Unifying Correspondence, Pose and NeRF for Generalized Pose-Free Novel View Synthesis from Stereo Pairs

Sunghwan Hong
Korea University

Jaewoo Jung
Korea University

Heeseong Shin
Korea University

Jiaolong Yang
Microsoft Research Asia

Seungryong Kim
Korea University

Chong Luo
Microsoft Research Asia

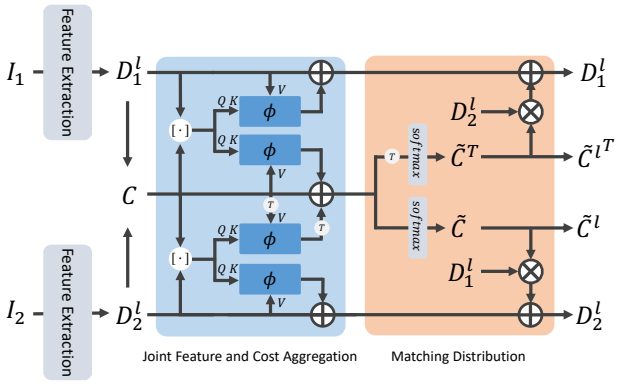


Figure I. **Overview of aggregation module.**

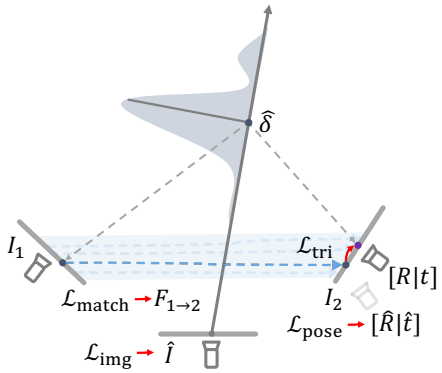


Figure II. **Illustration of our training losses.**

This document includes the following contents: 1) more architectural details of our method, 2) more training and evaluation details of our method and others, 3) distribution of our overlap-based data splitting, 4) more discussions about the experimental results, 5) additional quantitative and qualitative results for the comparision with other methods and our ablation study, and 6) discussion on limitations and future work.

A. Architectural Details

A.1. Feature Aggregation and Cost Filtering

Analogous to existing method, UFC [7], we use attention-based operations for refining both feature and cost volume. We present an overview of the adopted aggregation module in Fig. I.

A.2. Loss Signals

In Fig. II, we show an illustration of our training losses. As shown in the figure, the rendering loss is computed between $\hat{I}$ and I, pose loss is computed using the estimated camera pose $[\hat{R}|\hat{t}]$, flow loss is computed using the estimated flow $F_{1\to2}$ and the triplet consistency loss is computed using $\hat{\delta}$, $[\hat{R}|\hat{t}]$, and $F_{1\to2}$.

B. More Training and Evaluation Details

B.1. Training Details

B.1.1. Our Method

Training Strategy. Our training procedure closely resembles that of Du et al. [4], with distinctions in data usage and augmentation. Instead of applying random cropping and flipping, as done by Du et al., we used a subset of datasets without data augmentation. We train the model with 4 A6000 GPUs for $1\sim2$ days, iterating for 50K iterations, with 192 rays and 64 sampled points on epipolar lines. With this configuration, our rendering speed is approximately 0.4 FPS.

B.1.2. FlowCAM [10]

Architecture. The inference process of FlowCAM is divided into three distinct steps. First, each frame within

a video sequence undergoes feature extraction, yielding deep features and backward flows. Subsequently, using the single-view pixelNeRF algorithm to a surface point cloud, representing the anticipated 3D termination point for each pixel. Next, within each frame, confidence weights are computed among the points. This is achieved by utilizing the RAFT [11] predictions. Finally, these computed confidence weights are fed into a sequence of linear layers to derive the final confidence weights required for solving the Weighted Procrustes formulation [2], and then the desired views are re-rendered.

