already marked, we traverse to the next local mask until all of them are marked.

With the pre-computation of mask confidence values, the traversal is required only once. For efficiency, we establish a mapping table $\mathbb{M}$ between all local masks to their corresponding global masks. It is worth noting that even though we cannot guarantee each global mask in $\mathbb{M}$ corresponds to a building instance in the real scene at this stage, a few false correspondences will not affect the final 3D instance segmentation. This will be explained in Subsection III-C.

C. 3D building instance segmentation

3D roof instance segmentation. The existing multi-view semantic segmentation framework projects the 2D semantic labels back to the 3D model with the highest probability [48], [49]. For 3D instance segmentation, this framework is not suitable because the correspondences between local masks from multi-view images are unknown. Our mask clustering establishes the correspondences between the local masks from multiple views. We first project each vertex of the 3D model to all images and check its visibility using fields of view and depth maps. Then, using its corresponding local mask index at its projected position in the image, we retrieve its corresponding global mask from the mask mapping table $\mathbb{M}$.

For a vertex v, let MVI_v denote the set of multi-view images in which v is visible and GMI_p denote the global mask index corresponding to a multi-view image p. In some cases, a vertex v on the 3D surface model may be projected within multiple global masks. We denote the set of these global masks as $\{GMI_p|p \in MVI_v\}$ and thus $GMI_p = -1$ represents the background. From these global masks, the one with the largest quantity of corresponding local masks that were projected to by this vertex is associated with this vertex:

$$RID_v = maxCount(\{GMI_p|p \in MVI_v\}), \tag{6}$$

where RID_v is the roof ID of vertex v, and the function $maxCount(S)$ extracts the most occurring element in the set S. In case more than one global masks have the maximum count in the set S, the global mask with a smaller value of GMI will be chosen, as a smaller GMI value corresponds to a higher confidence of the global mask.

The advantage of determining the roof IDs in this way is that the most confident global mask can be automatically selected, and thus a user does not have to provide a specific number of target clusters (i.e., the number of global masks). This is because local masks with large errors in 2D roof segmentation are normally divided into groups containing small numbers of local masks. The global masks derived from these local masks usually have wrong predictions and therefore will be ignored in the upcoming 3D roof segmentation step, since only the global mask with the largest quantity of corresponding local masks is selected. Therefore, our 3D roof instance segmentation achieves higher prediction precision than the roof instance segmentation on the multi-view images. Their AP/AP50/AP75 values can be found in Subsection III-A and Table II respectively.

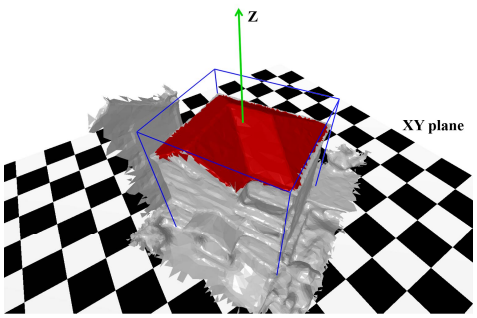


Fig. 5. Horizontal Oriented Bounding Box ($HOBB$). The top and bottom faces of the $HOBB$ are horizontal, and the other four faces are oriented according to the PCA orientation estimation.

With the roof IDs for all vertices, the segmentation of the 3D model can be easily obtained. Specifically, the roof ID of a triangle face is determined by the majority of the vertices that indicate the same roof ID, i.e.,

$$RID_t = maxCount(\{RID_v|v \in V_t\}), \tag{7}$$

where RID_t represents the roof ID of triangle t, and V_t represents the three vertices of the triangle t.

MRF-based 3D building segmentation. Based on 3D roof segmentation, the next step is to segment the entire 3D buildings. We first estimate a Horizontal Oriented Bounding Box ($HOBB$) for each roof instance, as shown in Fig. 5. To segment an entire building from a 3D scene, we expand the $HOBB$ on its four sides by a certain offset value (4 meters in all of our experiments). The triangles within the expanded $HOBB$ (excluding those that have been labeled by other roof instances) are selected as a candidate building. We denote its triangle set as $T = \{t_i\}$ and its edge set as $E = \{e_{ij}\}$, in which t_i represents the ith triangles in T, and e_{ij} represents the edge connecting t_i and t_j. We formulate the building segmentation as a foreground/background labeling process that minimizes the following energy function:

$$\psi(l) = \sum_{t_i \in T} \psi_{data}(l_i) + \sum_{e_{ij} \in E} \psi_{smooth}(l_i, l_j), \tag{8}$$

where l_i denotes the label of triangle t_i given by MRF-based segmentation. $l_i = 1$ indicates the foreground (i.e., a building triangle) and $l_i = 0$ the background (i.e., a non-building triangle).

The data term $\psi_{data}(l_i)$ represents the penalty of assigning a label l_i to a triangle t_i. The 3D roof segmentation provides us a good foreground initialization, we denote its triangle set as T_f. We take triangles on the boundary of T as a background initialization, denoted as T_b. We further denote the set of other triangles in T as T_r. As shown in Fig. 7(a), T_f, T_b, and T_r are visualized in red, green, and gray, respectively.

