Building CAD Model Reconstruction from Point Clouds via Instance Segmentation, Signed Distance Function, and Graph Cut

Takayuki Shinohara
PASCO Corporation
Tokyo Japan
taarkh6651@pasco.co.jp

Li YongHe
Tokyo Japan
yiorn_3951@pasco.co.jp

Mitsuteru Sakamoto
Tokyo Japan
moittu9191@pasco.co.jp

Toshiaki Satoh
Tokyo Japan
tuoost7017@pasco.co.jp

Abstract

Although three-dimensional (3D) modeling of buildings is gaining increasing significance across various real-world applications, the concise representation of buildings from point clouds acquired through unmanned aerial vehicles (UAVs) and other means remains a formidable challenge. In this paper, we introduce an innovative framework for the reconstruction of individual 3D building CAD models derived from point clouds generated by UAV-captured photographs. Our framework encompasses four pivotal components: An instance segmentation model designed to extract buildings from UAV-observed point clouds. Estimation of building surfaces through the utilization of neural networks and the signed distance function of point clouds. Edge estimation based on the inferred building surface. Estimation of building polygons derived from the identified edges. Experimental results obtained from the SPLAT3D dataset affirm the capability of our proposed methodology to generate high-quality building models, thereby offering substantial advantages in terms of accuracy, compactness, and computational efficiency. Furthermore, we demonstrate the robustness of our approach against noise and incomplete measurements, thereby showcasing its applicability to point clouds obtained through photogrammetry utilizing UAV-captured photos.

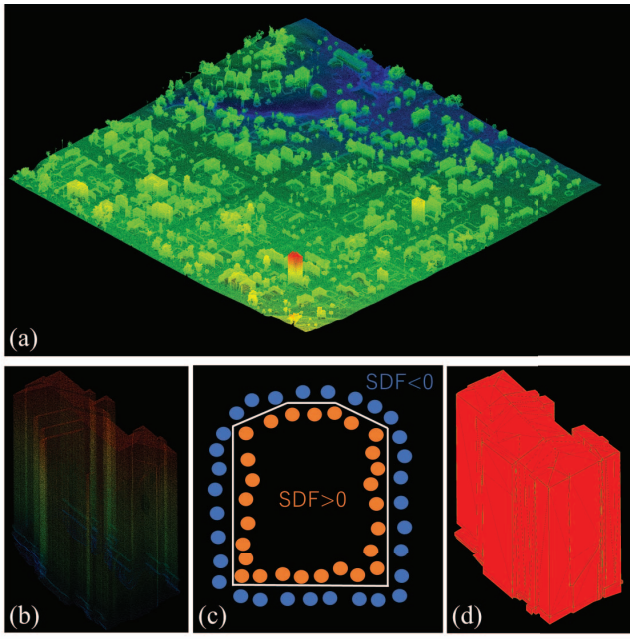

Figure 1. Our approach takes a point cloud as its input and generates three-dimensional (3D) building models as its output. In order to accomplish this, we employ advanced instance segmentation techniques, utilizing semantic class labels and instance labels, to extract buildings (b) from the given point cloud (a). Following that, we utilize signed distance functions (SDF) to reconstruct the surfaces of the buildings (c), enabling us to estimate the three-dimensional representation of the building's surface information (d).

1. Introduction

Three-dimensional (3D) CAD models of buildings have gained paramount importance in a wide array of applications, encompassing urban planning [19], solar potential analysis [32], and noise pollution assessment [40, 1]. The surge in augmented and virtual reality applications has additionally intensified the need for exceptionally detailed and accurate 3D representations of buildings [2]. Despite the many applications, the creation of CAD models relies on manual work, and automatic creation is desired.

In order to reconstruct individual 3D CAD models

of buildings from point clouds observed by LiDAR or SfM/MVS, it is essential to perform building extraction. Extensive research has been conducted on instance segmentation methods to address this requirement. By employing instance segmentation techniques on point clouds, it becomes feasible to discern not only the class label assigned to each point but also the distinct object units representing individual buildings. Consequently, this study incorporates building extraction through the utilization of instance segmentation methodologies.

The majority of existing generic 3D reconstruction methods have been primarily tailored for smooth surfaces represented by dense triangles, often overlooking the inherent piecewise planarity commonly observed in urban environments [24, 16]. In contrast, compact surface models, characterized by a significantly reduced number of faces, possess the ability to effectively capture the geometric intricacies of buildings. While some studies claim that it is possible to reconstruct compact surface models from point clouds [3, 33, 28, 34] or from dense triangle meshes [4, 27], these methods often encounter serious scalability issues. In light of these shortcomings, the primary objective of this study was to develop a robust methodology capable of directly reconstructing compact surfaces of buildings from point clouds.

We capitalize on the inherent capacity of 3D shapes to be implicitly encoded within a function space (implicit function), thereby transcending the limitations imposed by explicit representations such as point clouds, surface meshes, or voxels. Implicit functions have emerged as a prevalent means of describing 3D shapes, with the shape's surface being manifested as the zero-set of a signed distance field (SDF), as originally proposed by Kazhdan et al. [24]. The SDF is formulated through a trainable parameter that discerns whether a given point resides inside or outside the object. While extracting explicit geometry from the SDF can be accomplished through computationally intensive isosurfacing techniques, homogeneous functional representations present distinct advantages in the realm of geometric machine learning, primarily due to their uniform distribution. Notably, a recent learning-based SDF approach by Park et al. [35], termed DeepSDF, harnessed the function space within 3D geometric modeling. The DeepSDF directly learns an implicit function from the input point cloud and generates a smooth surface model of the object. Nevertheless, the extraction of a concise polygonal model from the implicit function remains an ongoing research challenge.