Training Strategy. For training, we take 256×256 input images. While RealEstate10K was already trained by the authors and the pretrained weights were available, we verify that the training scheme authors provide can reliably transfer to ACID, we attempted reproducing the results on RealEstate first, for which we were able to reproduce the results close to those reported in the paper. Given this success, we followed the same training procedure used and released by the authors for RealEstate to train on ACID. We train for 50K iterations with a single A6000 GPU, which takes approximately 1.5 days, with leaving other hyperparameters unchanged.

B.1.3. Rockwell *et al.* [8]

Architecture. In their architecture, there are three main components: Image encoder, ViT layer and Essential Matrix Module followed by MLPs. Taking 256×256 input image pairs as inputs, the image is resized to 224×224, and then the model first extracts deep features via vanilla resnet-18. Then the feature maps from the coarsest layer are fed to ViT-Tiny for self-attention operations. Subsequently, these feature maps are fed to the Essential Matrix Module, which performs the cross-attention that emulates the 8-point algorithm, and finally, the output is reshaped and fed to the pose regression MLPs.

Training Strategy. For training, we follow the same procedure and adopt the default hyperparameters used in the training scripts, as provided in the official github repository that the authors provide for both RealEstate10K and ACID. We use the same data sampling strategy as the one we used to train our model. Specifically, for each scene consisting of a video sequence, we use the first and the last frame as the input images, and the ground-truth relative pose for supervision is computed between them. We trained the network for a total of 120K iterations with batch size set to 32 using a single A6000 GPU.

B.1.3. RelPose [14]

Architecture. Relpose inference is divided into two steps. A pairwise pose prediction step is followed by a joint reasoning step of multiple pairwise estimated relative poses. By taking a set of images as input, they first group all possible pairs of images to estimate all the pairwise relative poses between images. Leveraging an energy-based model, the estimated pairwise relative poses recover a probability distribution over conditional relative rotations where the condition is given as the uniformly sampled relative pose $R \in \mathrm{SO}(3)$. Estimated poses are further refined in the joint reasoning step by inducing a joint likelihood over the camera transformations across multiple images and iteratively improving an initial estimate by maximizing this likelihood.

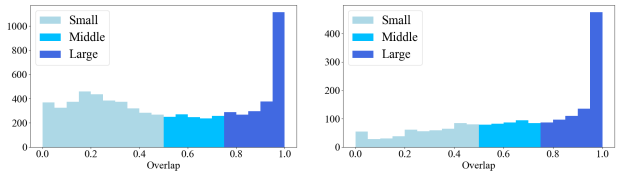
Training Strategy. For training, we follow the same training strategy as [4] and ours, since we aim to compare the performance of relative pose estimation given stereo pairs. However, when using only stereo pairs as input, the joint reasoning step cannot be done as there is only one estimated pose. To make a fair comparison, we increased the number of uniformly sampled relative pose $R \in \mathrm{SO}(3)$ from $N = 36864$ to $N = 250000$, which is the number of queries used in the second stage of the framework. The training was done for 400K iterations of batch size set to 64, using four A6000 GPUs.

B.1.4. DBARF [1]

Architecture. The architecture of DBARF consists of three components: an image encoder, a Pose and Depth Estimation Module, and a Renderer for novel view synthesis. By selecting a target image and nearby images from a scene graph, the ResNet-like [6] image encoder first extracts a feature map used for the subsequent steps. The feature maps of the nearby images are then warped to the target view using the currently estimated camera poses and depths to construct a local cost map for pose and depth estimation done with training a recurrent GRU. The estimated pose is then used as an input of the Renderer, where they use the IBRNet [13] to render novel views. To enable robust optimization of both the Pose and Depth Estimation Module and the Renderer, they adopt a staged training strategy of dividing the overall training process into three steps: training only the Pose and Depth Estimation Module, training only the Renderer, and jointly training the two components.

