SUM Parts: Benchmarking Part-Level Semantic Segmentation of Urban Meshes

Supplementary Material

6. Details on annotation tool

6.1. Face-based annotation

Protrusion score. By measuring the distance and angle from face f_i to support plane P_k^f, we define protrusion score $p_i = d_i + \omega_i \cdot \theta_i$, where

$$d_i = \max_{t \in \{0,1,2\}} \left(\text{dist}(v_{t,i}, P_k^f) \right)$$

is the maximum Euclidean distance from the vertices v_t of the face f_i to the support plane P_k^f. The angle weight θ_i is calculated by measuring the angle $\hat{\theta}_i = \cos^{-1}(\mathbf{n}_i \cdot \mathbf{n}_k)$ between the normal $\mathbf{n}_i$ of face f_i and the normal $\mathbf{n}_k$ of support planar segment P_k^f, defined as:

$$\theta_i = \frac{\min(\hat{\theta}_i, 180° - \hat{\theta}_i)}{90°}.$$

Geometric consistency. To measure the geometric consistency between adjacent faces, we utilize an interior shrinking ball algorithm derived from the 3D medial axis transform to compute the ball radii for each face [46, 56]. In urban mesh scenarios, larger shrinking balls typically correspond to major geometric structures such as the terrain or main surfaces of buildings, whereas smaller balls indicate sharp structures or protrusions (as shown in Fig. 15). Consequently, the size of these balls can indirectly reflect the local structural scale, suggesting that adjacent faces within the same geometric structure should have similar radii. The mesh shrinking ball radius is derived as

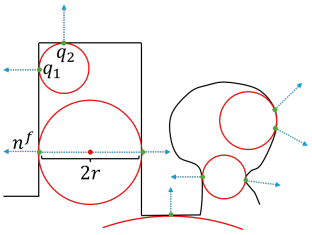

Figure 15. Cross-sectional view of interior shrinking balls (red) in urban scenarios.

$$r = \frac{\|q_1 - q_2\|^2}{2(\mathbf{n}^f \cdot (q_1 - q_2))},$$

where r refers to the radius r_i or r_j, and $\mathbf{n}^f$ denotes the normal $\mathbf{n}_i$ or $\mathbf{n}_j$ of the respective faces f_i or f_j; q_1 and q_2 are the tangent points on the faces.

Planar segment matching. We define the feature vector $\mathbf{F}^{(\mathbf{seg})}$ to quantify segment matching similarity, including:

- Geometric homogeneity: Differences in area between geometrically similar segments are calculated as:

$$\Delta A^{(seg)} = \frac{\left| area^{(c)} - area^{(t)} \right|}{area^{(t)}},$$

  where $area^{(c)}$ and $area^{(t)}$ are the areas of the candidate and template segments, respectively.
- Spatial distribution: Vertical distribution similarity is measured by comparing weighted average heights:

$$\Delta H^{(seg)} = \left| \frac{\sum_{i=1}^{m'} z_i \cdot a_i}{area^{(c)}} - \frac{\sum_{j=1}^{m''} z_j \cdot a_j}{area^{(t)}} \right|.$$

  where z_i and a_i denote the z-coordinate and area of each face f_i in the candidate segment $P_k^{(c)}$. z_j and a_j denote those in the template segment $P^{(t)}$.
- Spatial orientation: Similarity in vertical orientation is assessed between segments $P_k^{(c)}$ and $P^{(t)}$ [51].
- Shape sphericity: Calculated using eigenvalues from triangle vertices of the segment $P_k^{(c)}$ and $P^{(t)}$ to evaluate similarity [59].
- Photometric coherence: Color similarity is assessed using CIELAB [38] color distance and greenness [41] difference.

Protrusion matching. For seed expansion, in addition to spatial and segment scale constraints, we introduce optional topology constraints based on adjacency to optimize user focus and simplify inspection. In urban scenes, small protrusions (e.g., cars, balconies, dormers) reduce global matching efficiency by increasing inspection workload and computation (e.g., matching cars globally takes 5s, whereas planar facades take only 0.4s, see Fig. 4b). Therefore, we set topology constraints as the default for practical efficiency by confining the search space to planar segments of the template support surface (e.g., limiting annotations to the current facade for facade installations).
The feature vector $\mathbf{F}^{(\mathbf{str})}$ includes:

- Spatial compactness: The compactness of a protrusion is quantified by considering its volume. We expect similar protrusions to have comparable values, defined as

$$\Delta V^{(str)} = \left| \frac{vol^{(c)}}{vol_{box}^{(c)}} - \frac{vol^{(t)}}{vol_{box}^{(t)}} \right|,$$

  where $vol^{(c)}$ and $vol_{box}^{(c)}$ represent the volume of $f^{(c)}$ and its bounding box volume, respectively. $vol^{(t)}$ and $vol_{box}^{(t)}$ are the corresponding values for $f^{(t)}$.

- Surface complexity: We assume complex 3D shapes decompose into multiple planar segments. Surface complexity similarity is measured by the ratio of the number of planar segments in the template and candidate protrusions, defined as

$$\Delta N^{(str)} = \left(\frac{\max(n^{(t)}, n^{(c)})}{\min(n^{(t)}, n^{(c)})} \right)^{\mu},$$

  where $n^{(t)}$ and $n^{(c)}$ respectively represent the number of planar segments for the template and candidate protrusions, and $\mu = \min(n^{(t)}, n^{(c)})$.
- Structural features: Measuring the similarity of protrusions involves comparing their structural features through eigenvalue analysis including linearity, planarity, and sphericity [59]. We determine similarity by the ℓ_1 distance in the feature space, including differences in linearity $\Delta L^{(str)}$, planarity $\Delta P^{(str)}$, and sphericity $\Delta S^{(str)}$.

