

Sky2Ground: A Benchmark for Site Modeling under Varying Altitude

Zengyan Wang, Sirshapan Mitra, Rajat Modi, Grace Lim, Yogesh Rawat
{zengyan.wang, yogesh}@ucf.edu
Center for Research In Computer Vision
University of Central Florida

Abstract

We introduce *Sky2Ground*, a three-view dataset designed for varying altitude camera localization, correspondence learning, and reconstruction. The dataset combines structured synthetic imagery with real, in-the-wild images, providing both controlled multi-view geometry and realistic scene noise. Each of the 51 sites contains thousands of satellite, aerial, and ground images spanning wide altitude ranges and nearly orthogonal viewing angles, enabling rigorous evaluation across global-to-local contexts. We benchmark state of the art pose estimation models, including MAST3R, DUST3R, Map Anything, and VGGT, and observe that the use of satellite imagery often degrades performance, highlighting the challenges under large altitude variations. We also examine reconstruction methods, highlighting the challenges introduced by sparse geometric overlap, varying perspectives, and the use of real imagery, which often introduces noise and reduces rendering quality. To address some of these challenges, we propose *SkyNet*, a model which enhances cross-view consistency when incorporating satellite imagery with a curriculum-based training strategy to progressively incorporate more satellite views. *SkyNet* significantly strengthens multi-view alignment and outperforms existing methods by 9.6% on RRA@5 and 18.1% on RTA@5 in terms of absolute performance. *Sky2Ground* and *SkyNet* together establish a comprehensive testbed and baseline for advancing large-scale, multi-altitude 3D perception and generalizable camera localization. Code and models will be released publicly for future research. Project page: <https://sky2ground2026.github.io/sky2ground/>

1. Introduction

One of the *fundamental* tasks in computer vision has been to *jointly* localize multi-view images and construct *reliable* 3D models of a scene. This has found many commercial applications, such as geo-registering ground-images [23, 27, 37], *outdoor-scene* reconstruction [1, 25, 28], urban-view synthesis [14, 17, 39, 41], and complex navigation [16, 23, 26]. Classical pipelines like COLMAP [30, 31]

have leveraged hand-designed features and matching heuristics, which makes them *computationally-intensive* in practice. Recently, neural nets have achieved great success in inferring geometry parameters like camera poses/depth maps, thereby enabling *faster* predictions [15, 35, 36]. This progress has been largely facilitated by *pre-training* on ground-aerial datasets [34]. However, modeling large-scale scenes across *multiple* altitudes and viewpoints remains underexplored, notably *within the domain of satellite imagery*.

Unlike ground/aerial views, satellite views provide globally consistent coverage, stable geospatial reference frames, and fine spatial resolution [21]. Recent work uses satellite imagery for localization from oblique or UAV perspectives in GPS-denied environments [5, 21, 25]. We notice a *lack* of datasets which provide *simultaneous* access to ground/aerial/satellite views. Furthermore, to the best of our knowledge, there is *no work* focusing on joint-localization of cameras across *all three-views*, i.e. ground/aerial/satellite. However, we believe this problem to be *very-timely*, given the recent-efforts trying to achieve the *ambitious-goal* of *planet-wide* reconstruction [34].

Motivated by this, we propose a multi-altitude, cross-view dataset called **Sky2Ground** that *unifies synthetic* satellite, aerial, and ground-level imagery *with real* aerial and ground imagery collected from-the-wild. The *synthetic component* provides dense, globally consistent camera poses and structured 3D geometry, while the *real component* introduces photometric variability, occlusions, and environmental complexity. As a result, our dataset enables a *thorough evaluation* of cross-view localization and 3D reconstruction pipelines.

Furthermore, we perform a detailed benchmark study which reveals that existing pose estimation/reconstruction methods *struggle under* large distributional shifts introduced by satellite/real-images. We further show that while aerial and ground modalities *reinforce* each other during training, even a *single* satellite image can *destabilize* training, thereby highlighting the *challenge of* effective multi-altitude integration in current 3D models.

Inspired by these observations, we introduce *SkyNet*, a model designed to strengthen 3D reconstruction and local-



Figure 1. **Cross-view examples from the Sky2Ground dataset.** Satellite, aerial, and ground-level images for a variety of urban scenes in Sky2Ground, where each column corresponds to a unique site. These examples highlight strong viewpoint and appearance variations across modalities, revealing the challenges of cross-view matching and multi-scale scene understanding. Real images additionally introduce diverse lighting conditions, weather effects, and natural scene noise, further emphasizing the complexity of real-world cross-view perception.

ization models *in presence* of satellite imagery. SkyNet is a two-stream architecture consisting of two-components 1) a Ground-Aerial-Satellite (GAS-encoder) which processes all modalities together. It consists of restricted-global-attention module called Masked-Satellite-Attention (MSA) that *prevents* aerial/ground modalities from interacting with satellite. 2) a Sat-Encoder which explicitly processes satellites. Furthermore, SkyNet is trained via novel curriculum-based strategies: 1) gradually sampling views which are further from each other 2) using ‘aerial-images’ as a bridge between ground/satellite modalities. We find that these innovations enable *adapting* and even improving upon a model like VGGT, thereby being able to retain advantage of large-scale pre-training. We reveal that *simple-finetuning* of existing methods[15, 35, 36] *does-not* work, whilst our SkyNet *does not face* this issue.

Our contributions are summarized below:

1. *Sky2Ground Dataset*: A large-scale dataset integrating satellite, aerial, and ground-level imagery with *both* synthetic and real components, *providing annotations* for camera poses and dense depth maps.
2. *First benchmark* studying existing methods on cross-view localization/reconstruction involving *all three* ground/aerial/satellite imagery.
3. *SkyNet Architecture*: which outperforms Dust3r/Mast3r/Map-Anything/VGGT on cross-view localization, and requires *only a single-forward pass*.

2. Related Works

Cross-View Camera Localization: requires inferring camera poses (intrinsics/extrinsics) across *aerial, ground and*

satellite views. Classical SfM pipelines like Colmap[29, 31] face issues in matching-features across such diverse-viewpoints. Additionally, the large-scale (in meters) of captured images, as well as their varying-altitude further exacerbates this problem. Other works [8, 15, 36] instead learn canonical-transformation between a *pairs of views* via neural-nets. However, they require an additional *global-alignment* phase to map all prediction-pairs onto a *consistent* point-map representation. Recently, VGGT[35] has shown that *a single-forward pass* through transformer is enough to infer camera parameters.

In this work, we observe that simply fine-tuning VGGT on combination of *ground/aerial/sat* views *hurts* performance. We propose an alternate method named **SkyNet**, which we hope is a *small-step* towards fixing this non-trivial problem.

