

# A-3DGS: Accelerated 3D Gaussian Splatting from Aerial Imagery with SLAM and Sensor-Based Priors

Carl Schierig<sup>1</sup>, Martin Weinmann<sup>2</sup> & Max Hermann<sup>1,2</sup>

## Abstract

In combination with the ease of use and flexibility of unmanned aerial vehicles (UAVs), 3D reconstruction techniques are used to recreate environments across a broad range of domains, including urban monitoring, industrial inspection, and the entertainment industry. The introduction of 3D Gaussian Splatting (3DGS) has enabled high-quality novel view synthesis (NVS) with fast training times. However, existing 3DGS methods either rely on accurate prior camera poses, or struggle to converge on long or complex camera trajectories that are typical for UAV footage. To overcome this, we propose two methods that leverage priors commonly available from drone sensors or simultaneous localisation and mapping (SLAM) systems to efficiently recover camera trajectories from UAV videos. We evaluate these methods against both the original 3DGS using COLMAP poses and subsequent approaches like NoposeGS and CF-3DGS that jointly estimate camera poses and reconstruct the scene. Our results show that both of our proposed methods achieve competitive NVS quality while being significantly faster than existing approaches.

**Keywords** 3D Reconstruction · 3D Gaussian Splatting · Novel View Synthesis · Aerial Imagery · Sensor Data

## 1 Introduction

Creating an explorable, three-dimensional environment from a set of images is a task with applications in several fields, such as 3D reconstruction, visualisation and novel-view synthesis (NVS). The resulting images and geometry can be used in search and rescue missions, when maintaining industrial complexes or for the mapping of large-scale scenes. Other applications include the entertainment industry, where digital environments are used for special effects in movies or as building blocks for the virtual worlds of computer games.

The classical photogrammetry pipeline for reconstructing scenes from a set of images produces textured triangle meshes. In a first step, the relative poses of the images in the scene are reconstructed using Structure-from-Motion (SfM). SfM takes as input a set of images and outputs their position in the scene, a 3D point cloud and parameters of the cameras used to take them (Schönberger & Frahm, 2016). The refined extrinsic and intrinsic parameters can then be used to generate dense meshes, e.g. by using multi-view stereo (Schönberger et al., 2016). Many conventional approaches lack the ability to accurately represent reflections or translucent objects.

Because classical photogrammetry often produces meshes, it is useful in scenarios where the underlying geometry is needed, such as for collision checks or accurate depth. Newer approaches trade this geometric consistency for increased photorealism. Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021) employ a Multilayer Perceptron (MLP) which learns to represent scenes as volumes. While the approach provides excellent NVS results, the use of a large MLP results in slow training and rendering times.

This motivates the introduction of 3DGS (Kerbl et al., 2023). The approach uses anisotropic Gaussians to approximate a volume efficiently. It combines the visual quality of NeRF (Mildenhall et al., 2021) with fast training times.

However, 3DGS (Kerbl et al., 2023) still has limitations, one of those being the time it takes to process the input data. The approach needs accurately estimated intrinsic and extrinsic camera parameters to converge. A prominent approach for reconstructing the camera poses, COLMAP (Schönberger & Frahm, 2016), has a high runtime with respect to the number of input images. Thus, very large 3D reconstructions can still take hours or even days to complete. This makes current approaches infeasible for time-critical applications. Even applications with soft deadlines can profit from faster reconstruction times. For example, reducing the

<sup>1</sup> Fraunhofer IOSB, Karlsruhe, Germany, E-Mail: [carl.schierig, max.hermann]@iosb.fraunhofer.de

<sup>2</sup> Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, E-Mail: [martin.weinmann, max.hermann]@kit.de

time it takes to identify a problem in an industrial complex monitored with drones or robots is going to reduce the time it takes until problems can be identified. This opens a potential for saving resources, freeing them up for other tasks.

All the aforementioned approaches work on image-based data. While early approaches were designed for internet photo collections (Snavely et al., 2006), some newer pose estimation approaches such as ORB-SLAM 1-3 (Mur-Artal et al., 2015; Mur-Artal & Tardós, 2017; Campos et al., 2021) use video streams as their input.

