IPCD: Intrinsic Point-Cloud Decomposition

Supplementary Material

Contents

A. Details of model architecture

This section describes the detail architectures of IPCD-Net. In IPCD-Net, the set of feature extractor and head is implemented with PTv2. Fig. B (a) illustrates the feature extractor of PTv2, including patch embedding, grid down, grid up layers, and group vector attention (GVA) blocks. Patch embedding transforms the input point cloud into a set of patches, while the grid down and grid up layers extract hierarchical features. The GVA blocks extract interactions between neighboring points. Then, the resulting features are fed into MLP heads, which consists of two layers of multi-layer perceptrons (MLPs). IPCD-Net integrates the Projection-based Luminance Distribution (PLD). PLD is processed via SphereNet [9] as illustrated in Fig. B (b), by adapting the sampling grid to the spherical geometry, ensuring distortion invariance and preserving spatial relationships.

Base model decision. To decide the base model of IPCD-Net, we conduct an ablation study for the base models. PoInt-Net is implemented based on PointNet [33], hence the estimation performance may be improved by replacing the base model with more recent models such as PointNet++ [34], PointCNN [22], PTv1 [54], and PTv2 [47]. The results are summarized in Tab. F. The ablation study demonstrates the superior performance of PTv2 over other base models in the IPCD task, leading us to select it as the base for IPCD-Net.

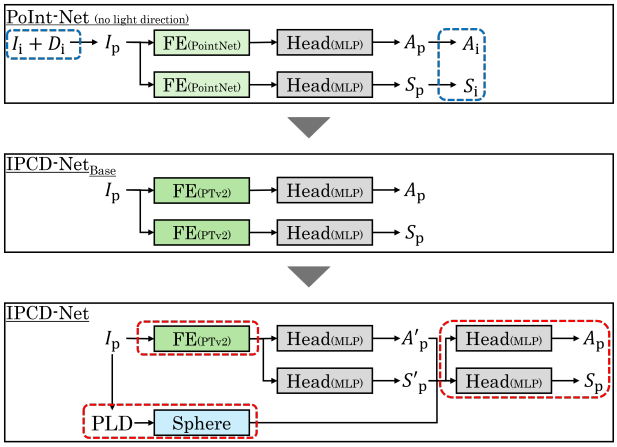

Figure A. Model architectures of PoInt-Net, IPCD-Net$_{Base}$, and IPCD-Net.

Aspect	PoInt-Net [48]	IPCD-Net (ours)
Input	RGB-D (2.5D)	Colored point cloud (3D)
Intermediate	Point cloud	–
Ground truth	Intrinsic images (2D)	Intrinsic points (3D)
Loss computing	Image space (2D)	Point space (3D)
Data structure	Structured (grid)	Unstructured (non-grid)

Table A. Task setting differences of PoInt-Net and IPCD-Net.

Relationship between PoInt-Net, IPCD-Net$_{Base}$, and IPCD-Net. As indicated in Fig. A and Tab. A, PoInt-Net [49] was originally formulated for IID with RGB-D input and image-space supervision. To adapt this design to the IPCD task, we extend it to operate fully in the point-cloud domain, yielding IPCD-Net$_{Base}$. Specifically, IPCD-Net$_{Base}$ directly takes colored point clouds as input, outputs intrinsic decompositions in point space, and computes losses in the 3D geometric domain. This formulation makes it the natural baseline for IPCD-Net. Building upon IPCD-Net$_{Base}$, IPCD-Net incorporates key extensions: (i) a shared encoder, which reduces parameters, stabilizes training, and suppresses leakage, and (ii) PLD, which provides implicit global lighting cues. These additions lead to improved consistency and robustness in intrinsic point-cloud decomposition. For reproducibility, we summarize the PLD parameters. The rendered views are uniformly sampled at 36 azimuth angles and 9 elevation angles (10° step each), resulting in $K = 324$ views over the upper hemisphere. Each view is rendered at a resolution of 256×256 pixels, with a fixed camera-to-scene distance of $d = 60$. The focal ratio is set to 128 and the principal point is fixed at $(0, 0)$. To ensure sufficient geometric fidelity, approximately 1.0×10^6 points are included within the field of view for each rendering.

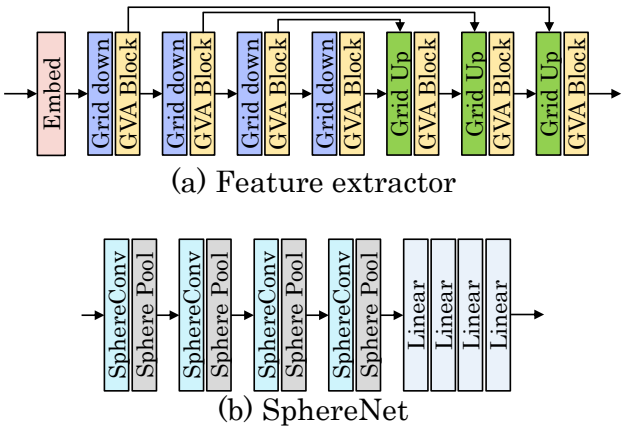

Figure B. Implementation details of (a) feature extractor and (b) SphereNet. The feature extractor is based on PTv2 with patch embedding, grid down, grid up, GVA block, and shared-weight MLP. ShereNet is implemented with SphereCNN and Sphere max polling, to process PLD feature on sphere surface.