In this paper, we introduce an innovative framework for the extraction of buildings and the subsequent 3D modeling process. Our proposed methodology excels at reconstructing polygonal building meshes that are compact, watertight, and possess an inherent learnable implicit surface representation, thereby facilitating explicit geometry construction. Figure 1 visually demonstrates the underlying principles of our building modeling approach. We consider buildings to be characterized by surfaces that exhibit both smoothness and piecewise planarity, with polygonal surfaces possessing an arbitrary number of sides representing the epitome of compactness. By leveraging a deep neural network to learn an implicit field, specifically guided by neural principles, we extract the building surface from a candidate set that embodies an explicit polyhedral embedding. To address varying levels of surface complexity, we formulate the surface extraction problem as a Markov random field (MRF), affording the capability to efficiently handle such intricacies. Through the judicious application of combinatorial optimization, we further regularize the occupancy of a building inferred from the deep implicit field, while simultaneously penalizing excessive surface complexity. Our reconstruction framework seamlessly integrates the notable strengths of deep implicit field inference, including its inherent efficiency and robustness, with the fidelity and precision associated with reconstruction approaches rooted in primitive assembly-based methodologies.

Compared with current state-of-the-art methods, our 3D modeling framework can obtain high-quality building models with significant advantages in terms of fidelity, compactness, and computational efficiency. The primary contributions of this paper are as follows:

- We propose a 3D building modeling method that combines instance segmentation and SDF.
- We propose a learning-based framework for building extraction and compact building model reconstruction from point clouds. This is the first work to use a deep implicit field for building reconstruction, and our method shows significant performance and quality advantages over state-of-the-art methods, particularly for complex building models.
- We introduce an MRF formulation for surface extraction from the occupancy learned by a neural network. Our formulation allows for complexity control and favors compactness in the final reconstruction and is far more efficient than the existing integer programming formulation.

2. Related Study

2.1. Instance Segmentation

Instance segmentation is a method that assigns class labels to each instance in isolation, rather than simply assigning class labels to individual points as in semantic segmentation[46, 45]. Numerous methodologies have been proposed for 3D instance segmentation, encompassing

bottom-up approaches [44, 43, 14, 26, 29], top-down approaches [20, 47, 47], and more recently, voting-based approaches [8, 15, 17, 22, 42]. MASC [29] employs a multi-scale hierarchical feature backbone, akin to our own; however, the multi-scale features are utilized to calculate pairwise affinities followed by an offline clustering step. Such backbones are also effectively utilized in other domains [12, 37]. DyCo3D [18] is another influential work and is among the few approaches that directly predict instance masks without a subsequent clustering step. DyCo3D relies on *dynamic convolutions* [21, 41], which is similar in concept to our mask prediction mechanism. However, it does not utilize optimal supervision assignments during training, leading to subpar performance. Optimal assignment of the supervision signal was first implemented by 3D-BoNet [47] using Hungarian matching. Similarly to our approach, D-BoNet predicts all instances in parallel. However, it utilizes only a single-scale scene descriptor, which cannot encode object masks of diverse sizes. The authors of [23] proposed a method called PointGroup, which can segment objects from both original and offset-shifted point sets. Their algorithm uses a simple yet effective technique that groups nearby points with the same label and expands the group progressively. Chen et al. [9] have extended the PointGroup method to develop HAIS, which further incorporates surrounding fragments of instances and refines them based on intra-instance prediction. In our work, we utilize the HAIS architecture as an initialization step for building extraction from point clouds.

2.2. 3D modeling

Recent advancements in developing deep learning-based implicit functions have demonstrated their potential for 3D reconstruction. These methods rely on learning a mapping from an input (such as a point cloud) to a continuous scalar field and then extracting the surface of the object using iso-surfacing techniques such as Marching Cubes [30]. While iso-surfacing is powerful in extracting smooth surfaces, it is limited in preserving sharp features and introduces discretization errors, creating deep implicit fields unsuitable for compact polygonal model reconstruction.

To address this limitation, researchers have incorporated constructed solid geometry (CSG) [11] into their methods. One such example is the end-to-end neural network BSP-Net [11], which reconstructs a shape from a set of convexes obtained via binary space partitioning. Similarly, CVXNet [13] is an architecture that represents a low-dimensional family of convexes. These methods learn to divide and conquer 3D space with implicit function; however, their inputs are images or voxels, unlike the point clouds our work addresses.

Most deep learning-based implicit function methods use a single latent feature vector, which imposes strong priors dependent on the training data, limiting their generalization ability. While this allows plausible surface reconstruction even with highly contaminated data, the feature space inevitably overfits the shapes in the training set, which may fail for shapes from unseen categories. The Points2Surf [16] architecture addresses this limitation by estimating a signed distance function (SDF) with both local and global feature vectors, demonstrating outstanding generalization capabilities in implicit field learning. In our work, we utilize the Points2Surf architecture as an initialization step for 3D reconstruction.