The low costs of drones and their ability to capture footage from many angles and altitudes easily, combined with the high end-to-end reconstruction time of 3DGS (Kerbl et al., 2023), lead to a need for approaches which can reconstruct Gaussian Splatting scenes from aerial imagery at a lower computational cost. This poses the following research question:

*RQ: How can the end-to-end reconstruction times of 3D Gaussian Splatting from aerial imagery be improved upon?*

This article aims to answer the research question by proposing two approaches for accelerating the construction of Gaussian Splatting. The approaches focus on accelerating the camera pose reconstruction step by replacing COLMAP with less resource intensive, but in turn also less accurate alternatives. The first approach uses trajectories reconstructed using ORB-SLAM3 (Campos et al., 2021), whereas the second approach uses trajectories reconstructed from UAV sensor data.

## 2 Related Work

The classical 3D reconstruction process consists of two steps: SfM, as introduced by Snavely et al. (2006), recovers camera poses and a sparse representation of the scene. Modern approaches like COLMAP (Schönberger & Frahm, 2016) make multiple improvements to the state-of-the-art in SfM and are widely used as a baseline for pose estimation in research.

In a second step, a dense scene representation is formed. This can be done with e.g. MVS, such as the approach presented by Schönberger et al. (2016). Many classical techniques closely follow the underlying geometry, e.g. by generating triangle meshes, but lack photorealism and handling of non-diffuse or low-texture areas.

To address these restrictions, Mildenhall et al. (2021) introduced Neural Radiance Fields (NeRFs). They propose an implicit volumetric scene representation encoded in an

MLP which is trained on images depicting the scene. To render novel views, ray marching is used. Because the neural network encodes view-dependent colours, the approach can correctly handle specular reflections. This however comes at the cost of long training and rendering times.

NeRFs have also been used for pose estimation as well as joint pose estimation and scene reconstruction. In iNeRF (Yen-Chen et al., 2021), pose estimation and refinement are performed by rendering an image and propagating the image loss back to the camera transform. MonoNeRF (Fu et al., 2023) uses two encoders to estimate relative camera poses and monocular depth for a video input. NeRFmm (Wang et al., 2022) jointly optimises the scene representation and camera parameters, both intrinsic and extrinsic, on forward facing scenes. BARF (Lin et al., 2021) uses a coarse-to-fine training approach for joint optimisation of scene geometry and extrinsic camera parameters. Last, NoPe-NeRF (Bian et al., 2023) integrates the optimisation of the relative camera transforms between adjacent frames into the scene reconstruction.

To overcome the performance limitations of NeRFs, Kerbl et al. (2023) propose using a set of anisotropic 3D Gaussians to explicitly represent the scene. Compared to previous works, training and rendering 3D Gaussian Splatting (3DGS) scenes is considerably faster, enabling novel applications of the technology. Several approaches follow up on 3DGS to improve scene reconstruction times and handling of large-scale scenes.

For more complex scenes, 3DGS generates millions of Gaussians, leading to large file sizes and high memory consumption during training and rendering. Lee et al. (2024), Papantonakis et al. (2024), Fan et al. (2024) and Girish et al. (2024) propose pruning strategies to reduce the number of Gaussians without sacrificing image quality.

Improving the Gaussian Splatting renderer can lead to major improvements in training times as in each iteration, a forward and a backward pass of a frame is rendered using a differentiable renderer. Diels et al. (2025) present four improvements to the 3DGS rendering pipeline. Mallick et al. (2024) propose faster equations for gradient backpropagation as well as updating Spherical Harmonics parameters less frequently. Armagan et al. (2025) combine multiple pruning and progressive training approaches to cut both rendering and training times in half. Last, Durvasula et al. (2023) point out a bottleneck in the differentiable renderer. They leverage the structure of GPU architectures to speed up rendering.

Another area of interest in 3DGS research is inverse rendering, i.e. reconstructing the underlying geometry from

scenes reconstructed with 3DGS. Huang et al. (2024) use rotated 2D Gaussians instead of 3D Gaussians in their scene representation to more closely follow the underlying geometry. In a similar manner, Guédon & Lepetit (2024) introduce a regularisation term which encourages 3D Gaussians to shrink along one axis, making them flatter. In addition, Chen et al. (2024) introduce another regularisation term which enforces consistent geometry from multiple views. Lyu et al. (2024) learn a signed distance field of the scene surface alongside the training of the 3DGS representation.