Model	MSE(10^{-2})↓		MAE(10^{-1})↓		PSNR↑	
	alb	shd	alb	shd	alb	shd
PoInt-Net [33]	5.80	6.61	1.93	2.08	12.8	12.0
PoInt-Net++ [34]	5.86	6.58	1.96	2.03	12.6	12.1
PointCNN [22]	5.09	5.21	1.82	1.73	13.2	13.2
PTv1 [54]	4.59	5.48	1.70	1.65	13.6	13.2
PTv2 [47]	**4.02**	**5.11**	**1.58**	**1.62**	**14.0**	**13.5**

Table B. Ablation study for albedo (alb) and shade (shd) estimation of base models with IPCD dataset. PoInt-Net [33], PoInt-Net++ [34], PointCNN [22], PTv1 [54], and PTv2 [47] are applied. PTv2 yielded the best score, hence PTv2 is selected as the base model of IPCD-Net.

B. Details of dataset preparation

To verify the performance of IPCD models and compare models, we have prepared a synthetic dataset of 30 diverse scenes od colored points clouds with albedo and shade ground truths. This section describes the details of the assets and lighting conditions.

Assets. We have prepared a diverse set of 30 assets to ensure the robustness of the IPCD task. These assets include a variety of environments such as high-rise buildings, residential neighborhoods, and houses situated within forests. Fig. C depicts the individual assets.

Lighting conditions. Subsequently, these assets were illuminated with lighting simulations of the "Daylight System" in 3ds Max to reflect various realistic environmental conditions. The position on Earth is fixed at Tokyo, Japan (latitude: 35.41, longitude: 139.45). For the time settings, we assigned three different times of day to each scene as shown in Tab. C. We simulated morning, noon and evening for each asset, gradually shifting the time.

Point cloud generation. After applying the lighting, the assets were exported as point clouds by sampling points from the mesh model surfaces. The colored point cloud is defined by color $I \in \mathbb{R}^{N\times 3}$ and position $P \in \mathbb{R}^{N\times 3}$, where N represents the number of the points. Each point i consists of 3D position $P_i = (x_i, y_i, z_i)$ and color $I_i = (I_i^r, I_i^g, I_i^b)$. The resulting dataset includes three types of point clouds for each scene: illuminated, albedo, and shade point clouds. We used random sampling to ensure uniform coverage of the surface, generating 1,000,000 points per asset.

Data format. All point clouds were stored in the PLY format, which includes the position and color for each point. The dataset is structured to ensure compatibility with common point cloud processing frameworks.

Computational Costs. The entire process, including asset preparation, lighting simulation, and point cloud generation, was performed using an RTX 4090 GPU. On average, it took approximately 8 hours per asset to complete the full pipeline, reflecting the computational intensity of the lighting and point cloud generation steps.

Train and test split. To evaluate the performance of the IPCD models, we split the dataset into training and test sets. The training set consists of 23 assets, while the remaining 7 assets are used for testing. This split ensures that the models are evaluated on unseen data, providing a reliable measure of their generalization capabilities. The test set includes "city04", "city08", "city12", "city16", "city20", "city24", and "forest28" assets, while the training set includes the remaining assets.

C. Detailed experimental results

This section provides detailed experimental results that were not included in the main paper due to space limitations. Additionally, we also provide the recent IID-model application to the IPCD task.

Compared IPCD models. We compare the performance of IPCD models including rule-based models (Baseline-A, Baseline-S, and Retinex), recent IID based models (CD-IID, IID-Anything, GS-IR), and IPCD specific models (IPCD-Net$_{\text{Base}}$, IPCD-Net). Baseline-A and Baseline-S treat the input point cloud directly as albedo and shade, respectively. Retinex aims to flatten regions where neighboring points have similar colors, thereby reducing attached shadows. IPCD-Net$_{\text{Base}}$ is a deep learning model trained in a supervised manner to estimate both albedo and shades. IPCD-Net further improves the performance by incorporating PLD, a shared network architecture, and a hierarchical feature refinement. Additionally, multi-view rendered model is also implemented for comparison.