Cross-View and Novel-View Synthesis: Recent advances in Neural Radiance Fields (NeRF) [22] and Gaussian Splatting [12] have significantly improved cross-view and novel-view synthesis across diverse capture settings. Several methods leverage aerial imagery [4, 6, 19, 20, 33, 38, 39, 42, 45] to model large-scale scenes [13, 32], while hybrid approaches combine aerial, and ground viewpoints to enhance coverage and geometric stability [7, 9, 11, 24, 43, 44]. Some works [19, 39] incorporate level-of-detail representations to handle varying scales and scene complexity, whereas others [44] fuse separate sub-models for aerial and ground views to improve cross-view consistency. Progressive strategies such as Dragon [7] gradually integrate images captured from different altitudes to stabilize reconstruction.

In contrast, our method performs a unified, full-scale reconstruction that jointly integrates aerial, ground, and satellite observations, demonstrating that improved pose estima-

tion directly translates to more accurate and coherent scene reconstruction.

Table 1. Comparison with existing datasets. A: Aerial, G: Ground, Sat: Satellite, Syn: Synthetic.

Dataset	Sites	Imgs	Syn A	Syn G	Real A	Real G	Sat
KITTI-360 [18]	1	300k	✗	✗	✗	✓	✗
nuScenes [2]	2	1.4M	✗	✗	✗	✓	✗
GauU-Scene [40]	6	4.7k	✗	✗	✓	✗	✗
UC-GS [46]	2	1.4k	✓	✓	✗	✗	✗
BungeeNeRF [39]	12	3.1k	✓	✗	✗	✗	✓
MatrixCity [17]	1	519k	✓	✓	✗	✗	✗
AerialMegaDepth [34]	137	132k	✓	✓	✗	✓	✗
Ours	51	80k	✓	✓	✓	✓	✓

Public Datasets: In Tab 1, we compare across existing datasets in the literature. However, we notice them *lacking* in several-aspects. For eg, nuScenes [2]/ KITTI-360 [18] are autonomous-driving datasets, and lack aerial/satellite views. Similarly, aerial-megadepth[34] contains ground/aerial views, but *lacks* satellite-views. Similarly, synthetically-rendered datasets (using game-engines/unity3d) like Matrix-City, Bungee-Nerf *remain limited* to *synthetic* ground/aerial scenarios. In contrast, our Sky2Ground dataset covers all three ground, aerial, and sat viewpoints, as well as manually-scraped *real* ground/aerial views, indicating its versatility.

To the best of our knowledge there is *no prior-work* that offers: 1) a comprehensive dataset across *all three* viewpoints with *both* real/synthetic images 2) an in-depth study analyzing impact of different viewpoints on camera-localization/rendering.

3. Sky2Ground Dataset

We present Sky2Ground: a dataset which consists of images captured from ground, aerial, and street viewpoints. In Fig 1, we show few samples from our dataset. It consists of several geographic locations, for eg, (*westminster pier, pyramid du louvre, eros*). For each location, we have *both* real/synthetic aerial/ground images, as well as satellite imagery captured at varying-altitude levels. Additionally, we provide densely-annotated depth-maps, and camera intrinsics/extrinsics for *each* captured-image. Next, we delve deeper into dataset-statistics.

Table 2. Sky2Ground dataset statistics. A: Aerial, G: Ground, Sat: Satellite, Syn: Synthetic. -: altitude for real is unknown, due to lack of ground-truth poses.

Attribute	Sat	Syn A	Real A	Syn G	Real G
Modality	Ortho-Rectified	RGB	RGB	RGB	RGB
#Images per site	120	1080	120	50–250	120
Altitude (AGL)	1–2 km	200–800 m	-	5–80 m	-

Dataset Statistics: Sky2Ground consists of 51 geographical locations, Fig 2(right), shows how they span the entire-earth, thereby ensuring diversity. In total, our dataset con-

tains 80k images. Tab 2 provides *individual* stats per-site. Sky2Ground consists of 120 *synthetic* satellite images, 1080 aerial-images, and 50 – 250 ground images. Further, each site contains 120 real ground/aerial images. In Fig 2, we show how the real-images capture weather-conditions, and diverse-lightning (day or night), and altitude levels. Next, we describe the collection-procedure involved in curating our dataset.

Collection-procedure: Sky2Ground consists of both real/synthetic images. We manually-curate *real-imagery* from publicly available Google Maps reviews and YouTube travel vlogs. Next, we filter-them to ensure accurate-coverage of a landmark. Similarly, we collect all the *synthetic data* via Google Earth Studio. However, note that owing to the large degree-of-freedom (*dof*) in 3D, we could capture a particular landmark from *infinite* viewpoints. Therefore, we rely on sampling cameras along different *trajectory-paths* to ensure accurate coverage. Next, we discuss how individual ground, aerial, sat images are curated.

For *ground views*, we simulate a single RGB camera performing a circular walk-around of the landmark. For *satellite Views*, we sample images from an altitude of 1-2Km, thereby getting a top-down coverage of the entire scene. Due to high altitude, the captured images contain perspective-distortions, which we fix via an additional *ortho-rectification* procedure.

Finally, for *aerial viewpoints*, we design a virtual triple-camera rig with left, center, and right views at yaw angles of -20° , 0° , and $+20^\circ$, respectively, following a downward helical trajectory. The trajectory begins around 800 m AGL and descends in stages of 100–150 m, forming three altitude bands: high (800 m), medium (500 m), and low (250 m). Each camera captures 60/120/180 frames per band, totaling 360 images per camera and 1,080 aerial views *per site*.

Annotation procedure: By default, google-earth gives us poses of cameras. However, it lacks dense-predictions such as depth-maps/surface-normals. These in turn are required to supervise a 3D-network like VGGT[35]. Therefore, we run dense-MVS pipeline via ColMAP. Each site takes roughly a day-and-half to run with PatchMatch Stereo, and Stereo-Fusion consuming most of our time.

4. Sky2Ground Benchmark

Here we present Sky2Ground Benchmark where we evaluate existing models for camera localization/rendering-based tasks. We describe the experimental setup, and then present additional analysis.