6.2. Texture-based annotation

Gaussian mixture model (GMM). G_k denotes the GMM for the k-th channel, defined as

$$G_k(S) = \sum_{m=1}^{M} \pi_{km} \mathcal{N}(x; \mu_{km}, \Sigma_{km}),$$

where S represents the superpixel S_0 or S_j. x is a pixel sample point of S, and M is the number of components in GMM (M set to 5 in all experiments in this paper). $\mathcal{N}(s; \mu, \Sigma)$ denotes the multivariate normal distribution, with μ representing the mean for superpixels S_0 or S_j, and Σ denotes their respective covariance matrices.

Local color consistency. For local color consistency, where $\rho_j = \Delta E_{00}(U_0, U_j)$ is the color distance (i.e., CIEDE2000 [38]) from the superpixel S_j to its seed S_0. To more accurately capture the intrinsic structure and variability within superpixels' color distributions, we employ a GMM to compute the average Lab color, represented by $U = \sum_{m=1}^{M} \pi_m \mu_m$, where U represents U_0 or U_j, with π_m as the mixing weight and μ_m as the mean for the m-th Gaussian component in the Lab color space. Additionally, seed samples for U_0 are taken from its first-order neighborhood, whereas samples for U_j come from its own pixels.

Region-based template matching. The feature vector $\mathbf{F}^{(\mathbf{reg})}$ includes:

- Shape index: To assess shape similarity, we use a shape index reflecting elongation or flatness, which is defined as: $r = \frac{\min(w,h)}{\max(w,h)}$, where w and h represent the width and height of the object's bounding box, respectively. The similarity between regions is calculated as: $\Delta I^{(reg)} = |r_c - r_t|$, where r represents the ratio r_c of the candidate region or r_t of the template region.
- Shape regularity: We assess shape regularity to calculate the similarity between areas. Similar to structural matching, compactness is used to describe how well a shape fills its bounding box, defined as:

$$\Delta A^{(reg)} = \left| \frac{area^{(c)}}{area^{(c)}_{box}} - \frac{area^{(t)}}{area^{(t)}_{box}} \right|,$$

  where $area^{(c)}$ and $area^{(t)}$ represent the area of the candidate and template regions, respectively, and $area^{(c)}_{box}$ and $area^{(t)}_{box}$ are the areas of their bounding boxes.
- Contextual features: Similar regions should have similar internal and external color distributions. We evaluate these differences using the Wasserstein distance, calculated as: $\Delta D^{(reg)} = \left| W(G_k(R^{(c)}_{in}), G_k(R^{(c)}_{out})) - W(G_k(R^{(t)}_{in}), G_k(R^{(t)}_{out})) \right|$, where $R^{(c)}_{in}$ and $R^{(c)}_{out}$ denote the interior and exterior pixel collections of the candidate region, respectively, and $R^{(t)}_{in}$ and $R^{(t)}_{out}$ for the template region. The external region R_{out} includes pixels covered but not selected during local expansion.

Scalability. Our workflow is fully compatible with deep learning-based frameworks like Semantic-SAM [33] and Mask DINO [32]. Combining them demonstrates the potential to accelerate template generation through prompt-based segmentation at various granularities and refine template matching with instance/object detection. Additionally, our 2D paint canvas (see Fig. 5) converts texture segments into images that are compatible with these segmentation methods. This, combined with our annotated dataset, allows direct training on 3D textured surfaces, setting the stage for future improvements in efficiency and accuracy.

7. Details on benchmark results

7.1. Evaluation of interactive annotation

Evaluation metrics. Traditional metrics, such as click counts to achieve specific Intersection over Union (IoU) or Average Precision (AP), are quantifiable but do not fully capture the true efficiency of the annotation process. The main shortcomings of these methods include:

- Evaluation limitations: Relying solely on IoU or AP does not fully capture annotation comprehensiveness. For example, a 90% IoU may still require multiple boundary adjustments for accuracy.
- Interaction limitations: Click-based interactions alone cannot perfectly annotate boundaries, often requiring tools like lassos or polygons. Additionally, standardized click positions do not account for individual user variations, hindering realistic efficiency assessment.

- Efficiency limitations: Average click counts do not reflect actual interaction efficiency due to varying user speeds. Measuring total annotation time provides a more accurate assessment of efficiency.

To address these issues, we developed an evaluation system comprising Intersection over Union (IoU), Boundary IoU (BIoU), number of operations (Oper), annotation time (Time), and smart interaction ratio (SR). BIoU assesses boundary annotation accuracy [13, 20]. Oper counts mouse clicks and keyboard keystrokes. Time measures annotation duration in seconds. SR quantifies the frequency of non-manual interactions (counting only click-based selections, excluding other operations). Our evaluation is based on user studies with u users across n test scenes, each with c categories. The average metrics are calculated as follows:

- Evaluating a single scenario. For a given scenario s, annotated by u users across c categories, $\overline{mIoU}$, $\overline{mBIoU}$, $\overline{mOper}$, $\overline{mTime}$, and $\overline{mSR}$, can be obtained by averaging the IoU, $mBIoU$, $mOper$, $mTime$, and mSR values across all users and categories.
- Evaluating multiple scenarios. To obtain $\overline{M}$, $\overline{B}$, $\overline{O}$, $\overline{T}$, and $\overline{S}$, the averages of for multiple scenarios, we take the average of each scenario's $\overline{mIoU}$, $\overline{mBIoU}$, $\overline{mOper}$, $\overline{mTime}$, $\overline{mSR}$ and then average these values:

$$\overline{M} = \frac{1}{n}\sum_{s=1}^{n}\overline{mIoU}_s \qquad \overline{B} = \frac{1}{n}\sum_{s=1}^{n}\overline{mBIoU}_s$$

$$\overline{O} = \frac{1}{n}\sum_{s=1}^{n}\overline{mOper}_s \qquad \overline{T} = \frac{1}{n}\sum_{s=1}^{n}\overline{mTime}_s$$

$$\overline{S} = \frac{1}{n}\sum_{s=1}^{n}\overline{mSR}_s$$

In the user study, we recorded each user's annotation progress and interactions in real-time, requiring at least 95% scene completion based on mesh area or texture pixels.