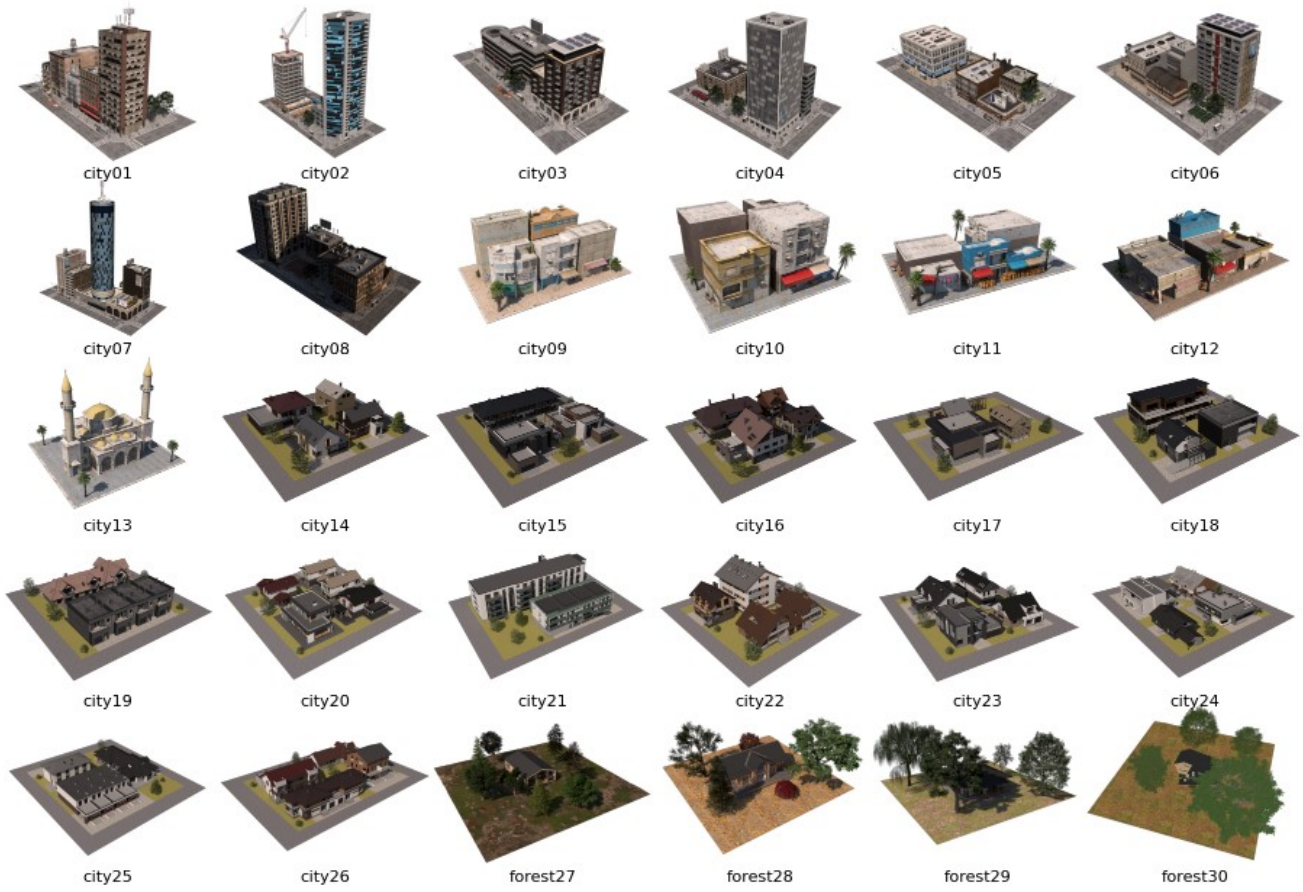

Figure C. Overview of the 30 assets used in the IPCD dataset. The assets include a diverse range of environments such as high-rise buildings, residential neighborhoods, and forest houses, providing a variety of geometric conditions for the IPCD task.

Multi-view rendered model. IPCD is a task of decomposing illuminated point clouds into albedo and shade components. Point clouds can be rendered from any viewpoint, and the rendered images can be used for the IPCD task. Thus, novel-view synthesis models for IID tasks based on neural radiance fields (NeRF) [28] or 3D Gaussian Splatting [18] can be applied to the IPCD task by rendering point clouds from multiple viewpoints. In this paper, we have implemented GS-IR [23], a novel-view synthesis model based on 3D Gaussian Splatting, to compare with the IPCD models. First, one of the test point clouds was rendered from various viewpoints, with a total of 36 azimuth angles and 9 elevation angles at 10-degree intervals, generating 36 $\times$ 9 images. Subsequently, a 3D Gaussian was optimized based on the rendered images, and the corresponding albedo for each image was calculated. By mapping the computed albedo back onto the original point cloud, we performed IPCD indirectly using GS-IR model. We repeated this process for all test point clouds and evaluated the performance of GS-IR in comparison to the IPCD models. Note that, GS-IR only utilizes the test point clouds and does not require the ground-truth albedo and shade components for training.