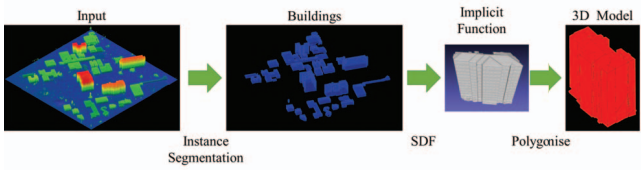

Figure 2. Our workflow of instance segmentation and neural-guided reconstruction workflow.

3. Proposed Method

3.1. workflow

Our proposed method follows a three-step workflow, as illustrated in Fig. 2. We perform instance segmentation on the point cloud to extract a point cloud of buildings, which is then used to create a 3D model of individual buildings. And then we used a pre-trained SDF network to obtain surface information of each building point cloud. Finally, we make a 3D building model from surface information using a Graph Cut optimization process.

1. **Instance Segmentation**. To begin, instance segmentation is utilized to extract point clouds of individual buildings from a larger point cloud obtained from photographs taken by a UAV. Instance segmentation is a technique that separates and instantiates each object within the input point cloud while also classifying the instance's category. For this purpose, we employ HAIS [9], which is a widely used approach for point cloud instance segmentation.

2. **SDF**. Next, we reconstruct the building's surface from the point cloud by utilizing a signed distance function (SDF) as an implicit function representation. First, we generate a linear cell complex that conforms to the planar primitives detected from the point cloud, thus partitioning the ambient 3D space. This adaptive partitioning is both efficient and respects the geometry of the building. The non-overlapping cells in the complex are used as candidates, whose outer shell forms the final surface. To learn a shape-conditioned implicit field that characterizes the building object represented

by the point cloud, we employ a deep neural network. The implicit field describes the object's spatial occupancy, providing a binary decision boundary for any query point in the 3D space.

3. **3D Modeling**. This step takes the learned implicit field as an occupancy indicator and outputs the boundary representation of the building's surface. We formulate surface extraction as a binary classification problem and solve it using MRF optimization that encourages compactness and guarantees the final model is watertight.

3.2. Instance Segmentation

In this study, we utilize the point cloud instance segmentation technique, HAIS [6], to extract buildings. The architecture of HAIS [6] comprises four principal components, as shown in Fig. 3. The first component is the point-wise semantic label prediction network, which extracts features from point clouds and predicts semantic labels and center shift vectors for each point. The second component is the point aggregation module, which generates initial instance predictions based on point-wise predictions. The third component is the set aggregation module, which expands incomplete instances to cover missing parts, and the fourth component is the intra-instance prediction network, which smooths instances to filter out outliers.

point-wise semantic label prediction. The term "semantic label" refers to a classification assigned to a point cloud based on its characteristics. In this study, the point cloud features obtained from the 3D UNet are fed into a two-layer Multi-Layer Perceptron(MLP), and the cross-entropy loss function is employed to classify individual points into their respective classes. It should be noted that this study focuses on the extraction of buildings, and the classification task involves differentiating between buildings and non-buildings.

center shift-vector. It is a common practice in point cloud instance segmentation to determine the shift vector or direction of the current voxel or point cloud towards the center of its corresponding instance. This technique helps to enhance the clustering effect and facilitates the network's ability to learn shape features. Similarly, in the present study, the point cloud features derived from the 3D U-Net model are also passed through a two-layer MLP to obtain the offset vectors. These offset vectors are then supervised using a loss function. Initially, only the offset vectors of the antecedent point clouds are taken into consideration, while the background point clouds are ignored. Moreover, each point cloud offset vector is assigned a different weight. This is because, when a point cloud is situated near the center of an instance, predicting the offset vector accurately becomes crucial. Conversely, predicting the offset vector accurately for distant point clouds, such as those at the instance boundary, is significantly more difficult and hence incurs a larger penalty. The weight assigned to each offset vector, denoted by w is calculated based on the true value of the offset vector. During training, $\mathcal{L}_{\text{shift}}$ is used to optimize the center shift vector prediction, which is formulated as

$$\begin{aligned} \mathcal{L}_{\text{shift}} &= \frac{1}{\sum_{p_i \in P} \mathbb{1}(p_i \in P_{\text{fg}})} \cdot \sum_{p_i \in P} \mathcal{L}(p_i), \\ \mathcal{L}(p_i) &= w(p_i) \cdot \parallel \triangle x_i^{\text{gt}} - \triangle x_i^{\text{pred}} \parallel_1 \cdot \mathbb{1}(p_i \in P_{\text{fg}}), \\ w(p_i) &= \min(\parallel \triangle x_i^{\text{gt}} \parallel_2, 1). \end{aligned} \tag{1}$$

$\mathbb{1}(\cdot)$ is the indicator function. P is the whole point set and P_{fg} is the foreground point set respectively. Background points are ignored in $\mathcal{L}_{\text{shift}}$. $w(p_i)$ operates as a point-wise weighted term. Points closer to the instance center rely less on the center shift vectors and should contribute less to the loss.