For triangles in T_r, our data term $\psi_{data}(l_i)$ is defined as:

$$\psi_{data}(l_i) = \begin{cases} (1 + d_i) + \theta_i \cdot (1 + d_i) & \text{if } l_i = 1 \\ 1 & \text{if } l_i = 0 \end{cases}. \tag{9}$$

Before explaining the meaning of d_i and θ_i, we define $\mathcal{P}$, a 2D polygon representing the simplified roof boundary edges, as shown in Fig. 6. We first extract boundary edges of the

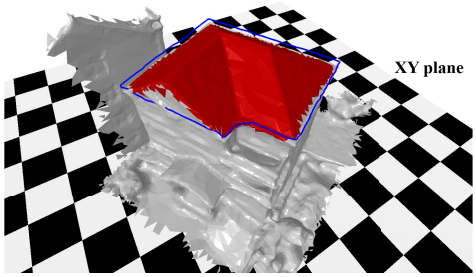


Fig. 6. Simplified roof boundary edge $\mathcal{P}$. The blue polygon represents the flat boundary of the roof on the XY plane. To make it easy to observe, we set its Z coordinate to be close to the height of the roof.

roof triangles using the Alpha Shape algorithm [50]. Then, we reduce the dimension of these boundary edges to the 2D horizontal plane by discarding their height and simplifying them through a RANSAC process [51]. The RANSAC process iteratively merges outline points of roofs to their neighbors if they are approximately collinear. The process stops until no more points can be merged. The simplified roof boundary edges constitute a 2D polygon that is referred to as the roof profile, denoted as $\mathcal{P}$. For each triangle $t_i \in T_r$, we find a line segment in $\mathcal{P}$ with a minimal distance to the center of t_i. We denote this minimal distance as d_i, and the cosine of the horizontal angle between the normal of t_i and the direction of this line segment as θ_i. $\theta_i = 0$ when the normal of t_i is perpendicular to this line segment, and $\theta_i = 1$ when their horizontal angle is 0. Then d_i is normalized by the maximum distance in the $HOBB$, and we sign the distance as negative if t_i is inside of $\mathcal{P}$. To make the foreground penalty and background penalty comparable on the roof profile (i.e., both penalties equal to 1 when $d_i = 0$ and $\theta_i = 0$), we use $1 + d_i$ instead of d_i.

In Equation 9, the $(1 + d_i)$ term is a distance constraint that guarantees only triangles close to $\mathcal{P}$ are taken into the building. The $\theta_i \cdot (1+d_i)$ term is an orientation constraint that guarantees that only triangles having similar orientations with the closest line segment are taken. The $(1+d_i)$ multiplication is to reduce the orientation constraint if the triangles are far away from $\mathcal{P}$.

The smoothness term $\psi_{smooth}(l_i, l_j)$ penalizes adjacent triangles t_i and t_j being assigned with different labels. We take the cosine angle between normals of adjacent triangles as the penalty for assigning different labels to the adjacent triangle pair, i.e.,

$$\psi_{smooth}(l_i, l_j) = \begin{cases} ||n_i \cdot n_j|| & \text{if } l_i \neq l_j \\ 0 & \text{if } l_i = l_j \end{cases} . \qquad (10)$$

Using the angles between faces, $\psi_{smooth}(l_i, l_j)$ favors segmentation at sharp edges rather than at planar regions.

D. Benchmark dataset

We have created a benchmark dataset *InstanceBuilding* that contains annotation for both UAV images and 3D urban scenes simultaneously. To evaluate our 3D instance segmentation method, we annotated 3D roofs and buildings for 4 large 3D urban scenes, which are reconstructed using *Bentley Acute3D ContextCapture*[2] from UAV images. Table I shows detailed information about these scenes, and their visualization can be found in Fig. 8 and the supplementary video. Note that the town is quite crowded, thus about $2/3$ buildings are attached to others, as shown in the last column of Table I. To facilitate the 3D annotation, we have developed a simple but efficient brush-based annotation tool. Similar to most 2D annotation tools [52] which semi-automatically extract pixels of an object by marking the closed boundary polygon of the object, our tool allows a user to segment a 3D building by casually drawing strokes on the building boundaries.

TABLE I
STATISTICS ON THE 3D MODELS OF OUR *InstanceBuilding* DATASET.

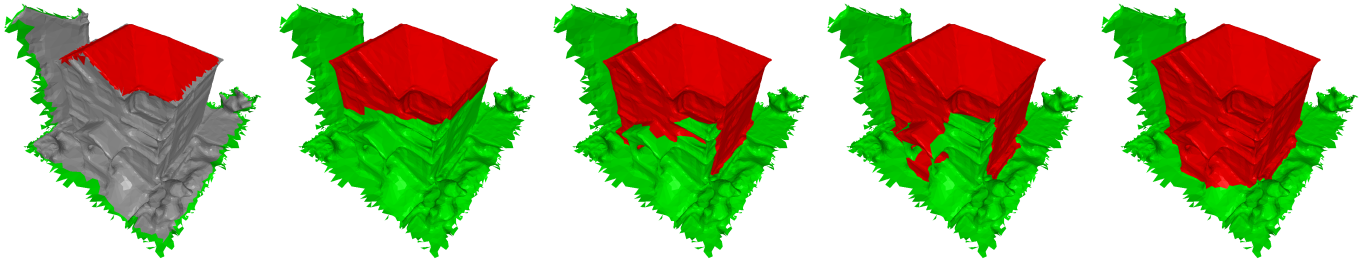
Scene	#Vertices	#Triangles	Area (km^2)	#Images (resolution)	#Buildings All / Attached
#1	1.13M	2.26M	0.076	79 (5472×3648)	185 / 145
#2	0.87M	1.72M	0.097	64 (5472×3078)	119 / 40
#3	1.14M	2.28M	0.081	284 (1916×994)	322 / 232
#4	0.60M	1.20M	0.18	240 (1536×994)	266 / 185

Our *InstanceBuilding* dataset also contains 608 annotated images with high resolutions. They are selected from around 20 thousand images acquired in more than 10 different cities. Some are directly captured by a consumer DJI drone Phantom 4 Pro with different cameras and flight altitudes, others are rendered by 3D models with textures as orthophotos with similar resolutions. There are about 16 thousand buildings in all these images, and their roofs are all manually annotated for the training of our 2D roof instance segmentation neural network. These annotated images are divided into 2 groups, 524 images for training and 84 images for validation.