Recent IID models. We have applied recent IID models to the IPCD task to compare with the IPCD models. The recent IID models include NIID-Net [26] [5], CD-IID [5] [6] and IID-Anything [7] [7]. As illustrated in Fig. F, the test point clouds were rendered from various viewpoints, and the rendered images were used as input to the recent IID models. Subsequently, the albedo and shade components were estimated from the rendered images, and the estimated components were mapped back onto the original point clouds. We evaluated the performance of the recent IID models in comparison to the IPCD models. Note that, publicly available parameters and pre-trained models were used since the source code for training is not available on GitHub pages of CD-IID and IID-Anything. On the other hand, NIID-Net is finetuned with IPCD-dataset. During render process, we generated 256 $\times$ 256 images from

[5]Github page: https://github.com/zju3dv/NIID-Net
[6]Github page: https://github.com/compphoto/Intrinsic
[7]Github page: https://github.com/zju3dv/IntrinsicAnything

Data ID	Name	Asset ID	Time of day	Data ID	Name	Asset ID	Time of day
01	city01-light1	city01	7 : 00	46	city16-light1	city16	8 : 30
02	city01-light2	city01	11 : 00	47	city16-light2	city16	13 : 30
03	city01-light3	city01	15 : 00	48	city16-light3	city16	16 : 30
04	city02-light1	city02	7 : 10	49	city17-light1	city17	7 : 00
05	city02-light2	city02	11 : 10	50	city17-light2	city17	11 : 00
06	city02-light3	city02	15 : 10	51	city17-light3	city17	15 : 00
07	city03-light1	city03	7 : 20	52	city18-light1	city18	7 : 10
08	city03-light2	city03	11 : 20	53	city18-light2	city18	11 : 10
09	city03-light3	city03	15 : 20	54	city18-light3	city18	15 : 10
10	city04-light1	city04	7 : 30	55	city19-light1	city19	7 : 20
11	city04-light2	city04	11 : 30	56	city19-light2	city19	11 : 20
12	city04-light3	city04	15 : 30	57	city19-light3	city19	15 : 20
13	city05-light1	city05	7 : 40	58	city20-light1	city20	7 : 30
14	city05-light2	city05	11 : 40	59	city20-light2	city20	11 : 30
15	city05-light3	city05	15 : 40	60	city20-light3	city20	15 : 30
16	city06-light1	city06	7 : 50	61	city21-light1	city21	7 : 40
17	city06-light2	city06	11 : 50	62	city21-light2	city21	11 : 40
18	city06-light3	city06	15 : 50	63	city21-light3	city21	15 : 40
19	city07-light1	city07	7 : 00	64	city22-light1	city22	7 : 50
20	city07-light2	city07	12 : 00	65	city22-light2	city22	11 : 50
21	city07-light3	city07	16 : 00	66	city22-light3	city22	15 : 50
22	city08-light1	city08	7 : 10	67	city23-light1	city23	7 : 00
23	city08-light2	city08	12 : 10	68	city23-light2	city23	12 : 00
24	city08-light3	city08	16 : 10	69	city23-light3	city23	16 : 00
25	city09-light1	city09	7 : 20	70	city24-light1	city24	7 : 10
26	city09-light2	city09	12 : 20	71	city24-light2	city24	12 : 10
27	city09-light3	city09	16 : 20	72	city24-light3	city24	16 : 10
28	city10-light1	city10	7 : 30	73	city25-light1	city25	7 : 20
29	city10-light2	city10	12 : 30	74	city25-light2	city25	12 : 20
30	city10-light3	city10	16 : 30	75	city25-light3	city25	16 : 20
31	city11-light1	city11	7 : 40	76	city26-light1	city26	7 : 30
32	city11-light2	city11	12 : 40	77	city26-light2	city26	12 : 30
33	city11-light3	city11	16 : 40	78	city26-light3	city26	16 : 30
34	city12-light1	city12	7 : 50	79	forest27-light1	forest27	7 : 40
35	city12-light2	city12	12 : 50	80	forest27-light2	forest27	12 : 40
36	city12-light3	city12	16 : 50	81	forest27-light3	forest27	16 : 40
37	city13-light1	city13	8 : 00	82	forest28-light1	forest28	7 : 50
38	city13-light2	city13	13 : 00	83	forest28-light2	forest28	12 : 50
39	city13-light3	city13	16 : 00	84	forest28-light3	forest28	16 : 50
40	city14-light1	city14	8 : 10	85	forest29-light1	forest29	8 : 00
41	city14-light2	city14	13 : 10	86	forest29-light2	forest29	13 : 00
42	city14-light3	city14	16 : 10	87	forest29-light3	forest29	17 : 00
43	city15-light1	city15	8 : 20	88	forest30-light1	forest30	8 : 10
44	city15-light2	city15	13 : 20	89	forest30-light2	forest30	13 : 10
45	city15-light3	city15	16 : 20	90	forest30-light3	forest30	17 : 10

Table C. Illumination settings by time of day for each asset. Each row represents an asset with its corresponding Data ID, Name, Asset ID, and the specific time of day used for the lighting simulation.