point aggregation module. Upon obtaining the aforementioned two features, we proceed to the initial aggregation step, wherein we first shift the point coordinates to bring them closer to the center of the instance, thereby facilitating subsequent aggregation. Next, we establish a threshold value: when the semantic label of two points is of the same class, and the distance between them after offset is less than this threshold, we initially consider these points to be part of the same instance, and connect them using an edge. This leads us to the following scenario: Fragments generally refer to points located on the boundary of instances that are challenging to partition.

set aggregation module. When both fragments and primary instances share the same semantic label and the distance between them is less than a certain value, the initial notion is to establish a threshold that allows them to be merged together. However, because there are many fragments belonging to different instances, this approach can be too drastic. As an alternative, the HAIS approach implements a flexible threshold. Specifically, the maximum value of "a" and "b" is taken as the threshold. The value of "a" means that the larger the instance, the further away its fragments may be. A tangible example of this is the gravitational pull of a planet, which can attract objects that are farther away from it. The value of "b" represents the category size count, as an instance's size is typically associated with its category. By estimating this, one can approximate the size of an instance and set an appropriate threshold value "r". By taking the maximum value of "a" and "r" a dynamic threshold can be obtained, which is used for further aggregation.

intra-instance prediction network Despite the series of steps outlined above, instances may still absorb incorrect fragments. To address this, instances are first divided into smaller pieces, and features are extracted through 3D convolution. This is followed by a segmentation using a common classification loss function. It is worth noting that only samples with real instances having a mask Intercect over Union (IoU) greater than 0.5 are obtained at this stage. The subsequent step involves confidence prediction, where the confidence score reflects the credibility of the instance. One simple approach to constrain confidence is to assign higher values to instances with greater IoU. For this masking process, the loss is formulated as,

$$\mathcal{L}_{\text{mask}} = -\frac{1}{\sum_{i=1}^{N_{\text{ins}}} \mathbb{1}(\text{iou}_i > 0.5) \cdot N_i} \cdot \sum_{i=1}^{N_{\text{ins}}} \Big\{ \mathbb{1}(\text{iou}_i > 0.5) \cdot \sum_{j=1}^{N_i} \big[y_j \cdot \log(\hat{y}_j) + (1 - y_j) \cdot \log(1 - \hat{y}_j) \big] \Big\}, \tag{2}$$

where N_{ins} represents the number of instances and N_i denotes the point number of instance i. Furthermore, score loss is employed to suppress over-detection by evaluating the plausibility of each estimated instance region as well as the masking process. This score loss is formulated as,

$$\mathcal{L}_{\text{score}} = -\frac{1}{N_{\text{ins}}} \cdot \sum_{i=1}^{N_{\text{ins}}} \big[\text{iou}_i \cdot \log(\hat{s}_i) + (1 - \text{iou}_i) \cdot \log(1 - \hat{s}_i) \big]. \tag{3}$$

loss function The whole network is trained from scratch in an end-to-end manner and optimized by a joint loss consisting of several loss terms,

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{shift}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{score}}, \tag{4}$$

where $\mathcal{L}_{\text{seg}}$ is the cross-entropy loss of semantic scores, and $\mathcal{L}_{\text{shift}}$, $\mathcal{L}_{\text{mask}}$ and $\mathcal{L}_{\text{score}}$ are defined in Eq. 1, 2 and 3 respectively.

3.3. SDF

A signed distance field (SDF) is utilized for estimating the surface of each instance of a point cloud representing buildings. In this study, Point2Surf [16] is employed as the SDF method.

The goal of surface estimation is to extract a subset of cells, denoted by $L \in C$, from the cell complex obtained through adaptive binary space partitioning, such that its occupancy O_L represents the interior space enclosed by the outer surface of the building. To achieve this, the occupancy of the building is learned as an SDF, where the value is the distance d from a point $\mathbf{x}$ to the building surface:

$$\text{SDF}(\mathbf{x}) = d : \mathbf{x} \in \mathbb{R}^3, d \in \mathbb{R}. \tag{5}$$

The sign of the value indicates whether the point lies inside (with a positive sign) or outside (with a negative sign) of the surface of the building.

Taking inspiration from the work on points-to-surface mapping [16], we employ a deep learning-based approach to acquire knowledge of the signed distance field from a given point cloud. More specifically, we develop a neural network that can estimate the signed distance value for any point $\mathbf{x} \in \mathbb{R}^3$ as follows:

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = s_\theta(\mathbf{x} \mid \mathbf{z}), \text{ with } \mathbf{z} = e_\phi(P). \tag{6}$$

Here, $\mathbf{z}$ corresponds to a latent representation of the building surface, encoded from the input point cloud P using an encoder e, and s represents the neural network. The encoder e and neural network s are parameterized by θ and ϕ, respectively. Following the neural network architecture of Point2Surf [16] for points-to-surface mapping, we decompose the signed distance field into two components: the absolute distance f^d and its sign f^s.

- **The absolute distance value**. The estimated absolute distance $\tilde{f}^d(\mathbf{x})$ can be obtained solely from the local neighborhood of the query point, as expressed by the following equation:

$$\tilde{f}^d(\mathbf{x}) = s_\theta^d\left(\mathbf{x} \mid \mathbf{z}_{\mathbf{x}}^d\right), \text{ with } \mathbf{z}_{\mathbf{x}}^d = e_\phi^d\left(\mathbf{p}_{\mathbf{x}}^d\right). \tag{7}$$

  Here, $\mathbf{p}_{\mathbf{x}}^d \in P$ refers to a set of neighboring points surrounding the query point $\mathbf{x}$.