Large scenes typically have higher computational and spatial demands, which need to be addressed when using 3DGS to reconstruct them. Lin et al. (2024) partition the scene into multiple cells, which are reconstructed in parallel and merged to create one large scene. In addition to this, Liu et al. (2024) also introduce a Level-of-Detail system for efficient rendering of large-scale scenes. Opting for a different approach, Ham et al. (2024) reconstruct buildings with drone and ground footage by adding generated views from intermediary elevations to the training set.

While all these approaches optimise or improve upon 3DGS in some way, they all require accurate camera poses, usually computed with an offline SfM approach. Even without additional optimisations to the 3DGS training step, a significant amount of the total reconstruction time is spent on SfM. On the other hand, faster pose reconstruction methods, such as SLAM approaches, produce less accurate poses, making them infeasible for use in the 3DGS reconstruction pipeline.

This is where NoposeGS (Schmidt et al., 2024) and CF-3DGS (Fu et al., 2024) step in. These approaches perform joint pose estimation and scene reconstruction by estimating the relative poses between subsequent camera frames. CF-3DGS (Fu et al., 2024) performs this step on a local scene initialised using monocular depth estimation before adding the camera to a global scene, which is progressively grown as more camera frames are added. NoposeGS (Schmidt et al., 2024) integrates the optimisation of extrinsic camera parameters directly into the backpropagation step by encoding the position and orientation of each camera in a differentiable space of transforms. Moreover, they introduce an anisotropy loss term and add additional regularisation to the pruning algorithm to ensure that Gaussians better follow scene geometry. This enables them to not only perform joint pose estimation and scene reconstruction, but also joint pose refinement and reconstruction. A newer approach, TrackGS (Shi et al., 2025), jointly estimates both extrinsic and

intrinsic camera parameters during reconstruction. The approach is limited to pinhole cameras.

However, NoposeGS and CF-3DGS struggle with reconstructing longer and more complex trajectories, as often produced by UAVs. We thus propose to leverage additional priors to estimate poses quickly for use with 3DGS.

For this, one of our approaches uses trajectories generated with ORB-SLAM3 (Campos et al., 2021), a keyframe-based SLAM approach which uses ORB features (Rublee et al., 2011). ORB-SLAM3 builds upon ORB-SLAM (Mur-Artal et al., 2015) and ORB-SLAM2 (Mur-Artal & Tardós, 2017). On top of its predecessors, it introduces additional handling for arbitrary camera models and multiple maps.

### 3 Methods

Although poses reconstructed by COLMAP are accurate, the approach is slow especially on larger scenes, taking approximately 50% of the total 3DGS reconstruction time in our experiments. Thus, replacing COLMAP with a faster method for computing camera poses will result in an overall speed-up if the 3DGS reconstruction of the scene is not slower than in the original approach.

The downside is that faster pose reconstruction methods often trade accuracy for speed. Because accurate poses are needed for the original 3DGS to converge, poses calculated by these faster approaches often require additional refinement.

#### 3.1 Joint Pose Refinement and Scene Reconstruction

Instead of first refining the poses and then reconstructing the scene with the original 3DGS (Kerbl et al., 2023), this work employs NoposeGS (Schmidt et al., 2024) to jointly refine the camera poses and reconstruct the scene.

NoposeGS integrates the pose refinement directly into the differential CUDA renderer by Kerbl et al. (2023). The integration into the differential renderer enables joint pose refinement at no significant additional computational cost. In fact, on most scenes, NoposeGS is faster than 3DGS, likely caused by two additional regularisation terms introduced to improve pose convergence by improving geometric convergence.

Without additional regularisation, Gaussians do not necessarily follow the underlying geometry of the scene closely. As this is needed for the poses to converge, the

regularisation terms aim to improve the geometric properties of the reconstructed scene.

The first regularisation term limits the maximum anisotropy of the Gaussians. The goal of this term is to prevent the scene from converging to a local minimum too quickly. The second term introduces an additional pruning strategy, capping the number of Gaussians during the first 10,000 iterations of training. While this leads to better poses and lower training times, it has a negative impact on the NVS results.

As CF-3DGS (Fu et al., 2024) cannot perform joint pose refinement and scene reconstruction, we chose NoposeGS as the 3DGS reconstruction approach.

## 3.2 Faster Camera Pose Reconstruction

We propose two different approaches for quickly obtaining estimated camera poses from UAVs. The first approach, *ORB-SLAM-NoposeGS*, uses poses obtained from monocular SLAM. The second, *Sensor-DPT*, uses GPS and north-aligned orientation data provided by the UAV to initialize the camera poses.