For accuracy and efficiency consideration, we modified the LabelMe [52] toolkit to visualize the corresponding heightmap window alongside the color image window. By synchronizing the annotation on both the image window and the heightmap window, volunteers can freely annotate on either of them. Based on our time recording of volunteer annotation, this double-window strategy saves the volunteers more than half of the time.

Our *InstanceBuilding* dataset contains building instance annotation for both 3D urban scenes and UAV images simultaneously, which makes it unique. Most of existing 3D datasets are designed for semantic segmentation, such as Vaihingen3D [53], Swiss3DCities [54], Hessigheim3D [55], and SUM [56]. The most related work about 3D instance segmentation dataset is the Urban Drone Dataset (UDD) [57] and UrbanScene3D [58]. UDD evaluates their 3D segmentation accuracy by projecting them to drone images, thus cannot be regarded as a 3D dataset. UrbanScene3D does not contain corresponding UAV images, has only 485 annotated buildings and very few buildings are attached to others. As shown in Table I, our *InstanceBuilding* dataset has 892 annotated buildings, of which 602 are attached to others. In such crowded

[2]https://www.bentley.com/en/products/brands/contextcapture



(a) (b) (c) (d) (e)

Fig. 7. 3D building segmentation based on MRF optimization. (a) Initial segmentation by the projection of the 2D roof onto the 3D model (foreground in red and background in green). (b) Result of direct MRF optimization, where the initial roof segmentation misses the balcony and the gate shelter. (c) Our segmentation result without the orientation constraint. (d) Our segmentation result without the distance constraints. (e) Our segmentation result using both orientation and distance constraints.

urban scenes, instance segmentation has a more significant advantage over semantic segmentation.

Compared to existing natural image datasets, such as MSCOCO [47], Cityscapes [59], and PASCAL VOC [60], our annotation on UAV images focuses on roof instances for UAV images. Compared to other remote sensing imagery datasets, such as SpaceNet [61] and xView [62], our UAV images have higher resolution and more viewing angles, thus are an important supplement to existing datasets. Though there are also many drone datasets in public, such as VisDrone [63] and DOTA [64], few of them focus on instance segmentation at the pixel level. With the increasing popularity of low-altitude UAV capturing devices, we believe that our dataset will play an important role in applications such as urban planning, smart cities, and other related fields.

E. Implementation details

2D roof segmentation. We use the Pytorch implementation of Swin Transformer released by [65] for roof instance segmentation from images. The manually labeled images (see Subsection III-D) were used to fine-tune the Swin Transformer network trained previously using the COCO dataset [47]. For 3D urban scenes that have corresponding UAV images, we directly segment these images. For 3D urban scenes that do not have corresponding UAV images, multi-view images are rendered using these 3D scenes from a set of different viewpoints sampled randomly at 70 meters higher than the average height of these scenes. At each viewpoint, 5 virtual cameras facing down, front, back, left, and right are placed.

MRF-based segmentation. With the aforementioned MRF setting, we treat Equation 8 as a classical max-flow/min-cut optimization and solve it using the graph cut algorithm [66]. The MRF-based segmentation result is shown in Fig. 7 (e). To prove the validity of Equation 9, we simply set the penalties to 1 for both foreground and background in Equation 9, as the triangles in T_r have no explicit foreground/background priorities. As a result, it does not produce correct segmentation as expected, as shown in Fig. 7 (b). Comparisons in Fig. 7 (c) and (d) show that the introduction of the orientation and the distance constraint overcomes the interference of structural variations and noises in the MRF optimization, thus improving the 3D building segmentation.

IV. Results and Discussion

We have evaluated our method with both drone images and virtually rendered images. All experiments were carried out on a machine with an Intel Core i7 processor, 32 GB memory, and an NVidia GeForce 1080 GPU.