- **The sign**. To estimate the sign $\tilde{f}^s(\mathbf{x})$ at point $\mathbf{x}$, relying on local sampling alone is insufficient because the occupancy information cannot be accurately estimated from the local neighborhood. Instead, a global uniform sub-sample $\mathbf{p}_{\mathbf{x}}^s \in P$ is taken as input, and the sign is estimated as follows:

$$\tilde{f}^s(\mathbf{x}) = \text{sgn}\left(\tilde{g}^s(\mathbf{x})\right) = \text{sgn}\left(s_\theta^s\left(\mathbf{x} \mid \mathbf{z}_{\mathbf{x}}^s\right)\right), \text{ with } \mathbf{z}_{\mathbf{x}}^s = e_\psi^s\left(\mathbf{p}_{\mathbf{x}}^s\right). \tag{8}$$

  Here, ψ is used to parameterize the encoder and $\tilde{g}^s(\mathbf{x})$ represents the logit expressing the confidence of point $\mathbf{x}$ being at a positive distance from the surface.

The two latent representations, $\mathbf{z}_{\mathbf{x}}^s$ and $\mathbf{z}_{\mathbf{x}}^d$, share information to formulate signed distance learning as follows:

$$\left(\tilde{f}^d(\mathbf{x}), \tilde{g}^s(\mathbf{x})\right) = s_\theta\left(\mathbf{x} \mid \mathbf{z}_{\mathbf{x}}^d, \mathbf{z}_{\mathbf{x}}^s\right), \text{ with } \mathbf{z}_{\mathbf{x}}^d = e_\phi^d\left(\mathbf{p}_{\mathbf{x}}^d\right) \text{ and } \mathbf{z}_{\mathbf{x}}^s = e_\psi^s\left(\mathbf{p}_{\mathbf{x}}^s\right) \tag{9}$$

This results in a joint prediction of the signed distance function $\tilde{f}^d(\mathbf{x})$ and the confidence of $\mathbf{x}$ having a positive distance to the surface, expressed as $\tilde{g}^s(\mathbf{x})$.

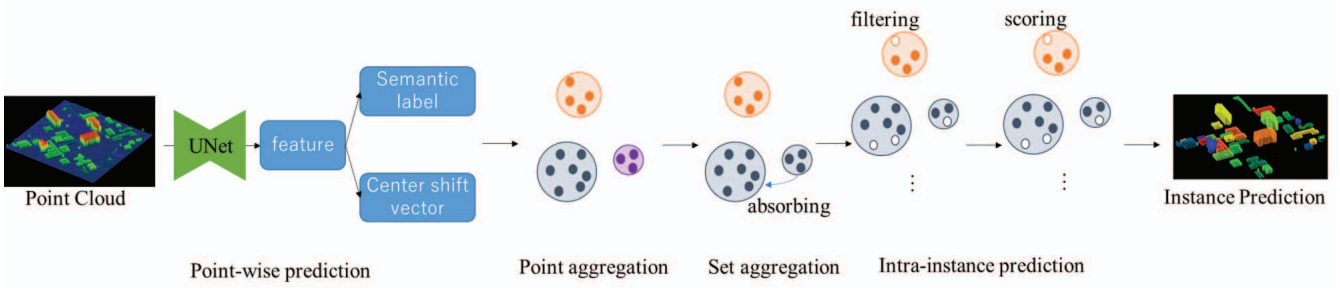

Figure 3. The HAIS framework, as described in [9], consists of several stages. First, the input point cloud undergoes point-wise feature learning via a 3D UNet-like structure with submanifold sparse convolution. Next, HAIS utilizes the spatial constraint of points to perform point aggregation with fixed bandwidth. Based on the results of this point aggregation, a set aggregation with dynamic bandwidth is performed to generate instance proposals. Finally, an intra-instance prediction is implemented for outlier filtering and mask quality scoring.

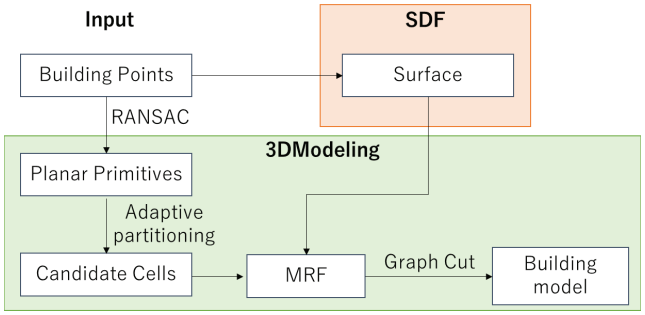

Figure 4. Modeling method using SDF(Point2Surf[16]) and making 3D model by polygonize process.

3.4. 3D modeling

To estimate the building surface based on the occupancy map obtained from SDF, we refer to the work by Chen et al. [10], who proposed a method called Point2Poly. Point2Poly utilizes adaptive space partitioning and surface extraction techniques to generate a surface representation from the occupancy map.