Training Strategy. For training, we first take 256×256 input images and then resize them to 224×224. As there are no provided weights for DBARF on RealEstate10K and ACID, we trained the network from scratch following the process provided by the authors. For both datasets, we selected six nearby views of the target view during training by selecting three frames before the target frame with a 10, 20, and 30 frame difference each and three frames after the target frame with a 10, 20, and 30 frame difference. We trained the network for a total of 200K iterations, where the

Figure III. **Qualitative results for component ablation study.** Consistent with the quantitative results (Table 3 of the main paper), each variant exhibits apparent differences in qualitative comparisons and shows the efficacy of our designed components.



(a) RealEstate-10K (b) ACID

Figure IV. **Distribution of the data splits.**

three stages of their proposed staged training were repeated every 10K steps. The training was done with a single A6000 GPU.

B.2. Evaluation Details

Differences of the evaluation strategy in original work of DBARF and FlowCAM and our adopted evaluation strategy. In the original evaluation strategy of DBARF, given a sequence of frames, DBARF picks an abitrary view, treating it as a target view, and selects nearby views, considering them as context images. Subsequently, pairwise pose estimation and depth estimation are performed between the target image and each of the context images. The estimated values are then fed to GeNeRF for rendering and evaluation. In our evaluation approach, we assume that only two context images and a relative camera pose to the target view are provided. This means that in contrast to the original evaluation setting, where target view is accessible for depth and pose estimation, our evaluation setting does not employ the target view for the model, but only accessible for metric computation.

In the evaluation setting of FlowCAM, FlowCAM first feeds all the frames to single-view PixelNeRF, and they are used for warping by an off-the-shelf optical flow network to output matching confidence that will be used for Weighted Procrustes formulation. Similar to DBARF, the target views are accessed for pose estimation in this evaluation setting. In our evaluation setting, we substitute the estimated poses with ground-truth poses except for the relative pose between I_1 and I_2, which is their prediction and remains unchanged.

Fixed Pose and Generalized NeRF. In Table 4 of the main paper, we conducted additional ablation study and analysis. While all other experiments follow the same evaluation protocol, some subtle changes are made for (a). Specifically, as PDC-Net encountered some cases where RANSAC failed to converge due to extremely small overlapping regions with barely any correspondence, we exclude such cases during metric calculations. For a fair comparison, for the other variants, *i.e.*, Rockwell *et al.*+[4] and ours, we also disregard those scenes, which leads to different results compared to our main table.

C. Data Split Details

We provide the statistics of each overlap-based data split of RealEstate10K and ACID in Fig. IV.

D. More Discussions

Pose Supervision. In this section, we discuss the results we obtained from the experiment that combines direct pose supervision to DBARF and FlowCAM framework. Contrary to initial expectations, although some quantitative performance improvements were observed for image quality assessment, adding direct pose supervision results in a decline in their pose estimation performance despite our meticulous efforts to optimize the hyperparameters and conduct multiple trials. We hypothesize that the performance degradation could be attributed to their original training strategy, which only assumes image sequences with small viewpoint changes and could have influenced overall performance largely. Another potential reason is that their incorporation of pose estimation and the rendering requires non-trivial implementation considerations to obtain a boosted synergy, or the confidence score produced by the off-the-shelf model might've made a disparity between the updated renderer and the optical flow prediction. Despite the challenges we faced and the hypothesis we made, we leave this exploration as future work since such investigation is beyond the scope of this paper.

E. More Results

E.1. Absolute Translation Error with Scales

Table I presents the results of translation estimation evaluated with both absolute error in meters and angular difference in degrees. Note that, as mentioned in the main paper, evaluating absolute error requires assessing the models' ability to gauge scales via, *e.g.*, object recognition, since translation scale is theoretically indeterminable in two-view geometry. This could potentially result in erroneous interpretations regarding the models' proficiency in estimating relative camera pose from two views.