Experimental setup: In Fig. 3, we show that our dataset comprises of five synthetic splits (CR, D1–D4). Here D1–D4 provide progressively denser-image coverage. Note that core images, (CR) are *common* to all the splits, i.e. (CR, D1–D4). We analyze existing methods on *D2* setup. We *evaluate* on a subset of *D2* images, i.e. the core-setup *CR*. This

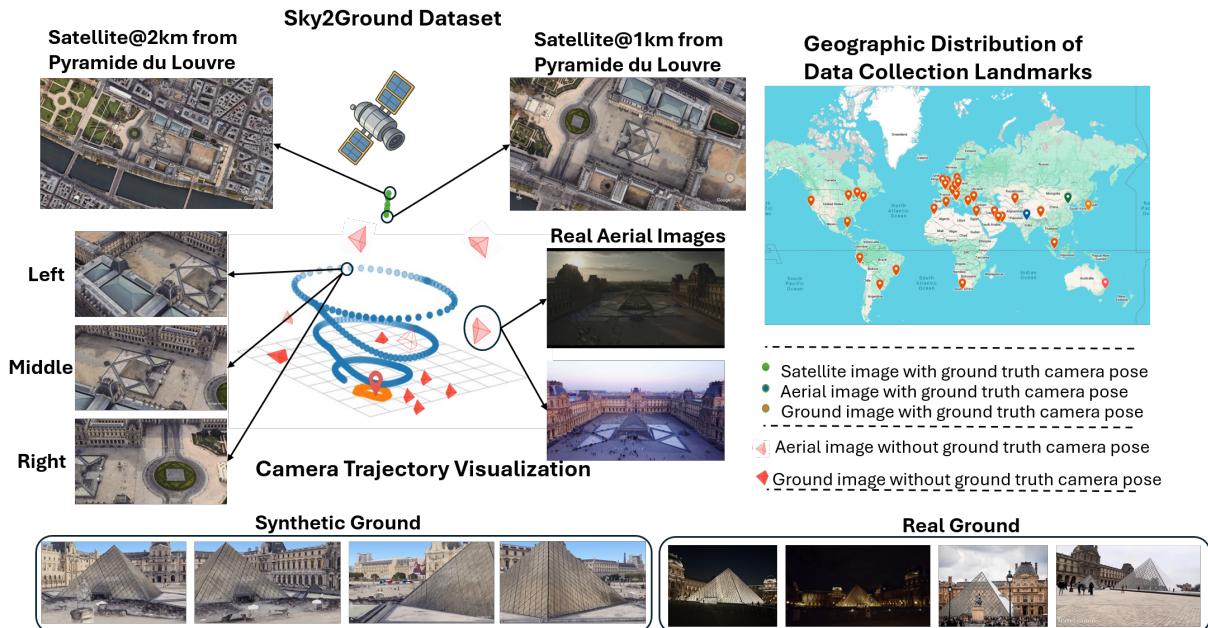


Figure 2. **Overview of the Sky2Ground dataset.** The middle trajectory illustrates camera poses from one of our collected sites. Dots indicate ground-truth camera positions for synthetic images, while red frustums represent the estimated camera poses for real images. The surrounding images showcase example satellite, aerial, and ground views—where the real images demonstrate more diverse illumination conditions and realistic noise. The top-right map depicts the geographic distribution of all collected landmarks, highlighting the dataset’s global coverage.

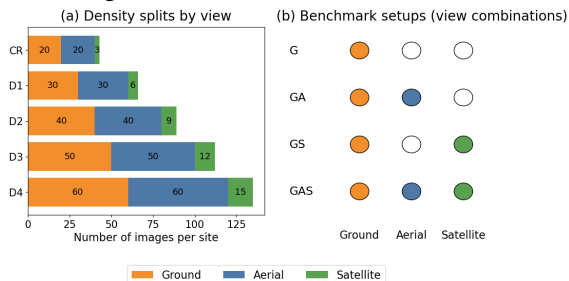


Figure 3. **Benchmark splits and modality setups.** (a) Image counts per split for synthetic CR - Core, D1 - Dense 1, D2 - Dense 2, D3 - Dense3 and D4 - Dense 4, across ground, aerial, and satellite views. (b) View combinations used in each benchmark setup: Ground (G), Ground+Aerial (GA), Ground+Satellite (GS), and Ground+Aerial+Satellite (GAS).

ensures that although the *total number* of images across $D1 - D4$ changes, evaluation setup *stays consistent*. Additionally, we evaluate various gaussian-splatting methods for scene-rendering. Therefore, our analysis covers *both* localization/rendering.

Models. We benchmark a wide variety of *learning-based* methods for camera-localization. For eg, we evaluate *pair-wise view processing* nets like Dust3r/Mast3r/MapAnything. Additionally, we evaluate *VGGT* which infers camera-poses in a single feed-forward. Additionally, we benchmark classical SfM techniques like COLMAP, and provide more results in the supplementary. For rendering, we evaluate models like 2DGS, 3DGS, and 3DGS-MCMC, which span 2D splatting, 3D Gaussian scene representations, and an MCMC-refined 3D variant.

Metrics: We evaluate camera-localization performance via Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA) [10]. More specifically, we threshold at 5° , and report $RRA@5^\circ / RTA@5^\circ$. This denotes, how well the model localizes cameras within 5° of each-other. We quantitatively evaluate photometric/perceptual quality of 3D-renderings via PSNR and DreamSim metrics. A higher value of PSNR is assumed better, whereas for DreamSim *lower-score* is better.

4.1. Zero-shot Analysis

Here, we first discuss about *zero-shot* localization performance of existing models, and variation as number of input views changes. Next, we discuss rendering results.

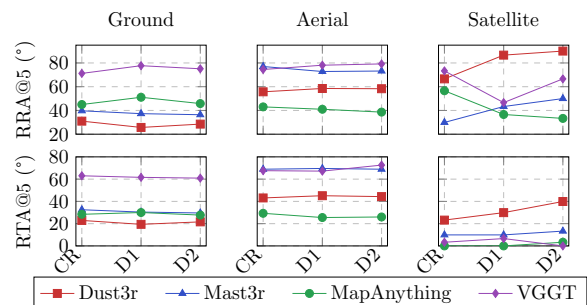


Figure 4. Comparison of $RRA@5$ and $RTA@5$ metrics for four methods (Dust3r, Mast3r, Map Anything, and VGGT).

Models ‘suffer’ when number of views are less: In Fig. 4, we plot how RRA/RTA varies as the number of input views increases *core* \rightarrow *dense1* \rightarrow *dense2*. We

observe models perform *worse* on sparse scenarios, i.e. *core*.

VGGT performs best on Ground/Aerial setup:

We note that VGGT (purple-curve) performs best on Ground/Aerial scenarios. One notable thing is that performance of VGGT *does not fluctuate* much as compared to Dust3r/Mast3r/Map-Anything, indicating that it does well in *sparse-scenarios*.