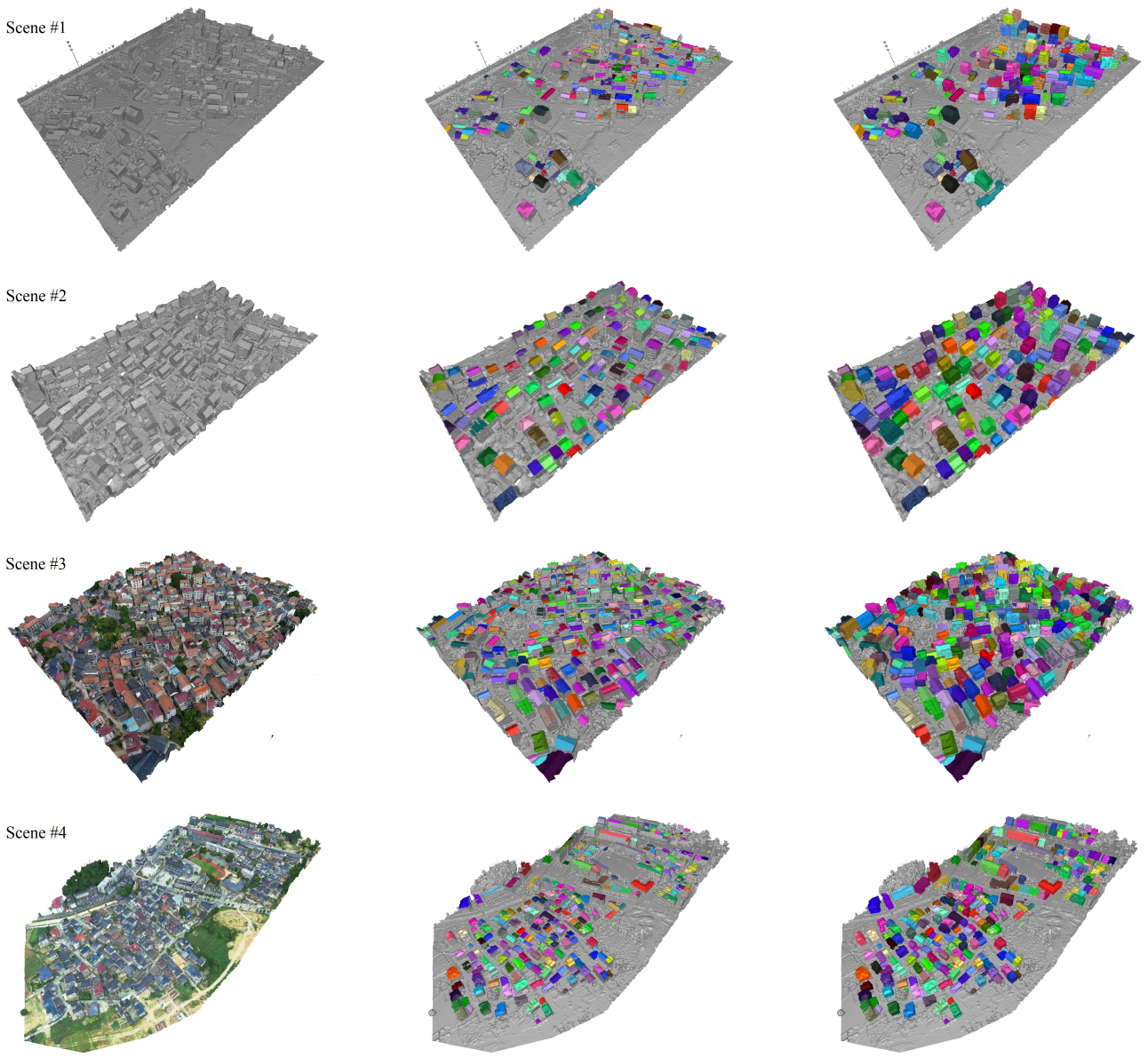
A. Qualitative results

We tested our method on four scenes (see Fig. 8), for which the statistics are shown in Table I. Scene #1 and Scene #2 have UAV images and the corresponding camera parameters. Scene #3 and Scene #4 do not have UAV images, we rendered images from multiple viewpoints instead, as explained in the subsection III-E. The roof instance segmentation results are shown in Fig. 8 (middle column) and the 3D building instance segmentation results are shown in Fig. 8 (right column). We can see that our approach successfully segmented most buildings, even if they are dense, varying in style, and attached.

B. Ablation analysis

We have evaluated our results on the 3D mesh models in the *InstanceBuilding* benchmark dataset. We first computed the *IoUs* between predictions and the ground truth based on the area of the mesh triangles. Then we used the commonly used instance-level evaluation metrics, namely AP, AP50, and AP75, in all evaluations [47], where AP50/AP75 indicates the average precision when *IoU* threshold is set to 0.5/0.75, and AP is averaged over 10 *IoU* thresholds of 0.5:0.05:0.95.

Height information. As demonstrated in Fig. 3, using RGBH images significantly improves the 2D roof instance segmentation. To understand how the height information improves 3D roof instance segmentation and its effects on 3D building instance segmentation, we have conducted a comparison on the four large 3D scenes from *InstanceBuilding* both with and without height information, using two different clustering methods, i.e., the spectral clustering and our method. The results of the comparison are given in Table II and Table III. These comparisons have revealed that our method using RGBH images achieves higher accuracy than the one without height information (i.e., using RGB images). This is because the additional heightmaps provide spatial information of the urban scenes to the neural network model, which makes the roofs and buildings more distinguishable.



(a) Initial scene models (b) 3D Roof segmentation results (c) 3D Building segmentation results

Fig. 8. Building instance segmentation results of four scenes. The two scenes on the top have UAV images with camera parameters, while the scenes #3 and #4 in the bottom do not have UAV images, for which we render images from multiple viewpoints instead.

TABLE II
COMPARISON OF FOUR DIFFERENT CLUSTERING METHODS ON 3D *roof* INSTANCE SEGMENTATION USING SWIN TRANSFORMER.

Scene	Spectral with RGB (K_1)			Spectral with RGBH (K_1)			Spectral with RGB (K_2)			Spectral with RGBH (K_2)			Ours with RGB			Ours with RGBH		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
#1	0.5858	0.7841	0.5966	0.6551	0.8239	0.7045	0.5989	0.7784	0.6364	0.6665	0.85227	0.7159	0.6455	0.8409	0.6818	**0.7114**	**0.8693**	**0.7727**
#2	0.5329	0.6973	0.5502	0.4579	0.6446	0.4463	0.5436	0.7021	0.5633	0.5826	0.8347	0.5785	0.5651	0.7190	0.5880	**0.6610**	**0.8395**	**0.6707**
#3	0.5778	0.7663	0.6095	0.5501	0.7284	0.6068	0.5900	0.8162	0.6482	0.5966	0.8398	0.6396	0.6179	0.8476	0.6604	**0.6434**	**0.9072**	**0.6618**
#4	0.4447	0.7444	0.4398	0.4526	0.7256	0.4549	0.4583	0.8008	0.4323	0.4635	0.7820	0.4474	0.5011	0.8383	0.4962	**0.5248**	**0.8459**	**0.5301**

The multi-view framework. The 2D segmentation in our method is applied to multi-view images, rather than orthophoto maps. To understand the advantages of this multi-view framework, we have also implemented an orthophoto-based instance segmentation solution for comparison. We first generated the orthophoto maps and their corresponding heightmaps of the same scenes through a render-to-texture technique. These two types of maps were combined to form the orthophoto RGBH maps, which were then used to detect the 2D roof instances. Since there was almost no occlusion for the roofs in the orthophoto maps, mask clustering was not necessary and we directly applied the 3D building segmentation to produce the final results. The statistics of the results are reported in Table IV.