E.2. More Qualitative Results

Fig. V shows the correspondences built by our method and the overlapping regions characterized by high confidence scores. As we can see, our method can detect matching points robustly across different scenarios.

Fig. VI visualizes the epipolar lines with the relative poses estimated by different methods. Visually inspected, our method yields more accurate results especially on challenging cases with small overlap.

Fig. VII and Fig. VIII present more novel view rendering results of different methods. On both datasets, our method yields outcomes that are sharper and more geometrically accurate.

We've also obtained more novel view rendering results our method under continuous viewpoint change, which can be found in the *accompanying video*.

E.3. Qualitative Results for Ablation Study

In Fig. III, we provide qualitative comparisons for each variant introduced for the component ablation study. Consistent with the quantitative results, each variant exhibits apparent differences in qualitative comparisons as well.

F. Limitations and Future works

As our work takes two input views as inputs, it fails to model dynamic scenes and view extrapolation. The visual examples are found in the *accompanying video*.

In our future work, we plan to incorporate designs that can aggregate information from more than two views, broadening our scope to encompass multi-view encoding or aggregation. Additionally, we plan to train our model on larger real-world data to enhance its universality and practical applicability to real-world situations.

Overlap	Task	Method	RealEstate-10K						ACID					
			Translation			Translation			Translation			Translation		
			Avg(m)↓	Med(m)↓	STD(m)↓	Avg(°)↓	Med(°)↓	STD(°)↓	Avg(m)↓	Med(m)↓	STD(m)↓	Avg(°)↓	Med(°)↓	STD(°)↓
Small	Matching	SP+SG [3, 5, 9]	0.973	0.759	0.840	12.549	4.638	23.048	0.979	0.661	1.094	22.214	7.526	33.719
		PDC-Net+ [5, 12]	0.696	0.597	0.591	6.913	2.752	15.558	0.667	0.573	0.714	15.664	4.215	29.640
	Pose Estimation	Rockwell *et al.* [8]	1.692	1.459	1.119	91.455	91.499	56.872	1.576	1.057	3.557	88.421	88.958	36.212
		RelPose [14]	-	-	-	-	-	-	-	-	-	-	-	-
	Pose-Free NeRF	DBARF [1]	2.782	2.549	1.803	126.282	140.358	43.691	2.134	1.187	6.959	95.149	99.490	47.576