Pair-wise view networks like Dust3r do well on satellite: In Fig. 4, we note that Dust3r achieves highest performance on localizing *satellite* images (red curve). We believe that this is due to *global-alignment* enforced in dust3r. Satellite-views contain a lot of mutually-common points, and an explicit matching step ensures proper alignment. However, we note that Dust3r/Mast3r *don't* perform well on *ground* scenarios.

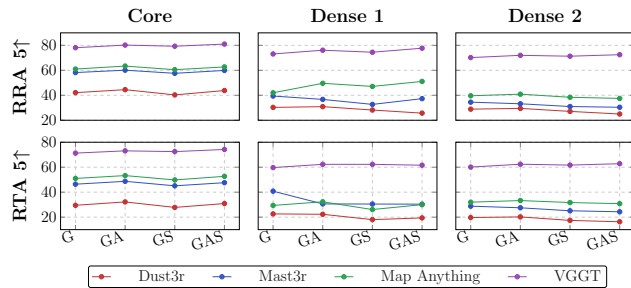


Figure 5. Comparison of models across view combinations.

Satellite images pose a ‘significant-problem’: One expectation might be that models *should perform better* on ground+aerial+satellite (GAS) setup than ground+aerial setup, because adding satellite provides more information to the neural-net. In Fig5, we surprisingly find that this is *not* the case. Instead, we observe *drops* in performance. Therefore, processing *satellite* modality remains a challenge.

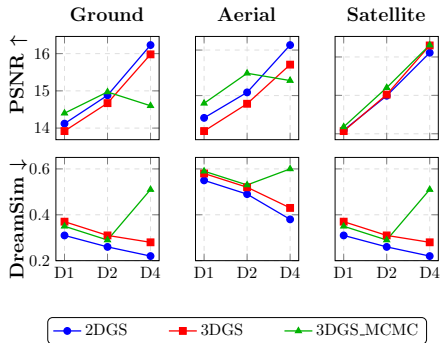


Figure 6. **Comparison of reconstruction quality across ground/aerial/satellite.** We report PSNR \uparrow and DreamSim \downarrow (lower is better). All methods benefit from increased camera density. 2DGS consistently achieves the best perceptual quality.

2D-GS consistently gives best rendering results across ground/aerial/satellite:

Earlier, we noticed that VGGT obtained the *best* performance out of all localization methods. We use VGGT zero-shot predicted poses, to initialize vari-

Table 3. Rendering results across synthetic and real-image setups. We report PSNR and DreamSim for each GS type (2DGS, 3DGS, 3DGS_{MCMC}) across aerial, street, and satellite viewpoints.

Data	GS Type	PSNR \uparrow			DreamSim \downarrow		
		Aerial	Ground	Satellite	Aerial	Ground	Satellite
Synthetic	2D	12.1	14.9	13.0	0.49	0.26	0.19
	3D	11.8	14.7	13.0	0.52	0.31	0.17
	3D _{MCMC}	12.5	15.0	13.3	0.53	0.29	0.21
Syn+Real	2D	11.1	13.9	12.6	0.61	0.40	0.26
	3D	11.1	13.4	12.6	0.64	0.46	0.28
	3D _{MCMC}	11.2	12.8	13.4	0.79	0.71	0.68

ous gaussian splatting methods (2DGS, 3DGS and 3DGS-MCMC). We evaluate performance as number of input views increases from $D_1 \rightarrow D_2 \rightarrow D_4$. The scaling-law (slope) suggests that 2D-Gaussian splatting consistently performs better.

Adding Real images harms rendering and yields noisy reconstructions: Tab. 3 shows that introducing real images into the training mix consistently degrades rendering quality across GS types and viewpoints. Real-world lighting variation, sensor noise, clutter, and textures absent from synthetic data create a strong domain mismatch, making it difficult for the model to blend real and synthetic imagery.

Fig. 7 shows that increasing density from D1 to D4 produces sharper contours and more complete details, while mixing in 30 aerial and 30 ground real images at D2 introduces noise and domain mismatch, especially in ground views where real features absent from synthetic data begin to appear.

4.2. Fine-Tuning-Analysis using Sky2Ground

Earlier we noted that adding satellite during *zero-shot inference* harms performance. In this section, we aim to study if *simply adding* satellite during training, i.e. jointly fine-tuning existing models could work. To facilitate a systematic understanding, we first train on Ground-aerial (GA) data, and then on Ground/Aerial/Satellite data. Out of 51 sites in Sky2Ground, we use 41 for training, and remaining 10 for evaluation.

Fine-Tuning with Ground-Aerial Data improves performance:

Tab 4, shows that training Dust3r on ground-aerial data improves on avg from 58.9 \rightarrow 64.1(RRA), 35.3 \rightarrow 39.3(RTA). Similarly, Mast3r improves from 53.2 \rightarrow 67.6(RRA), 37.3 \rightarrow 48.4(RTA). Next, we study the effect of adding ‘satellite’.

Adding satellite improves ‘pairwise-view’ nets like Dust3r/Mast3r

We note that adding satellite improves performance of nets like Dust3r/Mast3r from 64.1 \rightarrow 65.2 (RRA avg), 67.6 \rightarrow 69.4 (RTA avg) respectively. A similar trend is observed for RTA. An inductive-bias common to *both* Dust3r/Mast3r is that both process a *pair of views* together, which *might* explain this behaviour. However,

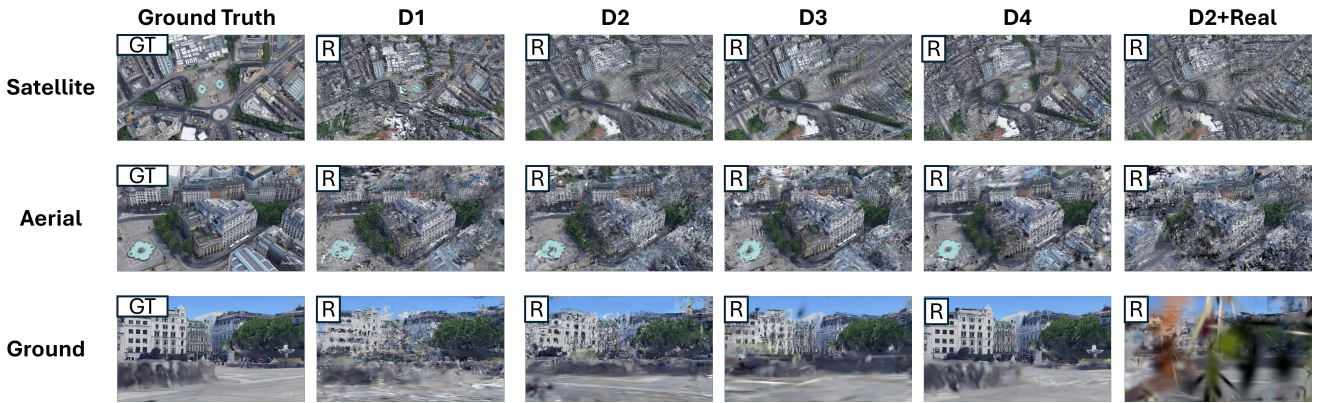


Figure 7. Rendering results across satellite, aerial, and ground viewpoints. Each row shows a different view, with the leftmost column showing the ground-truth (GT) image followed by model renderings (R). From D1 to D4, the input density increases, and real images are incorporated with D2, introducing additional noise and domain mismatch. In particular, the ground-view renderings in D2 + Real contain real-image features that do not exist in the synthetic input distribution, highlighting the challenge of mixing synthetic and real sources.