36 azimuth (10° step) and 9 elevation (10° step) angles, covering the hemisphere. The camera distance was set to 60, the focal length was 128 (focal ratio = image size/2), and the principal point was at the image center. Each rendering contained about 1M points to ensure sufficient density. The point radius was set to 0.01 with 10 points

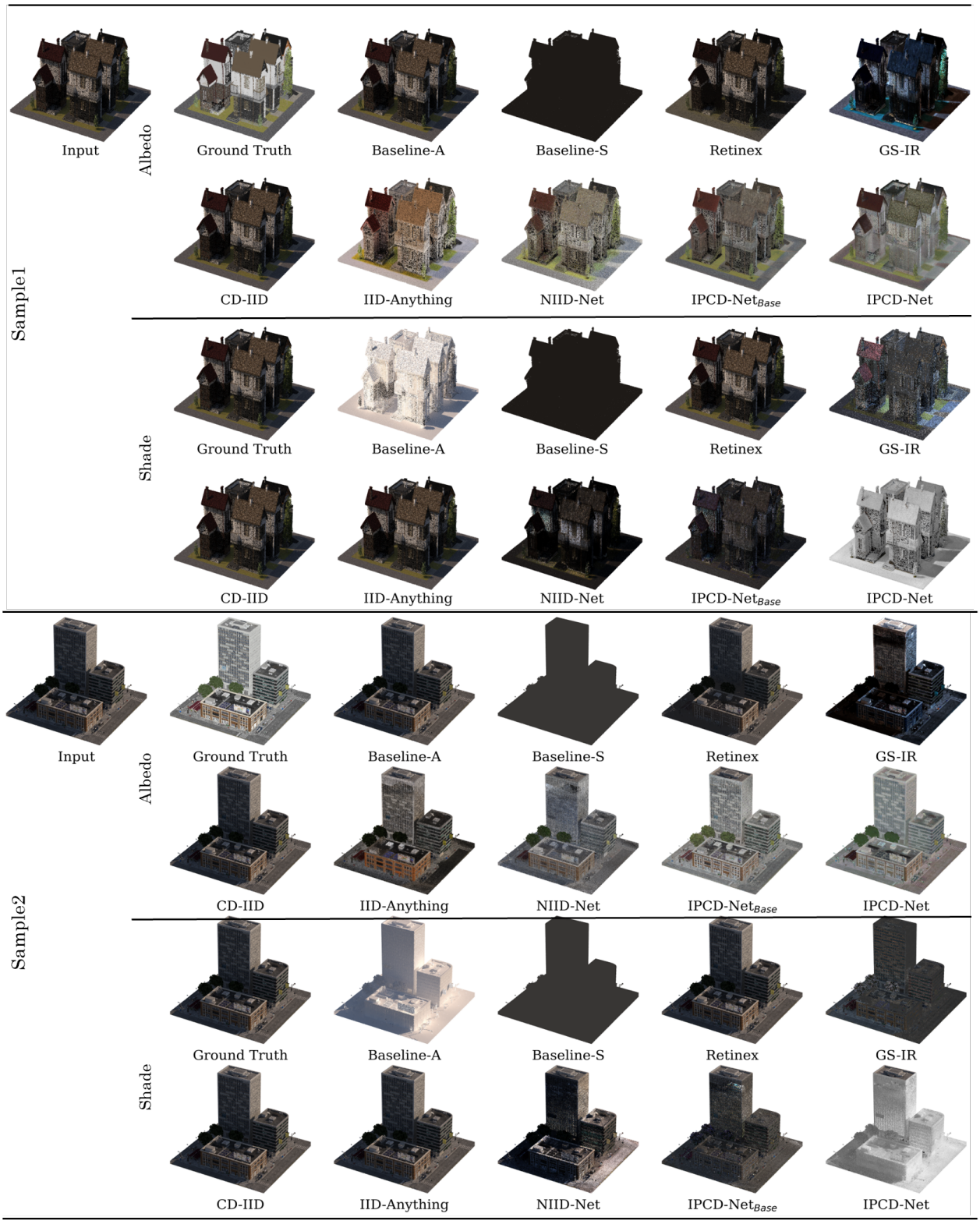

Figure D. Visual comparison of IPCD models including rule-based models (Baseline-A, Baseline-S, and Retinex), recent IID based models (CD-IID, IID-Anything, GS-IR), and IPCD specific models (IPCD-Net$_{\text{Base}}$, IPCD-Net).