		FlowCAM* [10]	**1.543**	**1.400**	**0.901**	112.094	118.786	33.168	0.092	**0.040**	0.166	94.618	89.410	40.611
		Ours	0.532	0.353	0.642	**11.862**	**5.344**	**21.080**	**0.378**	0.171	**0.533**	**23.689**	**11.289**	**30.391**
Medium	Matching	SP+SG [3, 5, 9]	0.390	0.344	0.261	9.295	3.279	20.456	0.528	0.466	0.431	16.455	5.426	29.035
		PDC-Net+ [5, 12]	0.360	0.322	0.253	6.667	2.262	18.247	0.612	0.563	0.482	14.940	4.301	27.379
	Pose Estimation	Rockwell *et al.* [8]	0.842	0.705	0.581	82.478	82.920	55.094	0.713	0.554	0.649	90.555	90.799	51.469
		RelPose [14]	-	-	-	-	-	-	-	-	-	-	-	-
	Pose-Free NeRF	DBARF [1]	0.816	0.574	0.782	79.402	75.408	54.485	0.772	0.473	0.931	77.324	77.291	49.735
		FlowCAM* [10]	0.068	**0.048**	0.065	127.306	133.035	32.911	0.080	**0.036**	0.211	96.228	89.828	42.405
		Ours	**0.203**	0.150	**0.178**	**10.187**	**5.749**	**15.801**	**0.324**	0.133	**0.615**	**21.401**	**10.656**	**28.243**
Large	Matching	SP+SG [3, 5, 9]	0.612	0.665	0.202	21.415	7.190	34.044	0.619	0.641	0.260	22.018	7.309	33.775
		PDC-Net+ [5, 12]	0.601	0.659	0.200	16.567	5.447	29.883	0.707	0.606	0.882	18.447	4.357	35.564
	Pose Estimation	Rockwell *et al.* [8]	0.468	0.363	0.377	91.851	88.923	57.444	0.431	0.304	0.457	86.580	87.559	50.369
		RelPose [14]	-	-	-	-	-	-	-	-	-	-	-	-
	Pose-Free NeRF	DBARF [1]	0.217	0.098	0.318	50.094	33.959	43.659	**0.281**	0.111	**0.488**	54.523	38.829	45.453
		FlowCAM* [10]	0.025	**0.011**	0.34	133.236	144.151	39.139	0.089	0.037	0.2987	99.362	93.467	42.823
		Ours	**0.095**	0.067	**0.102**	**15.544**	**7.907**	**24.626**	0.456	0.146	2.762	**22.935**	**10.588**	**30.974**
Avg	Matching	SP+SG [3, 5, 9]	0.749	0.629	0.654	14.887	5.058	27.238	0.703	0.610	0.676	20.802	6.878	32.834
		PDC-Net+ [5, 12]	0.696	0.0.597	0.591	10.100	3.243	22.317	0.671	0.587	0.744	16.461	4.292	31.391
	Pose Estimation	Rockwell *et al.* [8]	1.145	0.821	1.022	90.115	88.648	40.948	0.833	0.500	2.041	88.433	88.961	36.197
		RelPose [14]	-	-	-	-	-	-	-	-	-	-	-	-
	Pose-Free NeRF	DBARF [1]	1.603	0.930	1.787	93.300	102.467	57.290	0.939	0.366	3.901	71.711	68.892	50.277
		FlowCAM* [10]	0.089	**0.048**	0.125	121.645	32.418	33.753	0.612	**0.102**	5.275	97.231	91.536	42.067
		Ours	**0.332**	0.177	**0.506**	**12.766**	**7.534**	**15.510**	**0.404**	0.150	**1.965**	**22.809**	**14.502**	**21.572**