From this comparison, we can see that our multi-view method achieves higher AP than the orthophoto-based method. There are three main reasons for this improvement. The first advantage of using multi-view images lies in the mask clustering that integrates multiple segmentation results of the identical roofs from different viewpoints. This mask clustering process can be regarded as a cross-correction process, thus eliminating most of the incorrect instance masks from individual multi-view images. Secondly, it is very difficult to separate two roofs if they are attached and have similar textures using orthophoto maps. In contrast, our multi-view framework has much more information from building walls, thus achieving more accurate segmentation. Finally, the original drone images usually have high resolution without texture distortion, but orthograph maps may have these drawbacks due to the texture mapping and synthesis on the 3D meshes.

Mask clustering. The instance mask clustering is crucial to overcome the ambiguities in the multi-view instance masks. Many clustering methods are available in the literature, such as Mean-shift [67] and K-means [68], but none of them is suitable for our multi-view scenario. These clustering methods require computing the mean of features, which heavily relies on a good feature extractor. For example, the position, size, and mean color of masks cannot accurately describe their features. The other difficulty of these clustering methods is choosing the optimal values of parameters, such as the kernel function for Mean-shift, and the K value for K-means. Note that our clustering method does not require specifying the number of target clusters, which is the core advantage of our method.

What is more important is that it is still unknown how to design comparable features for masks from different views, especially when they have a large variation among different viewpoints. Compared with viewpoint-variant features, the spatial overlap between masks can be calculated accurately. Therefore, we directly calculate a similarity matrix using the spatial overlap between masks, rather than their features.

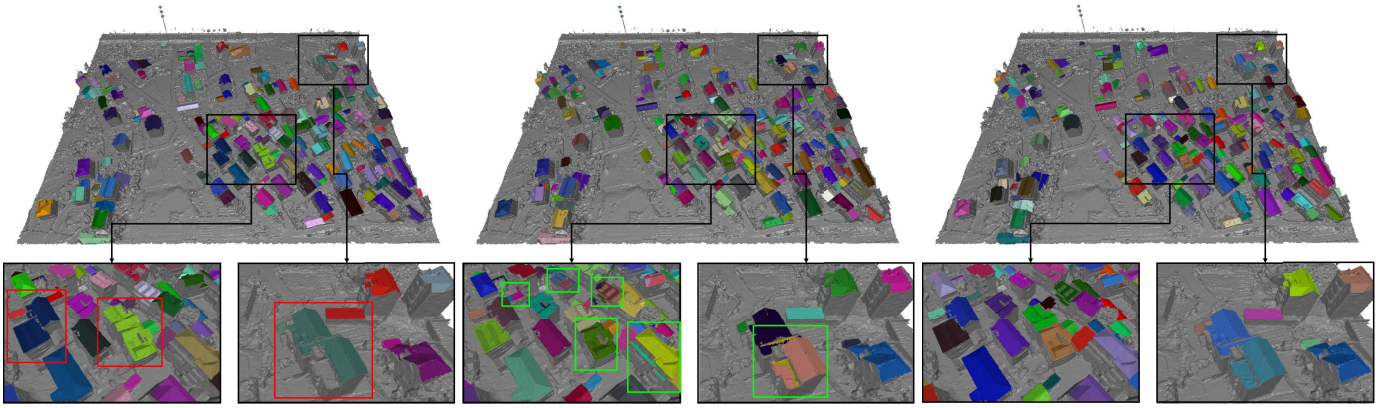
Based on the similarity matrix, one alternative option for mask clustering is spectral clustering [69]. We have evaluated it with different numbers of target clusters, noted as K, and compared our clustering method with it. For a fair comparison, we used two different values of K for the spectral clustering method on each scene: 1) K_1: the building number of the annotated 3D scene, and 2) K_2: the number of global masks estimated by our clustering method. Note that K_2 is typically larger than K_1, as some of the global masks did not correspond to real 3D roofs. A more detailed explanation is given in Subsection III-C. Specifically, K_1 has a value of $185/185, 119/119, 322/322, 266/266$, and K_2 has a value of $385/475, 202/376, 872/714, 646/641$ for Scene #1, #2, #3, and #4 with RGB/RGBH images, respectively.

The advantages of our clustering method over spectral clustering can also be concluded from Table II and Table III. No matter on the 3D roof or building instance segmentation, using K_1 or K_2, with or without height information, our clustering method always reaches higher precision than spectral clustering. Visual comparison for Scene #1 is shown in Fig. 9. From this comparison, we can observe that spectral clustering has the issue of under-segmentation when $K = K_1$. Since roof instance masks detected from the multi-view images are much more than the ground truth, and have prediction errors as well, thus the spectral clustering tends to incorrectly mix some roofs of attached buildings when $K = K_1$. Meanwhile, the spectral clustering has the issue of over-segmentation when $K = K_2$. Since K_2 is larger than the ground truth roofs in the 3D scene, it lost the ability to separate the correct and incorrect masks, leaving these masks separated. In contrast, our instance mask clustering successfully segments roof instances precisely without specifying the number of target roofs. More visual comparison results can be found in the supplementary video.

It is worth pointing out that the last block named "Ours with RGBH" of Table II also shows that we achieve higher AP/AP50/AP75 on the 3D roof instance segmentation than on the aerial images (values are shown in Subsection III-A), because our occlusion-aware mask clustering suppresses false prediction from individual images and thus improves the overall segmentation precision.

Alternatives for image instance segmentation. The core contribution of our work is the multi-view framework with a new instance mask clustering, not the image instance segmentation. Besides Swin Transformer, our framework can incorporate any other image instance segmentation method as well. To demonstrate its compatibility, we have testified it with Mask R-CNN [18], which is another widely-used image instance segmentation model. Table V shows a comparison between our framework based on multi-view images and the one based on orthophoto maps using Mask R-CNN. Similar to using Swin Transformer, the advantages of our multi-view method over the orthophoto-based method can also be found in this table. This demonstrates the effectiveness of our multi-view framework regardless of the chosen image instance segmentation method. Comparisons with the spectral clustering method and different MRF segmentation constraints using Mask R-CNN are shown in the supplementary document. For all these results, our method consistently achieves the highest accuracies. It is worth noting that most of the evaluation results for Mask R-CNN are lower than those of the recently developed Swin Transformer.