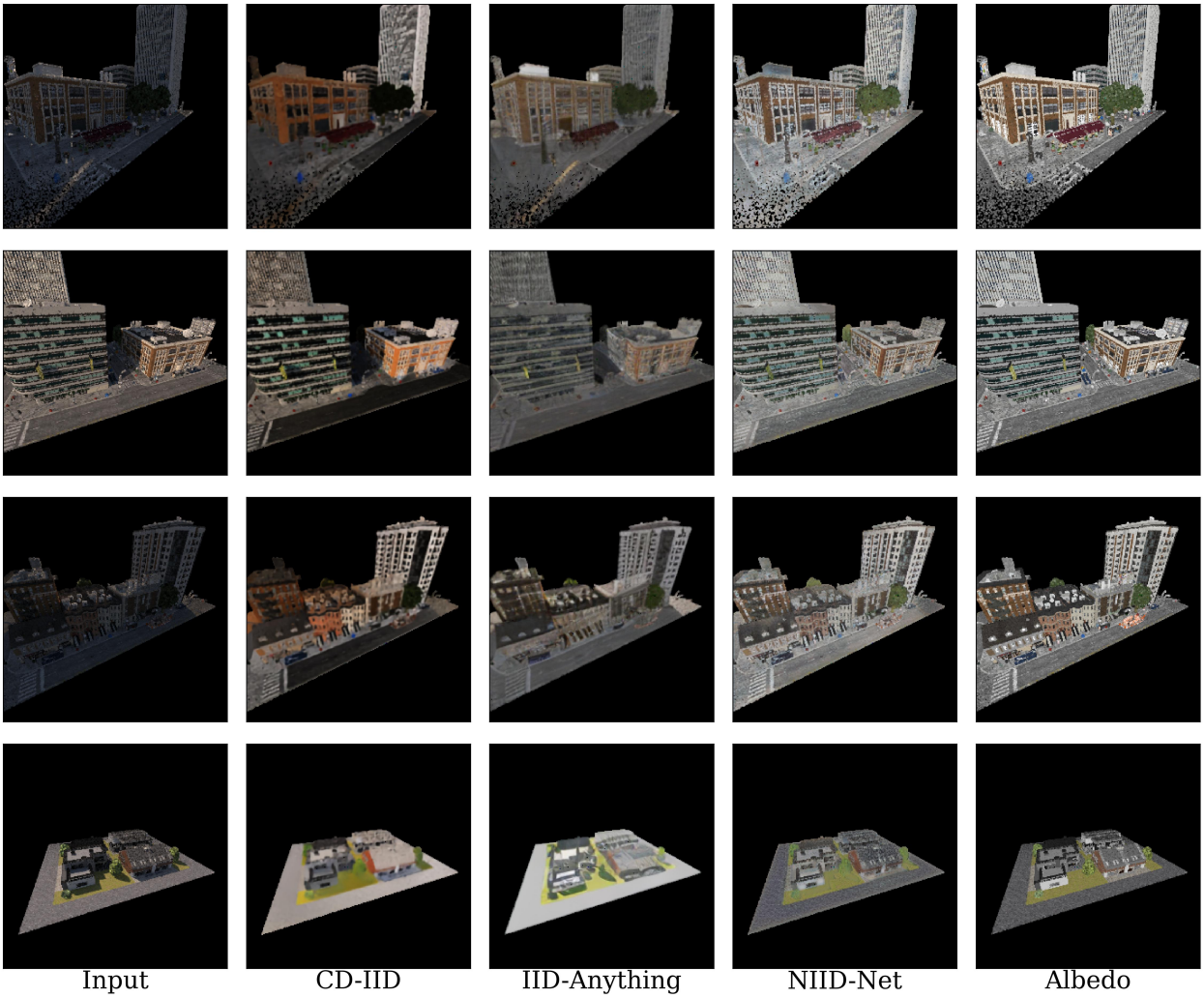

Figure E. Visual verification of rendered images and conventional IID outputs.

per pixel. As shown in Fig. E, rendering artifacts had only minor impact, especially with fine-tuned NIID-Net. These baselines are for reference only, since views are not real images, yet IPCD-Net still outperforms IPCD-Net$_{\text{Base}}$.

Visual results. We provide additional visual results of the albedo and shade components estimated by the IPCD models as shown in Fig. D. Especially, Baseline-A, Baseline-S, CD-IID, and GS-IR were not included in the main paper due to space limitations. Baseline-A and Baseline-S treat the input point cloud directly as albedo and shades, respectively, resulting in albedo and shade estimates that closely resemble the input point cloud. Retinex aim to flatten regions where neighboring points have similar colors, thereby reducing attached shadows. Cast shadows are beyond the scope of these methods, leading to inaccurate shade estimation. GS-IR, a novel-view synthesis model, indirectly estimates albedo by rendering point clouds from multiple viewpoints. GS-IR struggles to estimate albedo and shades, since these models are trained on a per-scene basis, making it difficult to generalize the relationship between input, albedo, and shade across different scenes. CD-IID and IID-Anything are recent IID models that estimate albedo and shades from rendered images. These models struggle to separate albedo and shade components, resulting in inaccurate estimates. In contrast, IPCD-Net$_{\text{Base}}$ and IPCD-Net successfully separate albedo and shade components, providing accurate estimates across various scenes. Especially, IPCD-Net$_{\text{Base}}$ maintains the color information of the input point cloud while successfully separating color shades from albedo.