Comparisons. Tab. 5 shows that our method outperforms segment-based annotation [19] in object region and boundary quality. Across all four test scenarios, it significantly reduces both interaction counts and annotation time. We also provide additional qualitative analysis, as shown in Fig. 16 and Fig. 17.

From Tab. 6, our method has slightly lower $\overline{M}$ than GrabCut [50], but achieves higher $\overline{mIoU}_s$ in most scenarios and excels in boundary quality. While SimpleClick [37] is more efficient in interaction count, our method outperforms others in most scenarios. Though slower than SAM [27], our method still surpasses other methods in interaction time. The need for manual corrections enhances annotation quality without significant time cost, and our approach delivers more accurate boundaries with similar correction workloads compared to deep learning methods. We achieve this

Metric	Method	Cour.	Stre.	Park.	Harb.
$\overline{mIoU}_s(\%)$	Seg	89.1	94.0	92.1	91.1
	Ours	**89.5**	**94.2**	**92.9**	**92.2**
$\overline{mBIoU}_s(\%)$	Seg.	**72.7**	**85.4**	70.6	70.3
	Ours.	72.4	84.5	**71.9**	**71.0**
$\overline{mOper}_s$	Man.	18154	1589	3559	3714
	Seg.	17645	1407	2529	2894
	Ours	**13231**	**909**	**1797**	**1957**
$\overline{mTime}_s(s)$	Man.	11401.0	969.5	2146.5	2215.8
	Seg.	10441.3	757.0	1441.4	1623.7
	Ours	**9107.2**	**498.0**	**1105.9**	**1257.5**
$\overline{mSR}_s(\%)$	Ours	**66.5**	**94.9**	**85.8**	**84.8**

Table 5. Performance evaluation of interactive mesh face annotation methods across four scenarios: Cour. (courtyard complex), Stre. (streets with vehicles), Park. (park with trees), Harb. (harbor with ships). Methods include Man. [52] and Seg. [19]. Highest values are shown in bold.

through: (1) Better quality from the user-defined template that enables pixel-level boundary control, outperforming deep learning-based clicks by approximately +3.5∼6.5% mIoU and +18.4∼19.5% boundary mIoU (Tab. 4), especially for regular shapes like windows in Fig. 11. (2) Higher efficiency offered by reusable, scale- and rotation-invariant templates, which reduces the interaction count by -18.7% compared to SAM (582 vs. SAM's 716) and annotation time by -23% compared to SimpleClick (663.3s vs. SimpleClick's 861.3s), benefiting repetitive structures (Tab. 4). Although SAM is slightly faster and SimpleClick requires fewer interactions, our intentional design using handcrafted templates instead of intensive smart clicks prioritizes higher-quality annotations while maintaining a similar total annotation time.

For regular-shaped objects, interactive clicking is suboptimal. As shown in Fig. 18 and Fig. 19, single clicks lack boundary precision, and multiple clicks do not significantly improve accuracy. Repetitive structures increase the annotation burden due to frequent clicking. Instead, users achieve high precision by drawing rectangles or polygons for elements like windows or doors. Our method enables efficient annotation of similar structures by creating a graphical template once. In summary, if manual corrections in semi-automatic annotations take as much or more time than fully manual annotations, the method loses its utility. Additional qualitative results from our annotation methods are presented in Fig. 20.

Ablation studies on template matching. Our feature design is grounded in geometric priors (shape properties and

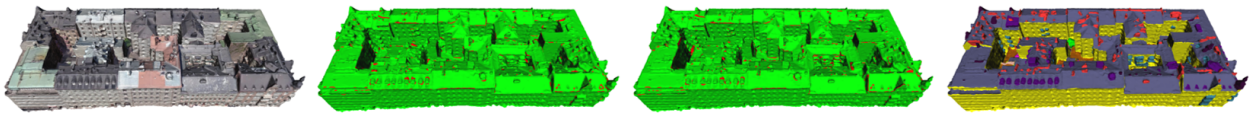

Figure 16. Qualitative analysis of interactive mesh face annotations and their error maps (shown in red) for the courtyard complex.

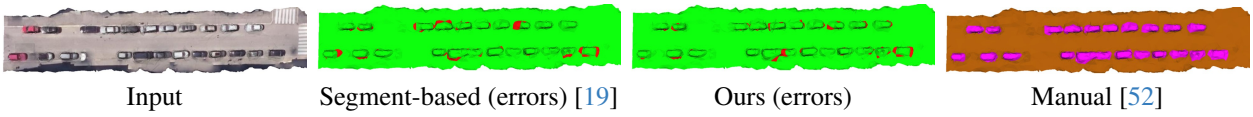

Figure 17. Qualitative analysis of interactive mesh face annotations and their error maps (shown in red) for the street with vehicles.

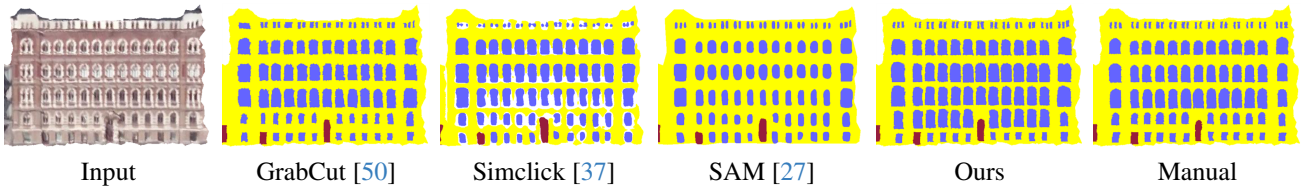

Figure 18. Qualitative analysis of interactive texture annotation results for the facade.