Table 4. **Localization results across Ground+Aerial+Satellite setup.** ZS = Zero-Shot, A-MD = Aerial-MegaDepth. RRA@5 / RTA@5 measure rotation / translation accuracy within 5° / 5 m. RRA Avg and RTA Avg denote mean performance across all domains.

Method	Data	Ground		Satellite		Aerial		Avg	
		RRA@5	RTA@5	RRA@5	RTA@5	RRA@5	RTA@5	RRA	RTA
Dust3r	ZS	28.5±0.71	21.6±0.53	90.0±1.92	39.9±0.84	58.3±1.37	44.3±1.18	58.9±1.23	35.3±0.79
Mast3r	ZS	36.4±0.86	29.6±0.77	50.0±1.04	13.3±0.28	73.2±1.49	68.9±1.26	53.2±1.07	37.3±0.91
VGGT	ZS	75.1±1.83	60.9±1.29	66.6±1.42	0.0±0.00	79.2±1.67	72.6±1.23	73.6±1.38	44.5±0.95
Dust3r	A-MD	25.5±0.64	23.2±0.71	83.3±1.78	29.9±0.82	83.6±1.74	64.8±1.35	64.1±1.27	39.3±0.92
Mast3r	A-MD	29.1±0.73	24.1±0.61	96.6±2.31	26.6±0.68	77.0±1.59	94.5±2.18	67.6±1.33	48.4±0.97
Dust3r	Sky2Ground	32.4±0.68	27.1±0.63	79.4±1.54	36.3±0.87	83.8±1.79	67.0±1.16	65.2±1.05	43.5±0.94
Mast3r	Sky2Ground	33.2±0.66	27.9±0.59	81.0±1.47	37.2±0.81	85.1±1.71	68.3±1.12	69.4±1.08	47.5±0.89
VGGT	Sky2Ground	50.0±1.25	46.1±1.17	86.6±1.94	53.3±1.41	29.7±0.61	31.5±0.69	55.4±1.09	43.6±0.97
SkyNet	Sky2Ground	76.7±1.52	64.2±1.27	88.9±1.93	57.3±1.14	84.0±1.48	78.1±1.06	83.2±1.37	66.5±0.98

choosing a pair out of N total views requires $\binom{N}{2} \approx O(N^2)$ feed-forwards through the net, thereby making the entire pipeline *extremely* slow. Next, we study the effect of adding ‘satellite’ during training to *single* feed-forward nets.

Adding satellite to ‘single-feedforward’ nets like VGGT severely hurts performance. In Tab4, we observe that VGGT suffers a *severe* drop, i.e. 73.6 → 55.4 (-18.2% RRA), 44.5 → 43.6 (-0.9% RTA). The above analysis leads to a *counter-intuitive* insight. Adding satellite *hurts* VGGT but not Dust3r/Mast3r. If extreme-distribution shift across ground/aerial/sat views were the *only* cause, we should have observed consistent-drops across *all* methods. This leads us to question: ‘How can we further-improve single-feedforward nets like VGGT?’. Building upon VGGT makes sense because we can then *bypass* classical SfM pipelines (aka COLMAP), global-alignment stage, and perform E2E optimization.

5. Improving localization with satellite imagery

As discussed in previous section 4.2, addition of satellite does not work even with fine-tuning. Inspired by this obser-

vation, here we present **SkyNet**, an architecture for cross-view camera localization. First, we formalize our problem statement and describe preliminaries.

Problem statement: Given N RGB images $I = \{I_g, I_a, I_s\}_N$ consisting of $\{I_g\}$ ground-views, $\{I_a\}$ aerial-views and $\{I_s\}$ satellite-views, our objective is to predict the corresponding camera parameters $\{g_i \in \mathbb{R}^9\}_{i=1}^N$ and depth map $\{D_i \in \mathbb{R}^{H \times W}\}_{i=1}^N$. Following VGGT[35], we represent g_i in quaternion notation.

Preliminaries on VGGT: Each of the N input-images is first passed through a dino encoder. The resultant features are then concatenated with additional camera tokens $\mathbf{t}_i^g \in \mathbb{R}^{1 \times C'}$ and four register tokens $\mathbf{t}_i^R \in \mathbb{R}^{4 \times C'}$. The camera/register tokens of first input view ($\mathbf{t}_1^g := \bar{\mathbf{t}}^g$, $\mathbf{t}_1^R := \bar{\mathbf{t}}^R$) are set to a *different* set of learnable tokens ($\bar{\mathbf{t}}^g, \bar{\mathbf{t}}^R$) than those of all other frames ($\mathbf{t}_i^g := \bar{\mathbf{t}}^g$, $\mathbf{t}_i^R := \bar{\mathbf{t}}^R$, $i \in [2, \dots, N]$), which are also learnable.