MRF-based building segmentation. As shown in Fig. 7, the orientation and distance constraints are important to achieve accurate 3D building segmentation. We have also evaluated the segmentation precision and recall by omitting one of them. The results are reported in Table VI, from which we can conclude that the distance constraint plays a more important role than



(a) Spectral clustering with K_1 (b) Spectral clustering with K_2 (c) Our clustering result

Fig. 9. Comparison of roof mask clustering of three methods on Scene #1. (a) Spectral clustering using the ground truth number of roof targets K_1 tends to under-segment (red rectangles) some roofs. (b) Spectral clustering using the number of global masks K_2 estimated by our method tends to over-segment (green rectangles) some roofs. (c) Our clustering method achieves more precise roof instance segmentation without specifying the target number.

TABLE III
COMPARISON OF FOUR DIFFERENT CLUSTERING METHODS ON 3D *building* INSTANCE SEGMENTATION USING SWIN TRANSFORMER.

Scene	Spectral with RGB (K_1)			Spectral with RGBH (K_1)			Spectral with RGB (K_2)			Spectral with RGBH (K_2)			Ours with RGB			Ours with RGBH		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
#1	0.5881	0.7784	0.6054	0.6362	0.8162	0.6973	0.6184	0.8216	0.6270	0.6843	0.8703	0.7351	0.6486	0.8324	0.6649	**0.7130**	**0.8919**	**0.7730**
#2	0.5311	0.7059	0.5210	0.4739	0.6302	0.4705	0.5615	0.7699	0.5666	0.5798	0.8235	0.5798	0.5709	0.7479	0.5966	**0.6655**	**0.8403**	**0.6807**
#3	0.4373	0.6429	0.4658	0.5369	0.7764	0.5745	0.5469	0.8168	0.5590	0.5730	0.8354	0.5994	0.6357	0.8882	0.6925	**0.6671**	**0.9286**	**0.7174**
#4	0.5739	0.7632	0.6128	0.5752	0.7481	0.6090	0.6192	0.8346	0.6504	0.6248	0.8459	0.6429	0.6534	0.8571	0.6992	**0.6726**	**0.8759**	**0.7105**

TABLE IV
COMPARISON BETWEEN OUR FRAMEWORK BASED ON MULTI-VIEW IMAGES AND THE ONE BASED ON ORTHOPHOTO MAPS USING SWIN TRANSFORMER.

Scene	Orthophoto-based			Ours (multi-view)		
	AP	AP50	AP75	AP	AP50	AP75
#1	0.6389	0.8432	0.6541	**0.7130**	**0.8919**	**0.7730**
#2	0.5975	0.7906	0.6059	**0.6655**	**0.8403**	**0.6807**
#3	0.5935	0.9006	0.6429	**0.6671**	**0.9286**	**0.7174**
#4	0.6199	0.8195	0.6391	**0.6726**	**0.8759**	**0.7105**

TABLE V
COMPARISON BETWEEN OUR FRAMEWORK BASED ON MULTI-VIEW IMAGES AND THE ONE BASED ON ORTHOPHOTO MAPS USING MASK R-CNN.

Scene	Orthophoto-based			Ours (multi-view)		
	AP	AP50	AP75	AP	AP50	AP75
#1	0.6751	**0.8595**	0.7189	**0.7270**	**0.8595**	**0.7730**
#2	0.5969	0.7176	0.5847	**0.6202**	**0.7227**	**0.6471**
#3	0.5475	0.8199	0.5714	**0.6270**	**0.8354**	**0.7019**
#4	0.6079	**0.7895**	0.6278	**0.6117**	**0.7895**	**0.6353**

the orientation constraint, but the best segmentation accuracy can only be achieved if they are both employed.

C. Effects of parameters

Our method involves a few parameters, among which β in the mask clustering step is the only parameter that we leave tunable for users (while all other parameters are fixed). In this subsection, we discuss how this parameter affects mask clustering.

Intuitively, the meaning of β parameter in our work is very similar to the threshold parameter of *IoU* in many existing object detection works where a mask is considered to be correctly predicted when the *IoU* between the detection mask and the ground truth is greater than this threshold. Empirically, this threshold is initially set to 0.5. Similarly, in our mask clustering, if the *IoU* of the two local masks is greater than β, they should be considered to belong to the same group. That is, they represent the same roof instance. In this work, we initially set $\beta = 0.5$ in all of our experiments. To determine the optimal value for β, we experimented with different values in all 4 scenes. As we can see from Fig. 10, the AP, AP50, and AP75 are always close to the highest values when $\beta = 0.5$. Note that slightly increasing/reducing β reduces/increases the confidence values but barely affects the ordering of the masks. This reveals that our mask clustering is tolerant to the β parameter.

D. Running time

The training took around 83 hours with 2000 epochs. The overall running time for segmenting a scene varied from 6 to 8 minutes, depending on the scene size, image resolution, and the number of images. The MRF optimization takes around 2.5 minutes on average for each scene. The computational complexity of multi-view image generation (only for scenes without drone images), heightmap generation, 3D vertex projection for the overlapping computation, and back projection for the clustered instance masks are $O(N * k)$, where N is the

TABLE VI
COMPARISON OF USING DIFFERENT SEGMENTATION CONSTRAINTS ON 3D *building* INSTANCE SEGMENTATION USING SWIN TRANSFORMER.