### 3.2.1 Camera Poses from Monocular SLAM

SLAM approaches reconstruct camera poses in real-time, while simultaneously constructing a map of the surroundings. We use ORB-SLAM3 (Campos et al., 2021) to estimate the camera trajectories. ORB-SLAM3 takes as inputs the ordered set of images of the video along with the intrinsic calibration of the camera used to capture it.

ORB-SLAM3 is a keyframe-based approach. Instead of registering every frame into the camera trajectory, it selects keyframes from the input stream and only estimates their trajectory. For a frame to be selected as keyframe, the following criteria must be met (Mur-Artal et al., 2015):

- 20 frames have passed since the last keyframe
- Frame must contain at least 50 features
- Less than 90% of features must be shared between frame and last keyframe

The approach outputs the extrinsic calibration of the selected keyframes and a sparse point cloud which follows the 3D geometry of the scene. The intrinsic camera calibration was estimated using COLMAP.

Because the intrinsic calibration of the camera is neither estimated by ORB-SLAM3 nor provided by the drone, it needs to be estimated separately.

The intrinsic and extrinsic calibration along with the point cloud are then passed to NoposeGS (Schmidt et al.,

2024) for scene reconstruction. NoposeGS provides two modes of operation. The first integrates pose refinement into the 3DGS approach by Kerbl et al. (2023). The second prepends joint pose estimation to the scene reconstruction by estimating the relative poses between adjacent images. The relative poses are then combined to an initial trajectory which is used for the joint pose refinement and scene reconstruction step.

For this approach, the joint scene reconstruction and pose refinement is used without refining the pose separately first. The results of this step are the reconstructed scene and the refined trajectory.

### 3.2.2 Camera Poses from UAV Sensor Data

UAVs often provide additional geospatial sensor data along with the footage. The first approach uses GPS, altitude and orientation data provided by the UAV.

For the chosen input frames, the data is read and converted into the COLMAP (Schönberger & Frahm, 2016) format. To avoid floating point precision issues, the images are placed into a local metric coordinate system whose origin lies at the centre of the trajectory. As in the previous approach, camera intrinsics are provided separately and are not computed as part of the reconstruction process.

The scene is then reconstructed using NoposeGS (Schmidt et al., 2024). Contrary to the previous approach, no point cloud is available to initialize the Gaussians in the scene. To circumvent this, we generate a point cloud by unprojecting monocular depth generated with DPT (Ranftl et al., 2021) into the scene.

## 4 Results

### 4.1 Experimental Setup

The proposed approaches are tested and compared against other approaches on videos from the following three datasets: *JB3D*, *Nordic* and *TMB-IPF*. The datasets were captured using different UAVs and consist mainly of orbital flights in oblique view at flight altitudes of 2m – 60m. The three datasets contain challenging elements such as fine structures or reflective surfaces. Only JB3D has sensor metadata for all videos. A 30-fps reference trajectory was created for all videos of the three datasets using COLMAP. Table 1 contains the videos selected from each dataset.

For the experiment, all high-resolution images were scaled down to a resolution of 1920×1080 pixels. Due to the high memory and computational demands of CF-3DGS (Fu

et al., 2024), the approach was evaluated using a quarter of that resolution, i.e. 960×540 pixels.

**Table 1** Scenes selected for evaluation

Dataset	Name	Length [mm:ss]
JB3D	Main Stage I	01:04
	Main Stage II	01:24
	Entrance I	03:06
	TV Tower	01:25
	Gardens II	00:43
Nordic	Ruin Roof	01:50
	Ruin	00:59
	Silo	03:50
TMB-IPF	Container 2m	01:06
	Area 10m	01:42

The train/test split follows the methodology used in 3DGS (Kerbl et al., 2023). Thus, every eighth image is used as a test image.

To ensure a fair comparison regarding image quality and reconstruction speed, all methods are evaluated on the keyframes selected by ORB-SLAM3 (Campos et al., 2021). This enables all approaches to use the same train and test image set. One of the baseline approaches, CF-3DGS (Fu et al., 2024), uses a slightly different metric for splitting the images into a train and a test set, but the size of each set remains the same. Since no orientation data is available for the *Nordic* and *TMB-IPF* datasets, they will not be evaluated with Sensor-DPT.

ORB-SLAM-NoposeGS uses the parameters given in Table 2. Sensor-DPT uses a lower initial camera learning rate of  $1.5e-3$  and a lower position learning rate going from  $1.6e-4$  to  $1.6e-5$ .