VGGT consists of L layers of alternating frame-attention and global-attention. First, frame-attention allows tokens of each frame to interact among themselves. Next, in global-attention, information across different-views is aggregated,

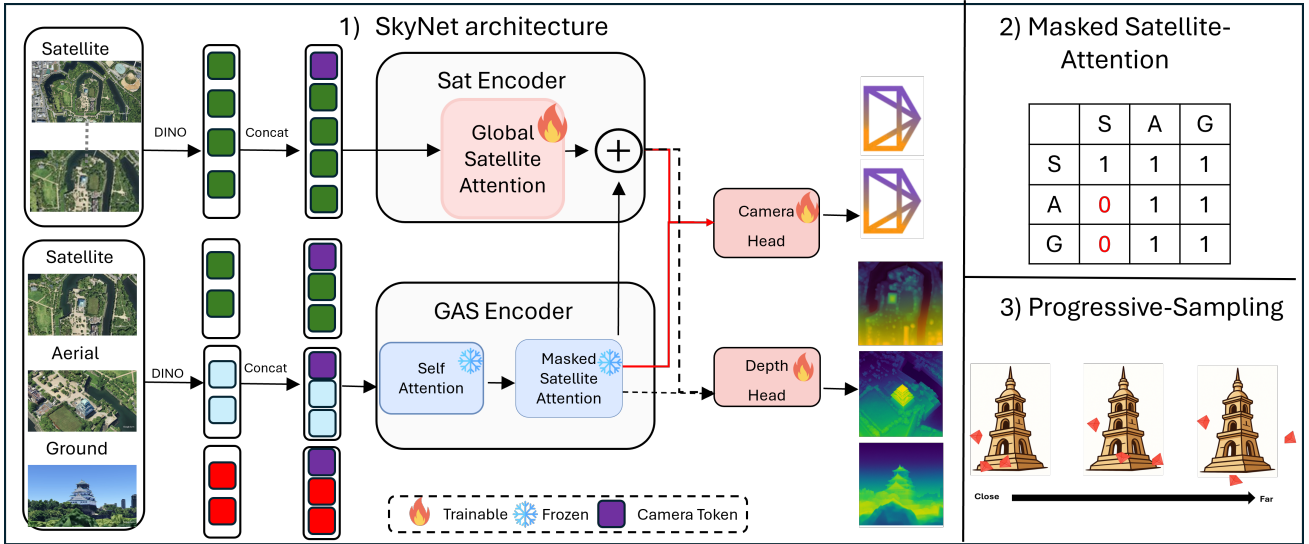


Figure 8. **SkyNet**: An architecture for cross-view camera localization. SkyNet consists of two encoders, 1) a Sat-Encoder processes input satellite images, and a GAS-Encoder processes ground/aerial/satellite views. Our model first patchifies the input images into tokens by DINO, and appends camera tokens for camera prediction. GAS-encoder then alternates between self-attention and Masked-Satellite Attention. Camera/Depth heads collect satellite tokens from Sat-encoder, ground-aerial tokens from GAS-encoder and predict camera-poses and depth-maps respectively. 2) Masked-Satellite Attention *restricts* aerial/ground from interacting with satellite modality. 3) Progressive-sampling gradually samples far-away cameras as training proceeds.

and aligned in the latent space. As in Sec 4.2, we notice two-issues with fine-tuning VGGT 1) It *disturbs* the pre-trained VGGT weights, thereby losing the advantage of large-scale pre-training 2) Global Frame-attention *forces* ground/aerial views to exchange information with satellite, but distribution-shift *hurts* performance.

5.1. SkyNet

Here we propose SkyNet, an architecture that processes input data in two parallel-streams. SkyNet consists of two components 1) a ground-aerial-sat (GAS)-encoder 2) a sat-encoder.

GAS-Encoder: The Ground-Aerial-Satellite (GAS-encoder) consists of L blocks. In each block, *all* input views first undergo a self-attention. Next, we perform *restricted-attention* termed as **Masked-Satellite-Attention (MSA)**. In Fig 8, MSA *restricts* ground/aerial views from interacting with *sat*. However, *sat* still interact with aerial/gnd views. The resulting attention values $V = [v_G, v_A, v_S]$ are passed to $L - 1$ identical blocks. The self-attention, and MSA block are initialized with pre-trained VGGT weights, and kept-frozen. After each block, *sat* features v_S are fed to the sat-encoder.

Sat-Encoder: Given multiple satellite images I_s , we feed-forward them through a dino encoder. The output tokens first undergo a global-attention, where all the *sat*-views interact with each other, resulting in the latent z . Next, we update $z = z + v_s$, where v_s are the refined *sat* features coming from the GAS-encoder.

Camera/Depth-Map Predictions: The camera parameters $(\hat{g}^i)_{i=1}^N$ are predicted from the output camera tokens

$(\hat{t}_i^g)_{i=1}^N$ using four additional self-attention layers followed by a linear layer. This camera-head decodes satellite-poses from the satellite-encoder, and aerial/ground poses from the GAS encoder. Additionally, we leverage a DPT head to output dense features $T_i \in \mathbb{R}^{C \times H \times W}$. Both camera/depth-head are *shared* across GAS encoder/satellite encoder.

Losses: We set the number of transformer layers as $L = 24$. We train SkyNet end-to-end via a multi-task loss $\mathcal{L} = \mathcal{L}_{\text{cam, sat}} + \alpha \mathcal{L}_{\text{cam, gnd/aerial}} + \mathcal{L}_{\text{depth}}$, where we empirically set $\alpha = 0.4$. The camera loss, depth loss, and other hyper-parameters assume similar implementation/initialization as VGGT[35].

5.2. Cross-View Training strategies

As humans, we learn how to solve a particular problem by solving easier instances first, and gradually increasing the level of difficulty. This has been known as ‘curriculum-learning’ in prior-literature. Inspired by this, we propose two training strategies for cross-view localization 1) **Curriculum Aware Camera-Sampling (CA-CS)**, 2) **Progressive-View Sampling (P-VS)**.

CA-CS sampling: During initial epochs, we sample cameras that lie closer to one other. As training proceeds, we gradually sample cameras far away from one another, for eg, non-overlapping cameras. Consider two-views with extrinsics (R_1, R_2) , and (t_1, t_2) . We estimate rotation/translation distance d_R/d_T as:

$$d_R = \frac{1}{\pi} \arccos\left(\frac{\text{Tr}(R_1^\top R_2) - 1}{2}\right), \quad d_T = \|t_1 - t_2\|_2, \quad (1)$$

where, $D(I_1, I_2) = d_R + \lambda_t d_T$, where we set $\lambda_t = 0.5$, balances the rotation and translation terms. We pre-compute

Table 5. **Localization results across Ground+Satellite setup.** RRA@5 / RTA@5 denote rotation and translation accuracy within 5° / 5 m. ZS = Zero-Shot.

Method	Data	Ground			Satellite			Overall Avg
		RRA@5	RTA@5	Avg	RRA@5	RTA@5	Avg	
Dust3r	ZS	29.2	19.1	24.2	90.0	56.6	73.3	48.7
Mast3r	ZS	32.8	31.5	32.2	73.3	33.3	53.3	42.7
VGGT	ZS	70.2	61.6	65.9	76.6	3.3	40.0	52.9
VGGT	Sky2Ground	34.5	40.2	37.4	83.3	33.3	58.3	47.8
SkyNet	Sky2Ground	71.3	64.9	68.1	88.0	36.3	62.2	65.1

and cache pair-wise distances of all the cameras in a scene. For a given ground-view I_1 , all the other views are sorted in increasing order of distance, and sampled with equal offset w.r.t each other. Our distance-metric doesn’t involve explicit-point/feature-matching as in [15, 36], and runs even on a CPU.