Scene	Without the orientation constraint			Without the distance constraint			With both constraints		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
#1	0.6438	0.8541	0.7189	0.6856	0.8673	0.7537	**0.7130**	**0.8919**	**0.7730**
#2	0.5605	0.8235	0.5966	0.6378	0.8325	0.6722	**0.6655**	**0.8403**	**0.6807**
#3	0.4252	0.8540	0.3882	0.6177	0.9224	0.6770	**0.6671**	**0.9286**	**0.7174**
#4	0.5289	0.8308	0.5301	0.6176	0.8722	0.6466	**0.6726**	**0.8759**	**0.7105**

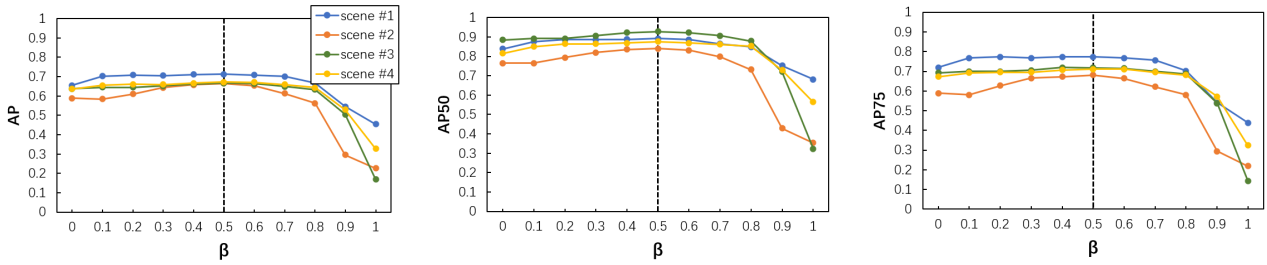
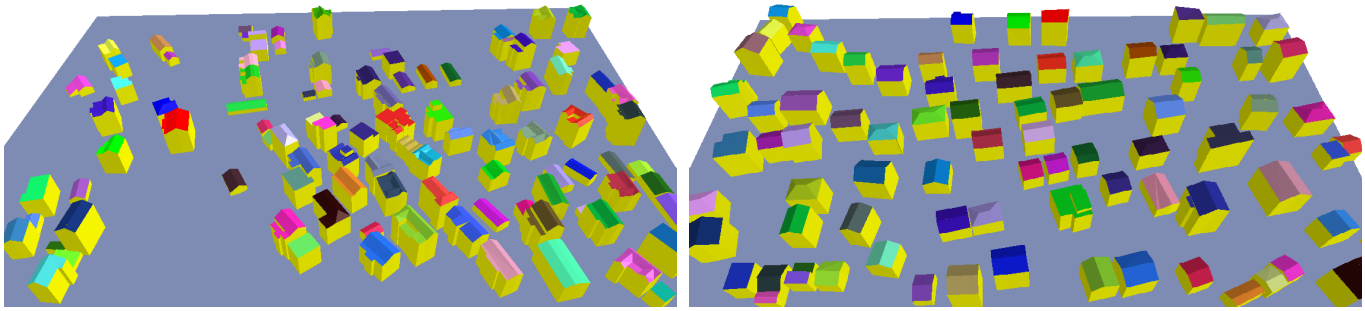


Fig. 10. Evaluation of the segmentation results with different β values using Swin Transformer. The AP (left), AP50 (middle), and AP75 (right) of the segmentation result are always close to the highest values when $\beta = 0.5$.



(a) Scene #1 (b) Scene #2

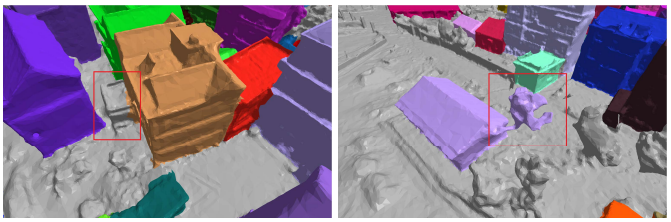
Fig. 11. Automatic simplification of buildings in two large urban scenes. The results are obtained by applying RANSAC for plane extraction followed by plane regularization.

number of faces in this 3D urban model, and k is the number of multi-view images. N could be a large number, but these computations are fully accelerated using the GPU, only taking less than 1 minute for each urban scene. The instance mask clustering has a computational complexity of $O(n)$, where n is the number of instance masks, thus it takes less than 1 second in total for each urban scene. It is much faster than many other traditional clustering methods.

E. Discussions

Applications. We have implemented a simple building simplification prototype using a RANSAC-based plane fitting [51] and plane regularization. Based on our 3D building instance segmentation, 3D buildings in large urban scenes are automatically simplified, as shown in Fig. 11. In such an application, instance segmentation is obligatory, as semantic segmentation is insufficient to separate individual buildings. We believe our 3D building instance segmentation method can benefit more smart city applications, such as urban planning.

Limitations. Since our approach falls into the multi-view paradigm, it relies on the quality of the 2D instance segmentation. Roof types that do not exist in the training dataset may not be precisely segmented. Two such examples are shown in Fig. 12. In Fig. 12 (a), a gated shelter was not reliably detected, and in Fig. 12 (b), a tall tree was segmented as a part of the nearby building. Enriching the training dataset may partially solve this issue. In addition, developing a neural network dedicated to separating roofs and trees may produce more reliable instance segmentation results.



(a) (b)

Fig. 12. Two failure cases of our method. (a) A gated shelter was not segmented as part of the roof in the image segmentation stage. (b) A tall tree was segmented as a roof instance because it has a similar height and texture as its nearby building.

V. Conclusion and Future Work

We have presented a multi-view framework for instance segmentation of 3D urban buildings. Based on occlusion-aware similarity matrices, a novel instance mask clustering method is proposed to eliminate the mask ambiguities among multi-view images. To further improve segmentation accuracy, roofs (instead of buildings) are firstly segmented, and RGB images are enriched with heightmaps. Our method takes full advantage of the multi-view framework to precisely segment 3D buildings in large urban scenes.