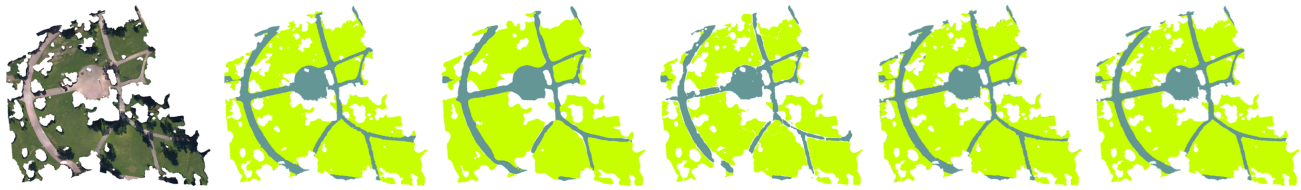

Figure 19. Qualitative analysis of interactive texture annotation results for the park.

structural distribution), label-free operation, and computational efficiency, validated through hierarchical ablation studies (mIoU) as follows. For matching: (1) Planar segments (e.g., roofs, best 88.4% in Fig. 16). Removing geometric homogeneity (-6.1%), spatial distribution (-13.8%), orientation (-13.3%), and shape sphericity (-1.8%) caused performance drops. (2) Protrusions (e.g., cars, best 97.0% in Fig. 17). When spatial compactness (-2.6%), surface complexity (-1.2%), and structural features (-1.6%) were removed, precise matching suffered significantly. (3) Regions (e.g., windows, best 90.7% in Fig. 18). Eliminating shape index (-5.2%), regularity (-38.1%), and contextual features (-20.3%) severely impaired boundary alignment and color consistency. These results highlight the essential role of each feature and their combined effectiveness, confirming our method's superior performance.

7.2. Evaluation of semantic segmentation

1) Face labeling track. Tab. 7 provides a detailed comparison of results for all face-labeled classes. Due to class imbalance, most methods show better performance in categories with more samples and poorer performance in categories with fewer samples. We conducted qualitative analyses on all methods except PointNet for two scenarios, as shown in Fig. 21 and Fig. 22.

2) Pixel labeling track. Tab. 8 provides a detailed comparison of results for all face-labeled and pixel-labeled classes. PointVector [17] surpasses other methods in all categories, particularly with pixel labels. However, compared to the categories shared with Tab. 7, the IoU results for most methods have decreased. This is mainly because the three mesh sampling methods produce relatively uniform point clouds, failing to capture the geometric density variations inherent in adaptive meshes. Additionally, the increase in the number of classes has exacerbated the issue of class imbalance. We performed qualitative analyses for all methods in two scenarios, with global and zoomed-in views, as shown in Fig. 23 and Fig. 24.

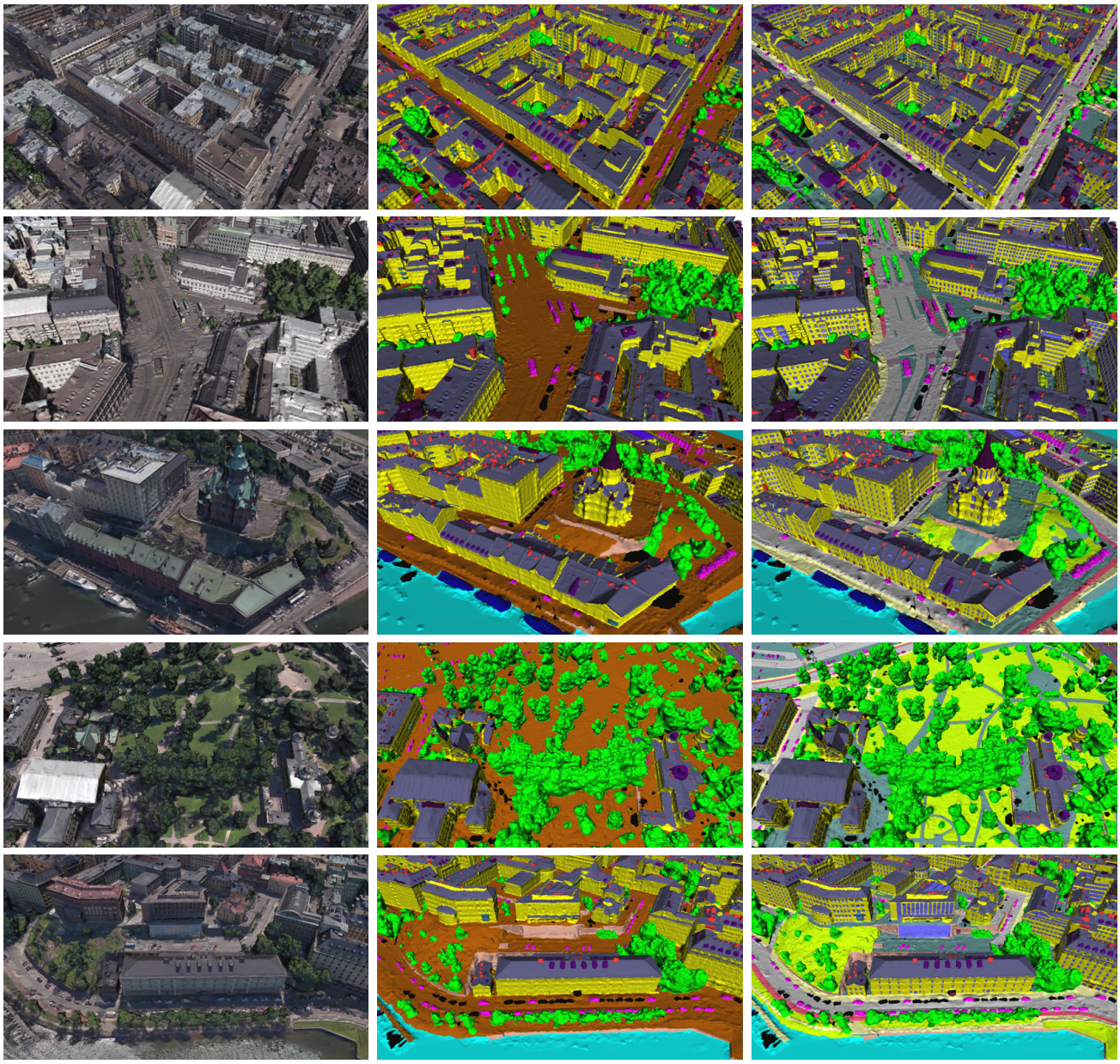

Figure 20. Examples of part-level annotated semantic urban meshes are displayed from the first to the third column, showing textured meshes, face-based semantic meshes (13 classes), and texture-based semantic meshes (19 classes), respectively.