P-VS sampling: Intuitively, *ground/sat* views are extreme-viewpoints. However, aerial-images can serve as a ‘bridge’ connecting these modalities together[34]. During training, we sample N total images, where $N = N_a + N_g + N_s$ images, N_a, N_g, N_s being number of ground/aerial/sat images. In beginning of our training schedule, we sample *more number* of aerial-views ($N_a \approx N$). Towards the end, we only retain ground/satellite images ($N_a \approx 0$). Therefore, our SkyNet gradually transitions from an easier problem, (localizing ground/aerial/sat views) to harder scenarios (ground/sat only).

5.3. Results

We evaluate SkyNet on *dense2* setup, and consider *both* ground/aerial/satellite, as well as ground/satellite setup. Note that ground/satellite is an *extremely-realistic* scenario since aerial-imagery might not be available in certain-regions, for eg, forests, and war-zones. We report both RRA@5 and RTA@5, with standard-deviation across 10 sites in Sky2Ground. In Tab4, we observe that SkyNet achieves a sota score of 83.2%(+9.6% RRA), 83.2%(+22.0% RTA). In Tab5, we report $avg = (\frac{RRA@5+RTA@5}{2})$. We observe our SkyNet improves VGGT even further to 65.1 on average, marking improvement of (+12.2%).

5.4. Ablations on SkyNet

In Tab 6, we discuss how different components in SkyNet perform in isolation w.r.t to each other.

Processing satellite along with aerial/ground images helps. Processing *sat* images in *both* Ground-Aerial encoder, and Sat-encoder fares better than processing satellite only in Sat-encoder (+0.9 vs +0.2).

Masked Sat-Attention is the most contributing component: In Tab6, using Masked-Sat attention results in highest

Table 6. **Ablation of SkyNet variants.** Effect of MSA (Masked Satellite Attention), CA-CS (Curriculum-Aware Camera Sampling), and P-VS (Progressive View-Sampling).

Method	GAS-Enc.	Sat-Enc.	MSA	CA-CS	P-VS	Avg
VGGT (trained)	A+G+S	-	✗	✗	✗	47.8
VGGT-ZS	A+G+S	-	✗	✗	✗	52.9
SkyNet (Ours)	A+G	S	✗	✗	✗	53.1 0.2↑
	A+G+S	S	✗	✗	✗	53.8 0.9↑
	A+G+S	S	✗	✓	✗	54.3 1.4↑
	A+G+S	S	✗	✗	✓	61.1 7.3↑
	A+G+S	S	✓	✗	✗	62.7 8.2↑
	A+G+S	S	✓	✓	✗	63.8 10.9↑
	A+G+S	S	✓	✗	✓	64.9 12.0↑
	A+G+S	S	✓	✓	✓	65.1 12.2↑

increase of 8.2. Masking constraint ensures that satellite-tokens are refined by both ground/aerial tokens. Furthermore, we retain strong zero-shot performance of VGGT in ground/aerial case, since ground-aerial tokens *never* interact with input satellite.

Progressive-View Sampling (P-VS) is the best training-strategy: Training SkyNet with *only* P-VS obtains +7.3, whereas with CA-CS only improves by 1.4. This shows that decreasing aerial-views during training is the better strategy.

Adding satellite features from VGGT encoder → Sat-encoder outperforms naive cross-attention. We perform an additional ablation, where instead of ‘adding’ satellite-tokens from the Ground-Aerial encoder, we replace it with cross-attention[3]. Avg performance *drops* from 65.1 to 59.7, indicating that cross-attention is not the best choice.

SkyNet’s improvements are not merely due to increasing-parameters: We create a flop-matched baseline by increasing VGGT to additional 24 layers. This ensures that parameter count becomes *identical* to proposed SkyNet. The performance improves from 47.8 to 50.4. However, it still lies below VGGT’s zero-shot performance (52.9), as well as SkyNet’s results (65.1), indicating that improvements are not *merely* due to increasing parameters, but due to MSA/CA-CS/P-VS components.

6. Conclusion

We introduced the problem of jointly localizing cameras across ground, aerial, and satellite views. We presented Sky2Ground, a dataset spanning both real and synthetic imagery, and showed that naïve fine-tuning of existing models is hindered primarily by the distribution shift introduced by satellite views. To address this, we proposed SkyNet, a cross-view extension of VGGT that leverages curriculum-inspired strategies such as progressive view and camera sampling. We sincerely hope that Sky2Ground may serve as a foundation for future research.

Limitations: Our method is inherently two-stage, in first-stage we predict-poses, and then rely on gaussian-splat for rendering. Exploring *unified-models* capable of predicting poses/gaussian-parameters remains an interesting direction.

Acknowledgments This work was supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0074. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