We have collected and annotated an RGBH drone imagery dataset and a 3D building instance segmentation dataset, named *InstanceBuilding*. We believe the new dataset could benefit research in 3D instance segmentation for various urban applications. Since most of the state-of-the-art learning-based 3D instance segmentation methods focus on indoor scenes, our multi-view instance segmentation framework explores a new avenue for large outdoor scenes.

Future directions. Our work focuses on buildings because they are the most important ingredients in the urban environment. One future direction is to extend our multi-view 3D instance segmentation framework to other urban objects and even indoor scenes. Drone images and the reconstructed 3D models may suffer from various degradation (such as noises), it is worth investigating the robustness of our method in such degraded scenarios[70]. Finally, applying our method to 3D point clouds of urban scenes could be an interesting future direction as well.

Acknowledgment

The authors would like to thank Quzhou Southeast Flysee Technology Ltd. for providing drone images, 3D urban models and parts of 2D annotations. This work was supported in part by National Natural Science Foundation of China (62172367), and in part by Natural Science Foundation of Zhejiang Province (LGF22F020022).

References

[1] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven, "High accuracy and visibility-consistent dense multiview stereo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 5, pp. 889–901, 2012.

[2] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," *arXiv preprint arXiv:1804.02505*, 2018.

[3] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[4] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[7] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3947–3956.

[8] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2569–2578.

[9] J. Hou, A. Dai, and M. Nießner, "3d-sis: 3d semantic instance segmentation of rgb-d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4421–4430.

[10] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4096–4105.

[11] Q.-H. Pham, T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8827–8836.

[12] C. Liu and Y. Furukawa, "Masc: multi-scale affinity with sparse convolution for 3d instance segmentation," *arXiv preprint arXiv:1902.04478*, 2019.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[15] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 6409–6418.

[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[21] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4974–4983.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of IEEE International Conference on Computer Vision*, October 2021.

[23] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," *arXiv preprint arXiv:1708.02551*, 2017.

[24] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy, "Semantic instance segmentation via deep metric learning," *arXiv preprint arXiv:1703.10277*, 2017.

[25] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Semi-convolutional operators for instance segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 86–102.

[26] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan, "Proposal-free network for instance-level object segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2978–2991, 2017.

[27] S. Kong and C. C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9018–9028.

[28] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5221–5229.

[29] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.

[30] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9157–9166.

[31] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8573–8581.

[32] J. A. Montoya-Zegarra, J. D. Wegner, K. Schindler *et al.*, "Semantic segmentation of aerial images in urban areas with class-specific higher-order cliques," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 127–133, 2015.

[33] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1715–1724.

[34] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[35] F. Zhang, N. Nauata, and Y. Furukawa, "Conv-mpn: Convolutional message passing neural network for structured outdoor architecture reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2798–2807.

[36] D. Cheng, R. Liao, S. Fidler, and R. Urtasun, "Darnet: Deep active ray network for building segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7431–7439.

[37] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2020.

[38] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2021.

[39] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[40] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4578–4587.

[41] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[42] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[43] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5868–5877.

[44] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in neural information processing systems*, 2018, pp. 820–830.

[45] L. Zhao and W. Tao, "Jsnet: Joint instance and semantic segmentation of 3d point clouds." in *AAAI*, 2020, pp. 12 951–12 958.

[46] M. Li, L. Nan, N. Smith, and P. Wonka, "Reconstructing building mass models from uav images," *Computers & Graphics*, vol. 54, pp. 84–93, 2016.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[48] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *ECCV*. Springer International Publishing, 2014, pp. 703–718.

[49] M. Blha, C. Vogel, A. Richard, J. D. Wegner, T. Pock, and K. Schindler, "Large-scale semantic 3d reconstruction: An adaptive multi-resolution model for multi-class volumetric labeling," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3176–3184.

[50] F. Bernardini and C. L. Bajaj, "Sampling and reconstructing manifolds using alpha-shapes," in *In Proc. 9th Canad. Conf. Comput. Geom*, 1997, p. pages.

[51] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[52] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 157–173, May 2008. [Online]. Available: https://doi.org/10.1007/s11263-007-0090-8

[53] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 152–165, 2014.

[54] G. Can, D. Mantegazza, G. Abbate, S. Chappuis, and A. Giusti, "Semantic segmentation on swiss3dcities: A benchmark study on aerial photogrammetric 3d pointcloud dataset," *Pattern Recognition Letters*, vol. 150, pp. 108–114, 2021.

[55] M. Kölle, D. Laupheimer, S. Schmohl, N. Haala, F. Rottensteiner, J. D. Wegner, and H. Ledoux, "The hessigheim 3d (h3d) benchmark on semantic segmentation of high-resolution 3d point clouds and textured meshes from uav lidar and multi-view-stereo," *ISPRS Open Journal of Photogrammetry and Remote Sensing*, vol. 1, p. 11, 2021.

[56] W. Gao, L. Nan, B. Boom, and H. Ledoux, "Sum: A benchmark dataset of semantic urban meshes," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 179, pp. 108–120, 2021.

[57] Y. Chen, Y. Wang, P. Lu, Y. Chen, and G. Wang, "Large-scale structure from motion with semantic constraints of aerial images," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2018, pp. 347–359.

[58] Y. Liu, F. Xue, and H. Huang, "Urbanscene3d: A large scale urban scene dataset and simulator," 2021.

[59] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[60] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[61] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "Spacenet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.

[62] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xview: Objects in context in overhead imagery," *arXiv preprint arXiv:1802.07856*, 2018.

[63] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[64] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[65] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[66] Y. Y. Boykov and M. . Jolly, "Interactive graph cuts for optimal boundary region segmentation of objects in n-d images," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1, July 2001, pp. 105–112 vol.1.

[67] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[68] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, no. 7, pp. 881–892, 2002.

[69] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[70] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1923–1938, 2018.