train strategy	shared encoder	HFR	PLD	SphereCNN	MSE (10^{-2})↓		MAE (10^{-1})↓		PSNR↑	
					alb	shd	alb	shd	alb	shd
step-by-step	-	-	-	-	4.02	5.11	1.58	1.62	14.0	13.5
simultaneous	-	-	-	-	3.58	3.99	1.46	1.48	14.7	14.4
simultaneous	-	✓	-	-	3.43	3.75	1.39	1.41	15.2	14.9
simultaneous	-	✓	✓	-	3.16	3.32	1.33	1.33	15.4	15.2
simultaneous	✓	-	-	-	3.24	3.41	1.35	1.38	15.5	14.9
simultaneous	✓	✓	-	-	3.00	3.46	1.30	1.40	15.6	14.9
simultaneous	✓	✓	✓	-	3.23	3.25	1.36	1.35	15.4	15.1
simultaneous	✓	✓	✓	✓	3.03	3.25	1.31	1.37	15.6	15.1

Table D. Detailed ablation study for IPCD-Net. The ablation study includes the training method (step-by-step or simultaneous), encoder type (separate or shared), hierarchical feature refinement (HFR), PLD usage (w/o or w/), and SphereNet in PLD.

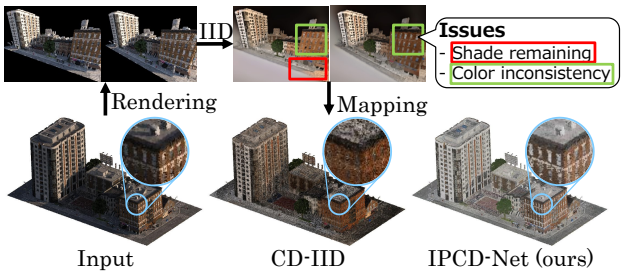

Figure F. Overview of IID model application to IPCD task. We first render images, then infer albedo and shade, and finally map them back to points.

PLD insert	MSE (10^{-2})↓		MAE (10^{-1})↓		PSNR↑	
	alb	shd	alb	shd	alb	shd
Middle only	3.04	3.37	1.31	1.40	15.6	14.9
Middle+Final	3.07	3.23	1.32	1.34	15.5	15.2
Final only	3.03	3.25	1.31	1.37	15.6	15.1

Table E. Detailed ablation study for PLD insert. The ablation study includes the PLD feature insertion at different stages of the network.

D. Detail ablation study

This section provides a detailed ablation study for IPCD-Net to investigate the impact of various factors on the performance of the model. The ablation study includes the training method (step-by-step or simultaneous), encoder type (separate or shared), hierarchical feature refinement (w/o or w/), and PLD usage (w/o or w/). The results are summarized in Tab. D. The ablation study demonstrates that the simultaneous training method outperforms the step-by-step training method, as it allows the model to learn the relationship between albedo and shade components more effectively. Additionally, the shared encoder yields better performance than the separate encoder, as it enables the model to extract features that are more relevant to both albedo and shade components. The hierarchical feature refinement also contribute to final output by more efficiently sharing the albedo and shade features. Finally, the use of PLD also improves the performance of IPCD-Net, as it provides additional information about the luminance distribution of the

Model	MSE(10^{-2})↓	MAE(10^{-1})↓	PSNR↑
Baseline-A [4]	5.78	17.4	12.8
Baseline-S [4]	14.3	29.9	8.82
Retinex [14]	6.12	17.8	12.6
CD-IID [5]	5.28	16.4	13.2
IID-Anything [7]	4.87	15.5	13.6
GS-IR [23]	9.16	23.4	10.8
NIID-Net [26]	4.14	13.4	14.4
IPCD-Net$_{\text{Base}}$ (ours)	1.62	8.71	18.2
IPCD-Net (ours)	**1.08**	**6.72**	**20.0**

Table F. Numerical comparison of relighting performance through IPCD.

Model	f1-score↑
Baseline-A [4]	0.411
Baseline-S [4]	0.219
Retinex [14]	0.478
CD-IID [5]	0.394
IID-Anything [7]	0.453
NIID-Net [26]	0.455
IPCD-Net$_{\text{Base}}$	0.476
IPCD-Net	**0.485**

Table G. Numerical comparison of the relative reflectance between point pairs, following well-known method for real-world scenes. Specifically, we rendered 12 SensatUrban samples and manually annotated 900 point-pairs with relative reflectance labels (darker/lighter/equal).