- [1] Javier Argota Sánchez-Vaquerizo. Urban digital twins and metaverses towards city multiplicities: uniting or dividing urban experiences? *Ethics and Information Technology*, 27(1):4, 2025. [1](#)
- [2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. [3](#)
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. [8](#)
- [4] Shihan Chen, Zhaojin Li, Zeyu Chen, Qingsong Yan, Gaoyang Shen, and Ran Duan. 3d gaussian splatting for fine-detailed surface reconstruction in large-scale scene. *arXiv preprint arXiv:2506.17636*, 2025. [2](#)
- [5] Gianpaolo Conte and Patrick Doherty. An integrated uav navigation system based on aerial image matching. In *2008 IEEE Aerospace Conference*, pages 1–10. IEEE, 2008. [1](#)
- [6] Yuanyuan Gao, Hao Li, Jiaqi Chen, Zhengyu Zou, Zhihang Zhong, Dingwen Zhang, Xiao Sun, and Junwei Han. Citygs-x: A scalable architecture for efficient and geometrically accurate large-scale scene reconstruction. *arXiv preprint arXiv:2503.23044*, 2025. [2](#)
- [7] Yujin Ham, Mateusz Michalkiewicz, and Guha Balakrishnan. Dragon: Drone and ground gaussian splatting for 3d building reconstruction. In *2024 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2024. [2](#)
- [8] Wenmiao Hu, Yichen Zhang, Yuxuan Liang, Yifang Yin, Andrei Georgescu, An Tran, Hannes Kruppa, See-Kiong Ng, and Roger Zimmermann. Beyond geo-localization: Fine-grained orientation of street-view images by cross-view matching with satellite imagery. In *Proceedings of the 30th ACM international conference on multimedia*, pages 6155–6164, 2022. [2](#)
- [9] Lihan Jiang, Kerui Ren, Mulin Yu, Linning Xu, Junting Dong, Tao Lu, Feng Zhao, Dahua Lin, and Bo Dai. Horizon-gs: Unified 3d gaussian splatting for large-scale aerial-to-ground scenes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26789–26799, 2025. [2](#)
- [10] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *International Journal of Computer Vision*, 2020. [4](#)
- [11] Neil Joshi, Joshua Carney, Nathanael Kuo, Homer Li, Cheng Peng, and Myron Brown. Unconstrained large-scale 3d reconstruction and rendering across altitudes. *arXiv preprint arXiv:2505.00734*, 2025. [2](#)
- [12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. [2](#)
- [13] Naoki Kikuchi, Tomohiro Fukuda, and Nobuyoshi Yabuki. Future landscape visualization using a city digital twin: Integration of augmented reality and drones with implementation of 3d model-based occlusion handling. *Journal of Computational Design and Engineering*, 9(2):837–856, 2022. [2](#)
- [14] Ville V Lehtola, Mila Koeva, Sander Oude Elberink, Paulo Raposo, Juho-Pekka Virtanen, Faridaddin Vahdatikhaki, and Simone Borsci. Digital twin of a city: Review of technology serving city needs. *International Journal of Applied Earth Observation and Geoinformation*, 114:102915, 2022. [1](#)
- [15] Vincent Leroy, Johann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. [1](#), [2](#), [8](#)
- [16] Haoyuan Li, Chang Xu, Wen Yang, Huai Yu, and Gui-Song Xia. Learning cross-view visual geo-localization without ground truth. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. [1](#)
- [17] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023. [1](#), [3](#)
- [18] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. [3](#)
- [19] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024. [2](#)
- [20] Yang Liu, Chuanchen Luo, Zhongkai Mao, Junran Peng, and Zhaoxiang Zhang. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. *arXiv preprint arXiv:2411.00771*, 2024. [2](#)
- [21] Roger Marí, Gabriele Facciolo, and Thibaud Ehret. SatNeRF: Learning multi-view satellite photogrammetry with transient objects and shadow modeling using RPC cameras. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1320, 2022. [1](#)
- [22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view syn-

- thesis. *Communications of the ACM*, 65(1):99–106, 2021. [2](#)
- [23] Niluthpol Chowdhury Mithun, Kshitij S Minhas, Han-Pang Chiu, Taragay Oskiper, Mikhail Sizintsev, Supun Samarasekera, and Rakesh Kumar. Cross-view visual geo-localization for outdoor augmented reality. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 493–502. IEEE, 2023. [1](#)
- [24] Rajat Modi, Vibhav Vineet, and Yogesh Rawat. On occlusions in video action detection: Benchmark datasets and training recipes. *Advances in Neural Information Processing Systems*, 36:57306–57335, 2023. [2](#)
- [25] Bhavit Patel, Timothy D Barfoot, and Angela P Schoellig. Visual localization with google earth images for robust global pose estimation of uavs. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 6491–6497. IEEE, 2020. [1](#)
- [26] Dechen Peldon, Saeed Banihashemi, Khuong LeNguyen, and Sybil Derrible. Navigating urban complexity: The transformative role of digital twins in smart city development. *Sustainable Cities and Society*, 2024. [1](#)
- [27] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. [1](#)
- [28] Yoones Rezaei and Stephen Lee. Sat2map: Reconstructing 3d building roof from 2d satellite images. *ACM Transactions on Cyber-Physical Systems*, 8(4):1–25, 2024. [1](#)
- [29] Johannes Lutz Schönberger and Jan Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [2](#)
- [30] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016. [1](#)
- [31] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. [1](#), [2](#)
- [32] Sanjay Somanath, Vasilis Naserentin, Orfeas Eleftheriou, Daniel Sjölie, Beata Stahre Wästberg, and Anders Logg. Towards urban digital twins: A workflow for procedural visualization using geospatial data. *Remote Sensing*, 16(11):1939, 2024. [2](#)
- [33] Jiadong Tang, Yu Gao, Dianyi Yang, Liqi Yan, Yufeng Yue, and Yi Yang. Dronesplat: 3d gaussian splatting for robust 3d reconstruction from in-the-wild drone imagery. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 833–843, 2025. [2](#)
- [34] Khiem Vuong, Anurag Ghosh, Deva Ramanan, Srinivasa Narasimhan, and Shubham Tulsiani. Aerialmegadepth: Learning aerial-ground reconstruction and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. [1](#), [3](#), [8](#)
- [35] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. [1](#), [2](#), [3](#), [6](#), [7](#)
- [36] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. [1](#), [2](#), [8](#)
- [37] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3961–3969, 2015. [1](#)
- [38] YuanZheng Wu, Jin Liu, and Shunping Ji. 3d gaussian splatting for large-scale surface reconstruction from aerial images. *arXiv preprint arXiv:2409.00381*, 2024. [2](#)
- [39] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision*, pages 106–122. Springer, 2022. [1](#), [2](#), [3](#)
- [40] Butian Xiong, Zhuo Li, and Zhen Li. Gauu-scene: A scene reconstruction benchmark on large scale 3d reconstruction dataset using gaussian splatting. *arXiv preprint arXiv:2401.14032*, 2024. [3](#)
- [41] Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kotschieder, Aljaž Božič, et al. Vr-nerf: High-fidelity virtualized walkable spaces. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023. [1](#)
- [42] Zhensheng Yuan, Haozhi Huang, Zhen Xiong, Di Wang, and Guanghua Yang. Robust and efficient 3d gaussian splatting for urban scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26209–26219, 2025. [2](#)
- [43] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–875, 2017. [2](#)
- [44] Chenhao Zhang, Yuanping Cao, and Lei Zhang. Crossview-gs: Cross-view gaussian splatting for large-scale scene reconstruction. *arXiv preprint arXiv:2501.01695*, 2025. [2](#)
- [45] Huiqing Zhang, Yifei Xue, Ming Liao, and Yizhen Lao. Bird-nerf: Fast neural reconstruction of large-scale scenes from aerial imagery. *Scientific Reports*, 15(1):37295, 2025. [2](#)
- [46] Saining Zhang, Baijun Ye, Xiaoxue Chen, Yuantao Chen, Zongzheng Zhang, Cheng Peng, Yongliang Shi, and Hao Zhao. Drone-assisted road gaussian splatting with cross-view uncertainty. *arXiv preprint arXiv:2408.15242*, 2024. [3](#)