8. Comparison of related datasets

Compare with SUM. Our proposed SUM Parts dataset extends beyond SUM's object-level labels [19], offering three key benefits: **(1)** finer geometric analysis, such as evaluating heat loss at the window-level rather than at the building-scale; **(2)** support for part-aware tasks, e.g., drone navigation for precise delivery by localizing windows, doors, and rooftop solar panel planning; **(3)** seamless integration with urban digital twins and BIM workflows.

Compare with KITTI-360. KITTI-360 [35] focuses on street-view LiDAR-image fusion for autonomous driving, providing 37 Cityscapes-aligned classes, including road-accessible static and dynamic objects ($\geq$0.1m resolution) labeled via manual selection and trajectory-based matching. In contrast, SUM Parts addresses broader urban planning and sustainability challenges using oblique photogramme-

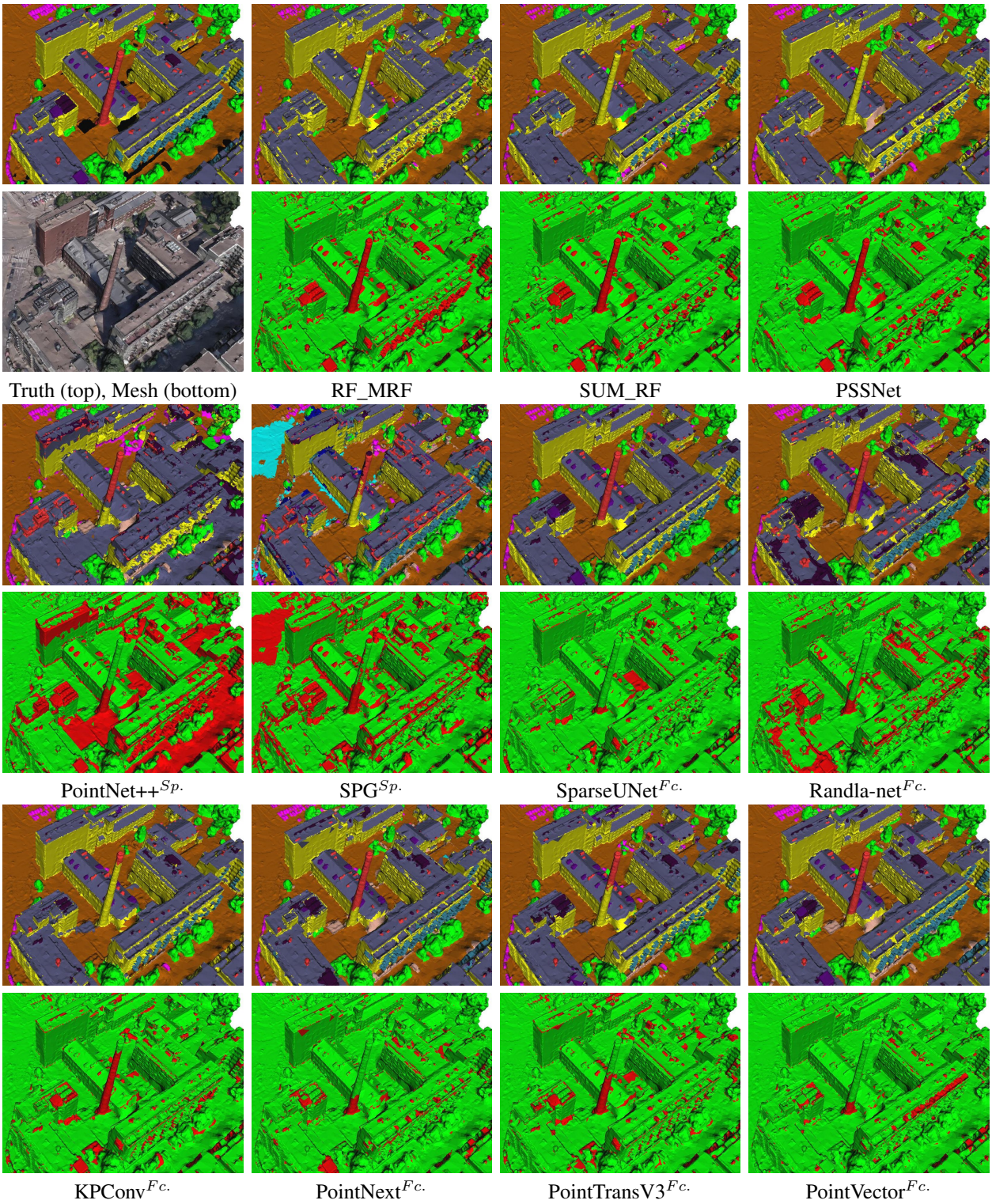

Figure 21. Qualitative analysis of semantic segmentation and error maps in the face labeling track for all methods except PointNet [12] in the first scenario. $^{Fc.}$ and $^{Sp.}$ denote face-centered and superpixel sampling, respectively.

Figure 22. Qualitative analysis of semantic segmentation and error maps in the face labeling track for all methods except PointNet [12] in the second scenario. $^{Fc.}$ and $^{Sp.}$ denote face-centered and superpixel sampling, respectively.