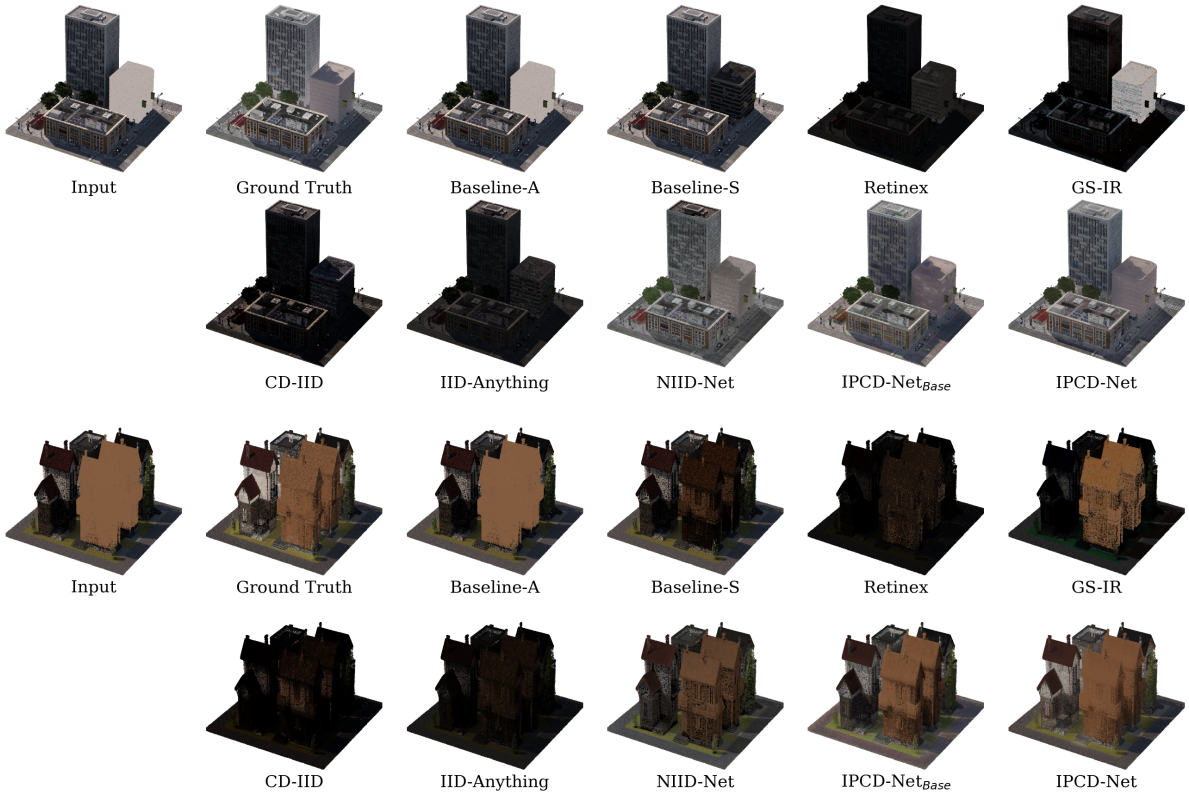

Figure G. Visual results of texture editing.

scene. The best performance is achieved by using the simultaneous training with shared encoder, hierarchical feature refinement, and PLD.

As illustrated in Fig. 2, the IPCD-Net consists of a shared encoder, pre-albedo head, pre-shade head, final albedo head, final shade head, and PLD. The PLD feature is inserted after pre-albedo and pre-shade heads. To investigate the impact of the PLD feature on the performance of IPCD-Net, we conducted an ablation study by inserting the PLD feature at different stages of the network. Tab. E summarizes the results of the ablation study. "Middle only" indicates that the PLD feature is inserted before the pre-albedo and pre-shade heads. "Middle+Final" indicates that the PLD feature is inserted before the pre-albedo and pre-shade heads, as well as before the final albedo and final shade heads. "Final only" indicates that the PLD feature is inserted before the final albedo and final shade heads. "Final only" represents IPCD-Net. "Middle+Final" shows the comparable performance to IPCD-Net, while "Middle only" shows slightly worse performance. This indicates that the PLD feature is more effective when inserted before the final albedo and final shade heads.

E. Details of point-cloud registration

This section provides additional information regarding the point-cloud registration experiment described in Sec. 4.4, including the problem setting, dataset preparation, registration pipeline, and evaluation metrics.

Problem Setting. The primary objective of the registration experiment is to evaluate the effectiveness of the IPCD task for point-cloud registration under varying illumination conditions. Changes in lighting can significantly impact the performance of registration, making it challenging to align point clouds captured at different times of day. To address this, we perform point-cloud registration using the estimated albedo components, which are more invariant to changes in illumination compared to the original point clouds.

Dataset Preparation. The test set of IPCD dataset is utilized for the registration experiment, consisting of point clouds captured at three different sunlit times of day for each asset. First, we apply the IPCD models to the test set to obtain the albedo components for each asset. Subsequently, we prepare pairs of point clouds for registration by selecting two different times of day for each asset, resulting in a total of 21 sets (7 assets $\times$ 3 combinations of times) for registration. Finally, the ground-truth albedo, input illuminated point clouds, and estimated albedo by IPCD models