Space partitioning. We present an algorithm for adaptive partitioning that subdivides the 3D space into a set of cells, each of which is a convex polyhedron. Our method extracts planes from the point cloud and applies a refined planar primitive using BSP to partition the space. To detect planes, we employ the RANSAC algorithm described in [38]. We also perform a refinement procedure that iteratively merges planes under specific proximity conditions to account for noise and outliers in the data. Our algorithm dynamically and locally updates a binary tree structure during the partitioning process and maintains cell adjacency information.

Surface extraction. After obtaining the cell complex and the occupancy information of its cells, the task of surface reconstruction involves obtaining a consistent classification of the cells into the categories of *interior* and *exterior*, followed by an outer shell extraction step. To accomplish this, we propose the use of a Markov random field (MRF) formulation for interior/exterior cell classification.

We represent the cell complex $C = c_i$ and denote the binary label assigned to a cell c_i as $x_i \in \text{interior}, \text{exterior}$. Our energy function is expressed as a weighted sum of two energy terms, i.e.,

$$E(\mathbf{x}) = D(\mathbf{x}) + \lambda V(\mathbf{x}), \tag{10}$$

where $D(\mathbf{x})$ and $V(\mathbf{x})$ are the data cost term and the smoothness cost term, respectively, and λ is the weight parameter that balances the two terms.

- **Data cost**. We present the definition of the data cost term such that it accurately reflects the classification confidence. Specifically, we define it as the measurement of the deviation between the classification and the previously estimated cell occupancy in the cell complex, i.e.,

$$D(\mathbf{x}) = \frac{1}{|C|} \sum_{c_i \in C} \left| x_i - \text{occu}(c_i) \right|, \tag{11}$$

  where $\text{occu}(c_i)$ refers to the learned occupancy of the cell c_i. We compute $\text{occu}(c_i)$ as:

$$\text{occu}(c_i) = \text{sigmoid}(\text{SDF}(c_i) \cdot \text{vol}(c_i)), \tag{12}$$

  where $\text{SDF}(c_i)$ denotes the signed distance value of the query point at the centroid of c_i, predicted by the neural network, and $\text{vol}(c_i)$ is the volume of c_i. We assign higher weights to the cells with larger volumes, regardless of their predicted signed distance. The sigmoid function $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ normalizes the signed distance to the range $(0, 1)$.

- **Smoothness cost**. We introduce an energy term that promotes the assignment of similar labels to adjacent cells. To ensure that our building surface model remains simple, we have designed this term to penalize

its complexity. Because the final surface is extracted from the outer shell of the interior cells, reducing its complexity is equivalent to limiting its surface area. Therefore, we define our smoothness cost as

$$V(\mathbf{x}) = \frac{1}{A} \sum_{\{c_i, c_j\} \in C} a_{ij} \cdot \mathbb{1}(c_i, c_j), \tag{13}$$

where $c_i, c_j \in C$ represents a pair of adjacent cells in the complex. Here, a_{ij} denotes the surface area of the common face of the two cells. We choose A as a normalization factor, which is equal to the maximum area of all faces in the cell complex. The indicator function $\mathbb{1}(c_i, c_j)$ takes a value of 1 if c_i and c_j receive different labels, i.e.,

$$\mathbb{1}(c_i, c_j) = \begin{cases} 0, & x_i = x_j \\ 1, & x_i \neq x_j \end{cases} \tag{14}$$

In essence, the smoothness cost acts as a regularization term, penalizing any zigzag artifacts that may appear on the final surface model.

By minimizing the energy function as specified in Equation 10 using the graph cut algorithm [5], one can obtain the interior cells. The final surface model can then be obtained by extracting the outer surface of the union of these interior cells. Because our adaptive binary space partitioning technique produces a valid polyhedral embedding, the resulting surface is inherently guaranteed to be watertight.

4. Experimental Results

4.1. Experimental Settings

Datasets. Experiments were conducted on a standard benchmark dataset called *STPLS3D* [7]. STPLS3D is a synthetic outdoor dataset closely mimicking the data generation process of aerial photogrammetry point clouds. Twenty-five urban scenes totaling 6 km^2 are densely annotated with 14 instance classes. We followed the common splits [7, 42] in the training and test phases.

Implementation Details. The implementation details adhered to those of established methodologies [23, 9]. The model was developed utilizing the PyTorch deep learning framework [36] and was trained for 120k iterations using the Adam optimizer [25]. The batch size was set to 4, and the learning rate was initialized to 0.001, then scheduled by cosine annealing [31]. The voxel size and grouping bandwidth b were set to 0.02 m and 0.04 m, respectively, while the score threshold for soft grouping τ was set to 0.2. At inference, the whole scene is fed into the network without cropping. For the SPLAT3D dataset with high point density, scenes are randomly downsampled at a ratio of 1/4 before cropping. At inference, the scene is divided into four parts before feeding into the model, and then the outputs from the four parts are merged to get the final results.