Metric	Method	Fac1.	Fac2.	Par1.	Par2.	Rod1.	Rod2.
$\overline{mIoU}_s(\%)$	Gra.	80.2	87.3	94.0	90.4	91.3*	**85.4**
	SAM	**81.0**	86.4	85.3	86.4	86.5	80.7
	Sip.	73.7	77.5	87.9	86.2	84.0	79.1
	Ours	79.2	**88.7**	**95.0**	**91.2**	**91.3**	82.1
$\overline{mBIoU}_s(\%)$	Gra.	27.8	45.4	63.7	45.8	49.9	50.1
	SAM	24.7	33.9	25.8	23.3	34.1	37.0
	Sip.	19.7	26.8	38.0	34.4	31.8	34.6
	Ours	**28.1**	**50.5**	**67.8**	**48.6**	**50.5**	**50.6**
$\overline{mOper}_s$	Man.	515	124	497	297	718	1764
	Gra.	717	156	462	243	960	1729
	SAM	715	319	363	319	1020	1684
	Sip.	**297**	119	**77**	**78**	**400**	**539**
	Ours	487	**105**	497	213	720	1468
$\overline{mTime}_s(s)$	Man.	816.9	185.6	636.7	280.1	801.7	1941.4
	Gra.	920.1	242.6	494.9	270.8	836.3	1920.6
	SAM	**565.5**	**150.6**	460.9	389.5	800.6	**1476.5**
	Sip.	1128.5	338.9	**197.7**	230.8	1526.4	1745.5
	Ours	631.1	155.8	565.9	**189.1**	**767.9**	1670.3
$\overline{mSR}_s(\%)$	Gra.	7.1	11.8	59.3	66.5	9.9	26.7
	SAM	**94.7**	**92.8**	**78.8**	**78.6**	**49.2**	36.0
	Ours	20.3	21.2	51.6	75.8	32.2	**40.8**

Table 6. Performance evaluation of interactive texture annotation methods across six scenarios: Man. (manual), Gra. (GrabCut [50]), SAM (Segment Anything [27]), Sip. (SimpleClick [37]), Fac1./Fac2. (facades 1 & 2), Par1./Par2. (parks 1 & 2), Rod1./Rod2. (roads 1 & 2). *GrabCut on Rod1. achieved 91.29%, slightly below our method's 91.31%. Highest values are in bold.

try meshes. Key differences include: **(1)** Labeling granularity: SUM Parts offers both object- and part-level annotations (21 CityGML-aligned classes) for fine-grained urban infrastructure details. **(2)** Annotation tools: Our mesh-texture semi-automatic selection tools (click, stroke, lasso) with 2D/3D template matching ensure efficient annotation. **(3)** Coverage: SUM Parts provides full-city coverage, annotating all static objects ($\geq$0.5m resolution), including vehicle-inaccessible areas. Hence, SUM Parts complements KITTI-360 for broader urban applications.

	terr.	hveg.	faca.	wate.	car	boat	roof.	chim.	dorm.	balc.	roin.	wall	OA	mAcc	mIoU
RF_MRF	81.6	86.6	81.3	84.5	24.8	3.7	73.3	27.6	0.0	4.8	0.4	5.9	85.2	45.3	39.5
SUM_RF	84.8	88.1	84.0	79.0	42.5	10.6	77.7	42.4	3.5	22.2	4.7	12.7	86.9	53.6	46.0
PSSNet	80.7	90.5	85.2	64.2	52.6	13.0	78.1	44.0	6.6	25.7	6.9	16.6	86.3	56.4	47.0
PoinNet$^{Sp.}$	52.6	7.1	38.6	59.9	0.0	0.0	22.8	0.0	0.0	0.0	0.0	0.0	50.6	22.0	15.1
PoinNet++$^{Sp.}$	67.9	68.7	59.2	86.1	24.2	11.1	51.1	24.9	0.0	0.0	3.3	1.1	69.0	46.9	33.1
SPG$^{Sp.}$	53.4	55.3	62.5	40.5	27.4	13.1	64.3	33.9	5.1	11.3	3.9	9.9	64.9	55.0	31.7
SparseUNet$^{Fc.}$	88.6	91.7	88.6	76.7	75.6	14.6	82.3	70.1	27.0	49.0	28.0	33.9	90.3	71.7	60.5
Randla-net$^{Fc.}$	86.7	92.3	81.6	**87.1**	82.9	**41.2**	71.6	55.6	21.6	27.6	19.0	21.1	86.7	76.3	57.4
KPConv$^{Fc.}$	86.9	90.8	88.3	81.5	66.4	16.5	81.9	66.7	16.1	45.3	21.2	28.2	90.1	64.7	57.5
PointNext$^{Fc.}$	91.0	95.0	90.4	81.6	91.2	17.9	83.1	74.6	33.8	56.0	30.0	39.3	91.8	77.2	65.3
PointTransV3$^{Fc.}$	88.6	90.1	87.9	78.9	72.1	16.1	81.0	66.2	21.4	45.2	25.0	36.4	89.9	70.2	59.1
PointVector$^{Fc.}$	**92.3**	**96.8**	**91.7**	85.1	**95.2**	22.0	**85.9**	**82.6**	**47.9**	**62.4**	**38.6**	**40.0**	**93.1**	**80.7**	**70.0**

Table 7. Comparison of 3D semantic segmentation methods for face labeling using optimal sampling. Semantic categories: 'terr.' (terrain), 'hveg.' (high vegetation), 'faca.' (facade), 'wate.' (water), 'roof.', 'chim.' (chimney), 'dorm.' (dormer), 'balc.' (balcony), and 'roin.' (roof installation). $^{Fc.}$ and $^{Sp.}$ denote face-centered and superpixel sampling, respectively. Results are presented as IoU (%), Overall Accuracy (OA %), mean Accuracy (mAcc %), and mean IoU (mIoU %). Highest values in IoU, OA, mAcc, and mIoU are highlighted in bold.