Table I. Translation estimation performance evaluated with both absolute error (in meters) and angular error (in degrees). Note that since translation scale is theoretically indeterminable in two-view geometry, evaluating absolute error requires assessing the models' ability to gauge scales via, *e.g.*, object recognition. This could potentially result in erroneous interpretations regarding the models' proficiency in estimating relative pose from two views. *: FlowCAM [10] results have been updated to rectify an error in the numerical values originally presented. We apologize for any inconvenience caused.

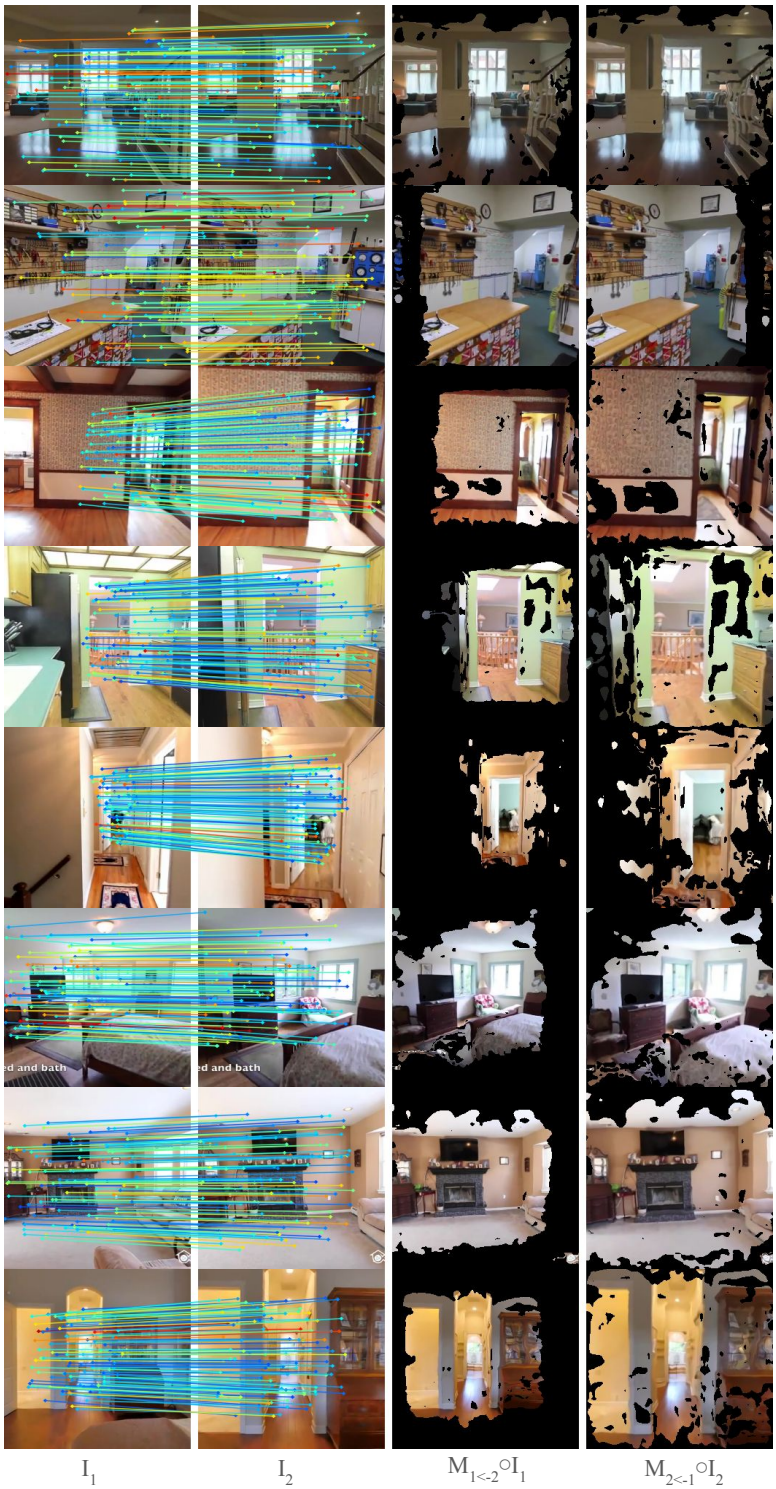


Figure V. **Randomly selected correspondences and confident regions.** For each pair of images, we visualize a set of randomly selected correspondences (left), and from the complete set of correspondences, and those with confidence score of higher than a threshold τ are shown as visible (right).