4.2. Instance Segmentation.

To assess the instance-level performance of our proposed method, we evaluated it on SPLAT3D data; the resulting extraction outputs for various types of buildings are presented in Fig. 5. This figure provides visual evidence of the effectiveness of our hierarchical aggregation and intra-instance prediction techniques, particularly for objects with large sizes and fragmentary point clouds, where grouping all points together presents a significant challenge. Our proposed approach addresses this issue by producing precise instance segmentation masks. The trained model extracted 0.91 for the mAP and 0.88 for the trained Mask3D [39]. In the extraction results for building point clouds, the mAP is comparable to that of Mask3D [39], the current best performance. In this experiment, the performance of our model was higher than applying Mask3D's trained model for multi-class object detection to extract only buildings.

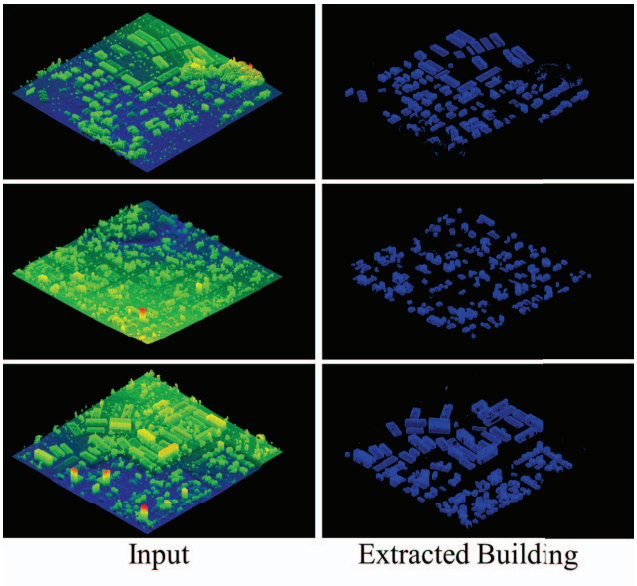

Figure 5. Building extraction results on the test dataset.

4.3. 3D modeling

Using the neural network trained on the building point clouds extracted by HAIS from the *SPLAT3D* dataset, our proposed method can reconstruct buildings of various architectural styles, as demonstrated by the results presented in Fig. 6. The signed distance function (SDF) and graph-cut-based modeling method accurately characterize the simple shape of the buildings. However, errors can arise when dealing with subtle structures, which can be attributed to

uncertainty in planar primitive detection and regularization imposed on surface extraction.

To address the issue of noise in actual point cloud data, a 3D model was reconstructed using a learned points2surf-based SDF from a synthetic point cloud with intentionally varied noise levels, as shown in Fig. 7 Specifically, to evaluate the robustness of the learned model to noise, Gaussian noise was applied to each point, and in doing so, point clouds with three different Gaussian noise levels were created and tested. The SDF model method was trained on point clouds with low noise levels in the $[0, 0.001R]$ range, but reconstructed reliably for point clouds with significantly higher noise levels up to $0.005R$ (see Fig. 7 Weak Noise). This corresponds to a measurement error of as much as 0.5 m on a 100 m square building. However, point clouds of $0.01R$ (Fig. 7 middle noise) or $0.03R$ (Fig. 7 strong noise) are not sufficient for modeling. Our research is focused on assembling the initially detected planar primitives into a compact polygonal building model, rather than detecting the planar primitives themselves. Therefore, we assume that the dominant planes, such as walls and roofs, can be identified from the input point cloud. However, this may not always be feasible when dealing with noisy or incomplete scans. Several steps have been devised to reduce inaccuracies due to primitive detection, such as primitive refinement and MRF complexity, but these may still be insufficient when the provided primitives are incomplete or contain large errors.

Figure 6. Example of 3D models from our method.

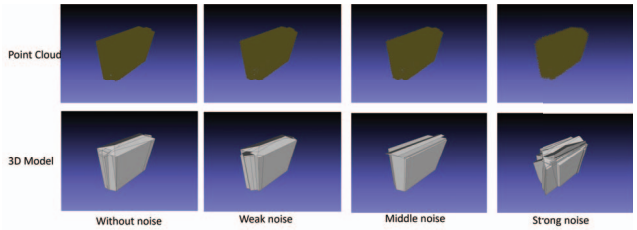

Figure 7. Robustness to noise. Note that our neural network was trained on point clouds without noise, and at the time of inference, we have fed the point cloud into the trained model with different noise levels $[0, 0.005R]$, where R denotes the radius of the bounding sphere of the input point cloud.

5. Conclusion

We introduced a pioneering and effective approach for the reconstruction of urban buildings by combining instance segmentation and an implicit representation learned as an occupancy indicator for explicit geometry extraction. Through our novel occupancy learning strategy and the use of the MRF formulation, we demonstrated that our method produces high-quality building models with notable benefits in terms of accuracy, compactness, and computational efficiency. To our best knowledge, this is the first study to explore the use of a deep implicit field for building reconstruction from UAV-obtained point clouds.

In our future work, we aim to enhance our current method by integrating the construction of explicit geometry into a neural network, creating an end-to-end pipeline. Furthermore, we intend to incorporate user interactions into the reconstruction pipeline to improve the usability of the method in challenging scenarios. Although our MRF-based surface extraction technique is efficient enough to allow for interactive editing, we plan to further refine our approach by ensuring that all operations related to building modeling are differentiable, and by training instance segmentation and SDF end-to-end. It should be noted that our building extraction and 3D modeling pipeline consist of distinct deep learning models, and as such, the results of the 3D modeling of the building are not reflected in the instance segmentation results during deep learning model training. Hence, in the future, we plan to integrate all building modeling operations and make them differentiable for seamless end-to-end training of the instance segmentation and SDF models.