	hveg.	faca.	wate.	car	boa.	roof.	chim.	dorm.	balc.	roin.	wall	wind.	door	lveg.	impe.	road	roma.	cycl.	side.	OA	mAcc	mIoU
PoinNet$^{Sp.}$	0.5	13.3	16.5	0.0	2.1	7.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.5	0.0	0.0	0.0	0.2	17.3	9.8	2.6
PoinNet++$^{Sp.}$	72.7	47.5	86.4	34.9	12.4	52.4	28.1	0.0	5.3	5.6	0.4	13.0	5.0	42.4	31.2	14.6	9.5	0.0	7.3	55.4	35.2	24.7
SPG$^{Sp.}$	58.2	50.8	18.4	24.1	2.7	60.4	39.9	3.1	13.6	4.4	10.5	2.4	4.0	13.4	14.6	31.0	0.0	1.7	12.1	51.5	34.5	19.2
SparseUNet$^{Rd.}$	88.8	**70.0**	5.9	51.6	2.5	**79.8**	55.0	12.3	**45.4**	22.6	**31.5**	32.0	12.3	15.2	43.8	44.6	5.2	0.6	35.8	72.9	45.1	34.5
Randla-net$^{Sp.}$	90.5	60.9	84.6	67.6	**22.7**	74.7	53.3	0.6	29.3	16.2	26.3	33.4	12.8	59.8	48.8	50.2	31.5	0.0	37.1	73.5	57.7	42.1
KPConv$^{Sp.}$	84.0	68.5	81.7	68.6	21.8	78.2	**66.4**	**25.0**	41.8	**29.6**	31.5*	36.1	14.9	21.4	35.8	50.0	7.3	13.4	34.1	74.4	58.3	42.6
PointNext$^{Po.}$	90.1	66.2	87.9	68.1	16.3	74.5	59.7	14.9	35.6	19.1	31.0	33.2	13.7	55.5	51.4	55.5	29.0	6.9	40.0	76.0	57.6	44.7
PointTransV3$^{Rd.}$	85.9	59.9	74.6	64.7	17.8	75.9	58.7	15.3	37.2	16.2	29.3	11.8	7.9	27.1	43.3	51.5	3.5	7.2	33.4	70.6	54.1	38.0
PointVector$^{Sp.}$	**92.7**	66.6	**92.0**	**70.2**	19.8	76.8	60.8	21.8	37.0	20.6	30.8	**37.1**	**16.5**	**59.8**	**53.9**	**57.4**	**35.0**	**16.4**	**45.0**	**77.0**	**63.8**	**47.9**

Table 8. Comparison of 3D semantic segmentation methods for pixel labeling using optimal sampling strategies: 'hveg.' (high vegetation), 'faca.' (facade surface), 'wate.' (water), 'roof.' (roof surface), 'chim.' (chimney), 'dorm.' (dormer), 'balc.' (balcony), 'roin.' (roof installation), 'wind.' (window), 'lveg.' (low vegetation), 'impe.' (impervious surfaces), 'roma.' (road marking), 'cycl.' (cycle lane), and 'side.' (sidewalk). Additionally, $^{Sp.}$ denotes superpixel sampling, $^{Rd.}$ for random sampling, and $^{Po.}$ for Poisson-disk sampling [14]. Results are presented as IoU (%), Overall Accuracy (OA %), mean Accuracy (mAcc %), and mean IoU (mIoU %). *KPConv's IoU for wall is 31.46%, slightly below SparseUnet's 31.47%. Highest values in IoU, OA, mAcc, and mIoU are highlighted in bold.