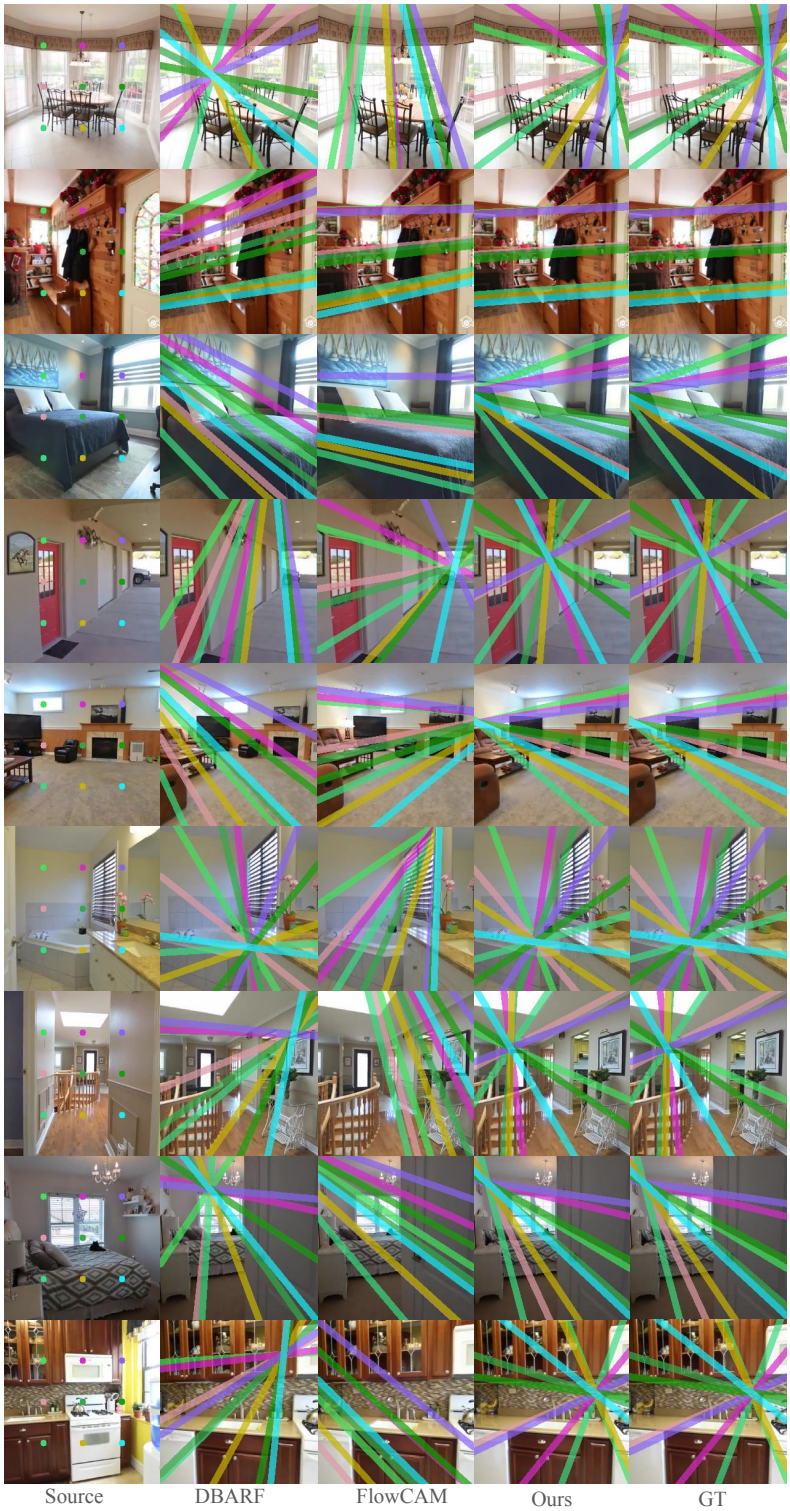
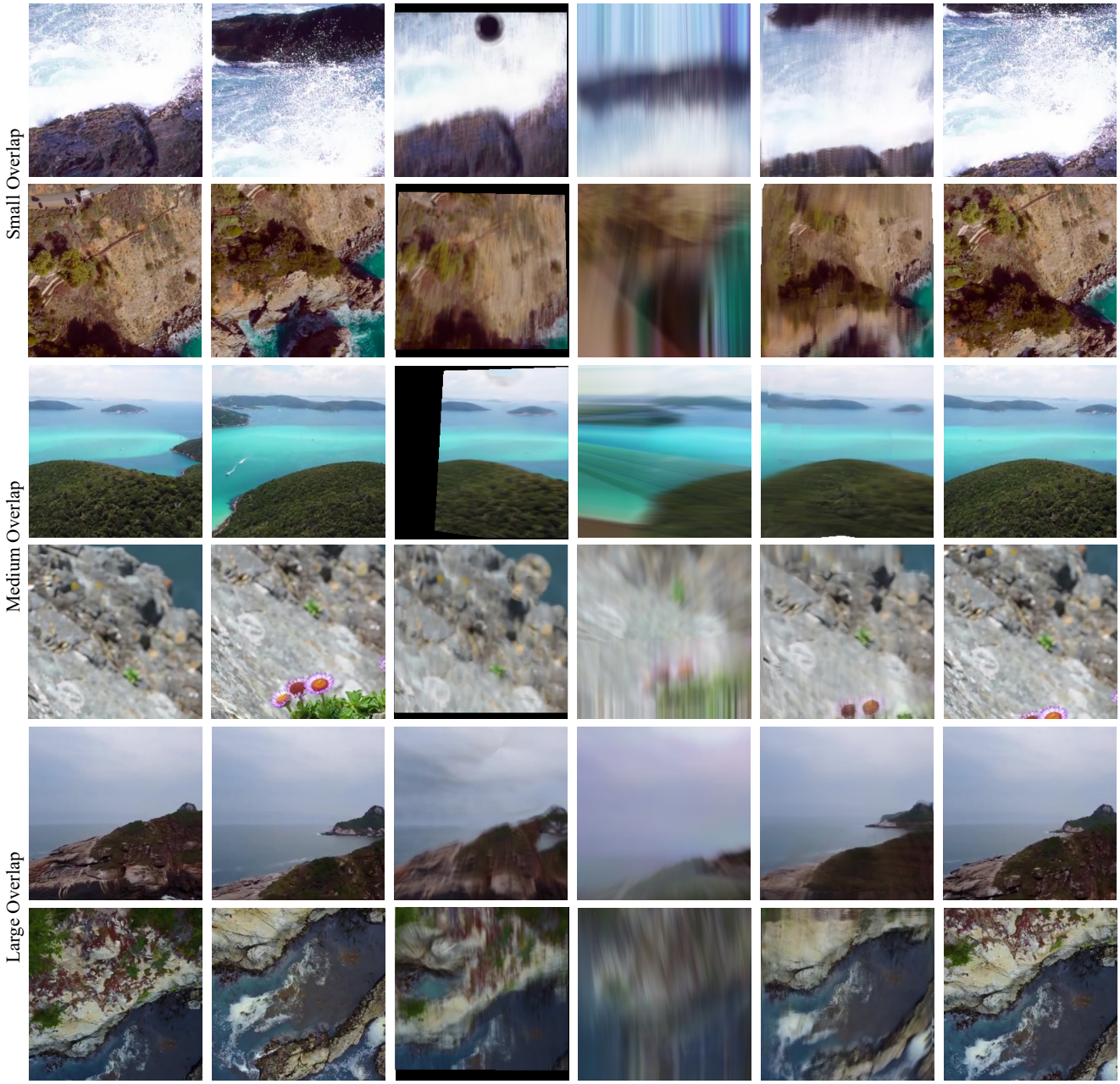


Figure VI. **Comparisons of visualized epipolar lines.**

(a) I_1 (b) I_2 (c) DBARF (d) FlowCAM (e) Ours (f) Ground Truth

Figure VII. **Qualitative comparison on RealEstate10K.**



(a) I_1 (b) I_2 (c) DBARF (d) FlowCAM (e) Ours (f) Ground Truth

Figure VIII. **Qualitative comparison on ACID.**

References

[1] Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24–34, 2023. 2, 5

[2] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2514–2523, 2020. 2

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 5

[4] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023. 1, 2, 4

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[7] Sunghwan Hong, Seokju Cho, Seungryong Kim, and Stephen Lin. Unifying feature and cost aggregation with transformers for semantic and visual correspondence. *arXiv preprint arXiv:2403.11120*, 2024. 1

[8] Chris Rockwell, Justin Johnson, and David F Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by vits. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 2, 5

[9] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 5

[10] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann. Flowcam: Training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. *arXiv preprint arXiv:2306.00180*, 2023. 1, 5

[11] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2

[12] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 5

[13] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2

[14] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision*, pages 592–611. Springer, 2022. 2, 5