Acknowledgements

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used. We thank STPLS3D and Helsinki 3D city models for providing dataset. We also thank the reviewers.

References

[1] Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2220–9964, 2015.

[2] Christoph Blut and Jörg Blankenbach. Three-dimensional CityGML building models in mobile augmented reality: a smartphone-based pose tracking system. *International Journal of Digital Earth*, 14(1):32–51, 2021.

[3] Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. Piecewise-planar 3D reconstruction with edge and corner regularization. In *Computer Graphics Forum*, volume 33, pages 55–64. Wiley Online Library, 2014.

[4] Vasileios Bouzas, Hugo Ledoux, and Liangliang Nan. Structure-aware building mesh polygonization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167:432–442, 2020.

[5] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient ND image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[6] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 392–401, 2020.

[7] Meida Chen, Qingyong Hu, Thomas Hugues, Andrew Feng, Yu Hou, Kyle McCullough, and Lucio Soibelman. STPLS3D: A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset. *arXiv:2203.09065*, 2022.

[8] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical Aggregation for 3D Instance Segmentation. In *ICCV*, 2021.

[9] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *ICCV*, 2021.

[10] Zhaiyu Chen, Hugo Ledoux, Seyran Khademi, and Liangliang Nan. Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing*, 194:58–73, 2022.

[11] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. BSP-Net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020.

[12] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *CVPR*, 2022.

[13] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. CvxNet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31–44, 2020.

[14] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3d bird's-eye-view instance segmentation. In *Pattern Recognition: 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10–13, 2019, Proceedings 41*, pages 48–61. Springer, 2019.

[15] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *CVPR*, 2020.

[16] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2Surf: Learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, pages 108–124. Springer, 2020.

[17] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D Instance Segmentation. In *CVPR*, 2020.

[18] Tong He, Chunhua Shen, and Anton van den Hengel. DyCo3D: Robust Instance Segmentation of 3D Point Clouds through Dynamic Convolution. In *CVPR*, 2021.

[19] Grant Herbert and Xuwei Chen. A comparison of usefulness of 2D and 3D representations of urban planning. *Cartography and Geographic Information Science*, 42(1):22–32, 2015.

[20] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *CVPR*, 2019.

[21] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic Filter Networks. *NeurIPS*, 2016.

[22] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *CVPR*, 2020.

[23] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020.

[24] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICVLR*, 2015.

[26] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3D Instance Segmentation via Multi-Task Metric Learning. In *ICCV*, 2019.

[27] Minglei Li and Liangliang Nan. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:135–150, 2021.

[28] Minglei Li, Liangliang Nan, Neil Smith, and Peter Wonka. Reconstructing building mass models from UAV images. *Computers & Graphics*, 54:84–93, 2016.

[29] Chen Liu and Yasutaka Furukawa. MASC: Multi-Scale Affinity with Sparse Convolution for 3D Instance Segmentation. *arXiv:1902.04478*, 2019.

[30] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

[31] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICVLR*, 2017.

[32] Rita Machete, Ana Paula Falcão, M Glória Gomes, and A Moret Rodrigues. The use of 3D GIS to analyse the influence of urban context on buildings' solar energy potential. *Energy and Buildings*, 177:290–302, 2018.

[33] Claudio Mura, Oliver Mattausch, and Renato Pajarola. Piecewise-planar reconstruction of multi-room interiors with arbitrary wall arrangements. In *Computer Graphics Forum*, volume 35, pages 179–188. Wiley Online Library, 2016.

[34] Liangliang Nan and Peter Wonka. PolyFit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2353–2361, 2017.

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[37] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. FCAF3D: Fully Convolutional Anchor-Free 3D Object Detection. *arXiv:2112.00322*, 2021.

[38] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer Graphics Forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[39] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. 2023.

[40] Jantien Stoter, Ravi Peters, Tom Commandeur, Balazs Dukai, Kavisha Kumar, and Hugo Ledoux. Automated reconstruction of 3D input data for noise simulation. *Computers, Environment and Urban Systems*, 80:101424, 2020.

[41] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional Convolutions for Instance Segmentation. In *ECCV*, 2020.

[42] Thang Vu, Kookhoi Kim, Tung M Luu, Xuan Thanh Nguyen, and Chang D Yoo. SoftGroup for 3D Instance Segmentation on Point Clouds. In *CVPR*, 2022.

[43] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *CVPR*, 2018.

[44] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively Segmenting Instances and Semantics in Point Clouds. In *CVPR*, 2019.

[45] Haoyi Xiu, XIN LIU, Weimin Wang, Kyoung-Sook Kim, Takayuki Shinohara, Qiong Chang, and Masashi Matsuoka. Diffusion unit: Interpretable edge enhancement and suppression learning for 3d point cloud segmentation. *Available at SSRN 4346396*.

[46] Haoyi Xiu, Takayuki Shinohara, and Masashi Matsuoka. Dynamic-scale graph convolutional network for semantic segmentation of 3d point cloud. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 271–2717, 2019.

[47] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In *NeurIPS*, 2019.