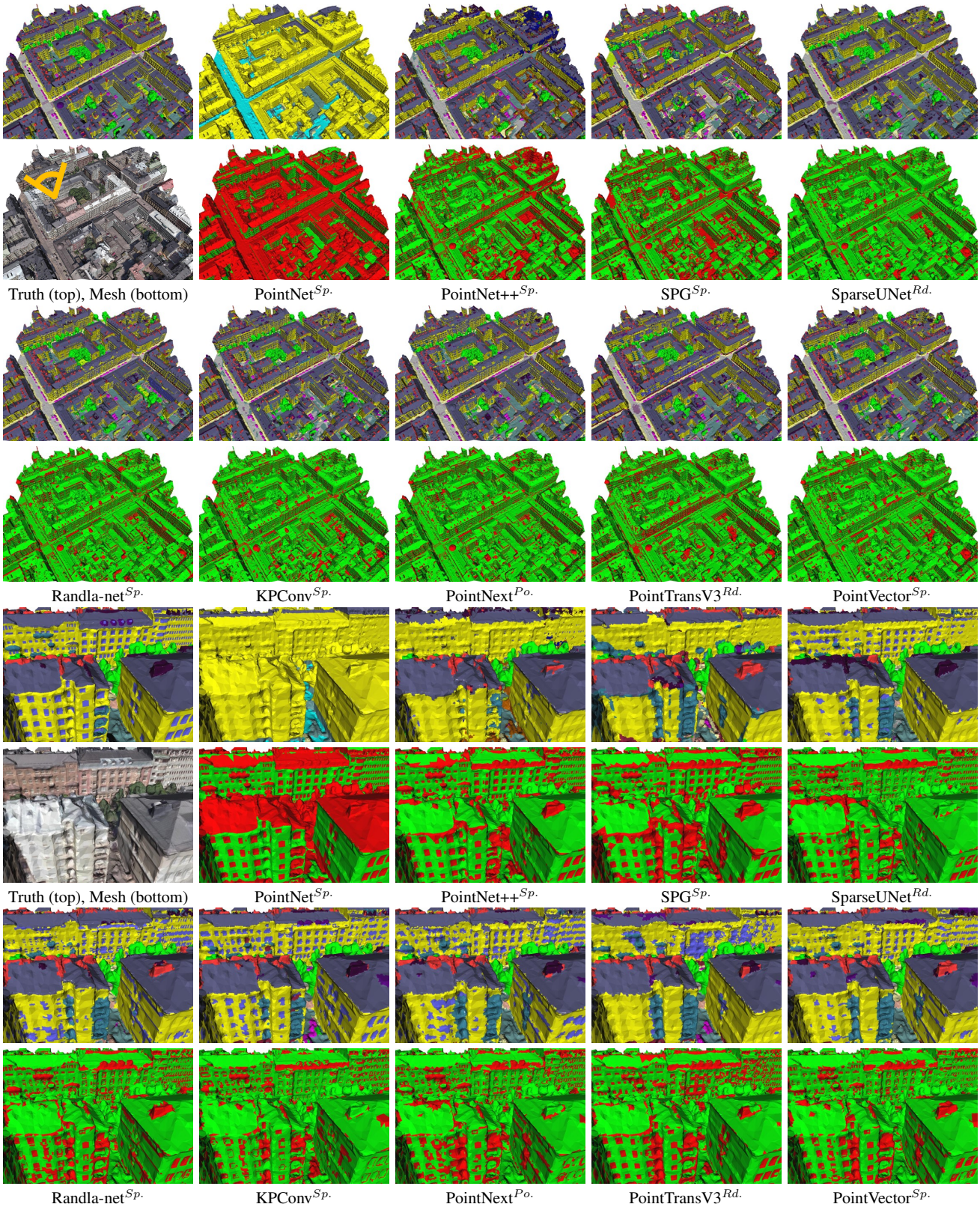

Figure 23. Qualitative analysis of semantic segmentation and error maps in the pixel labeling track for all methods in the first scenario. $^{Sp.}$ denotes superpixel sampling, $^{Rd.}$ for random sampling, and $^{Po.}$ for Poisson-disk sampling [14]. The zoomed-in view direction is indicated in the input mesh image.

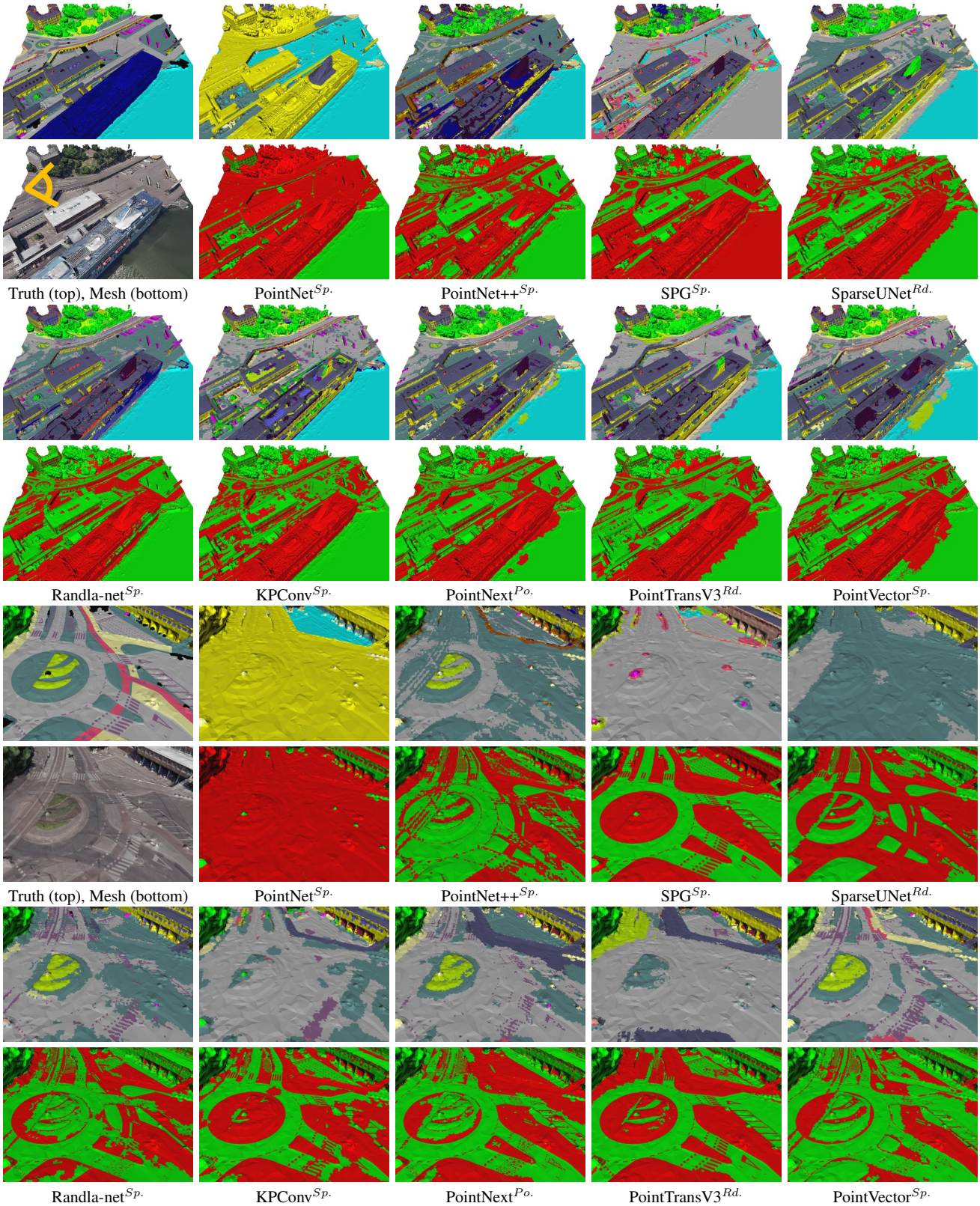

Figure 24. Qualitative analysis of semantic segmentation and error maps in the pixel labeling track for all methods in the second scenario. $^{Sp.}$ denotes superpixel sampling, $^{Rd.}$ for random sampling, and $^{Po.}$ for Poisson-disk sampling [14]. The zoomed-in view direction is indicated in the input mesh image.