**Table 2** Parameters for ORB-SLAM-NoposeGS

Parameter	Value	
Iterations	30,000	
Densify until iteration	30,000	
Camera LR	Initial	$1e-3$
	End	$1e-5$
	Max Steps	30,000
Position LR	Initial	$1e-3$
	End	$1.6e-5$
Opacity Loss Weight	0.01	
Anisotropy Max Ratio	10	
Initial Max Gaussian Count	256,000	

### 4.1.1 Metrics

To compare the image quality, Structural Similarity Index Measure (SSIM) (Wang et al. 2004), Peak Signal-to-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) are used. The PSNR is defined as

$$PSNR(x, y) = 10 \cdot \log_{10} \left( \frac{MAX_I(x, y)^2}{MSE(x, y)} \right).$$

The total reconstruction time is measured as the sum of the pose reconstruction and scene reconstruction time. Where possible, the pose reconstruction and scene reconstruction times were also recorded separately, though this was not possible for joint reconstruction approaches where the poses are constructed alongside the scene. Pose reconstruction time includes the time it takes to generate the point cloud where the initial Gaussians are placed. This is important for Sensor-DPT, where DPT accounts for nearly the whole pose reconstruction time.

The quality of the trajectory is evaluated using the Absolute Trajectory Error (ATE), Absolute Position Error (POS) and Absolute Rotation Error (ROT) metrics.

Given a ground truth trajectory  $C_i = (t_i, R_i)_{i=1, \dots, n}$  and an aligned estimated trajectory  $\hat{C}_i = (\hat{t}_i, \hat{R}_i)_{i=1, \dots, n}$  with  $t_i, \hat{t}_i \in \mathbb{R}^3$  and  $R_i, \hat{R}_i \in SO(3)$ , the trajectory metrics are computed as follows:

$$ATE = \sqrt{\frac{1}{n} \sum_i \|t_i - \hat{t}_i\|_2^2}$$

$$POS = \text{mean}\{\|t_i - \hat{t}_i\|_2\}$$

Let  $\theta_i$  be the angle, in degrees, between the orientation of cameras  $C_i$  and  $\hat{C}_i$ .

$$ROT = \text{mean}\{\theta_i\}$$

The trajectories are aligned and evaluated using the *evo* Python library (Grupp, 2017). All distance metrics are measured in meters, all angular metrics in degrees.

### 4.1.2 Hardware

All results were computed on an Nvidia L40 GPU with 48GB VRAM. The total amount of CPU cores available to the approaches was limited to 40. Due to the large amounts of data handled, the data was stored on a network filesystem. This may affect the time it took to load the images into memory.

## 4.2 Evaluation

Our experiment evaluation reveals that both of our proposed approaches are consistently faster than existing approaches.

**Table 3** Quantitative Results on the *JB3D* Dataset. The results for CF-3DGS are only averaged over three scenes, as the approach does not converge on the other videos.

Method	Image Quality			Pose Quality			Time [hh:mm:ss]		
	PSNR↑ [dB]	SSIM↑	LPIPS↓	POS↓ [m]	ROT↓ [°]	ATE↓ [m]	Poses↓	Scene↓	Total↓
COLMAP + 3DGS	<b>31.219</b>	<b>0.916</b>	<b>0.122</b>	-	-	-	00:29:10	00:30:30	00:59:41
CF-3DGS (540p)	22.536	0.610	<u>0.253</u>	4.732	12.283	5.567	-	-	02:20:10
NoposeGS	20.801	0.522	0.546	33.365	51.551	35.794	-	-	00:35:48
ORBSLAM-NoposeGS	<u>26.735</u>	<u>0.785</u>	0.290	<b>0.318</b>	<u>1.148</u>	<b>0.344</b>	<u>00:05:20</u>	<u>00:19:38</u>	<u>00:24:58</u>
Sensor-DPT	24.710	0.698	0.372	<u>0.615</u>	<b>0.856</b>	<u>0.666</u>	<b>00:00:31</b>	<b>00:17:12</b>	<b>00:17:43</b>

While no other approach can match the quantitative results of the original 3DGS (Kerbl et al., 2023) with COLMAP (Schönberger & Frahm, 2016) poses in NVS, ORBSLAM-NoposeGS and Sensor-DPT follow closely. They produce competitive results while being faster than all other existing approaches.

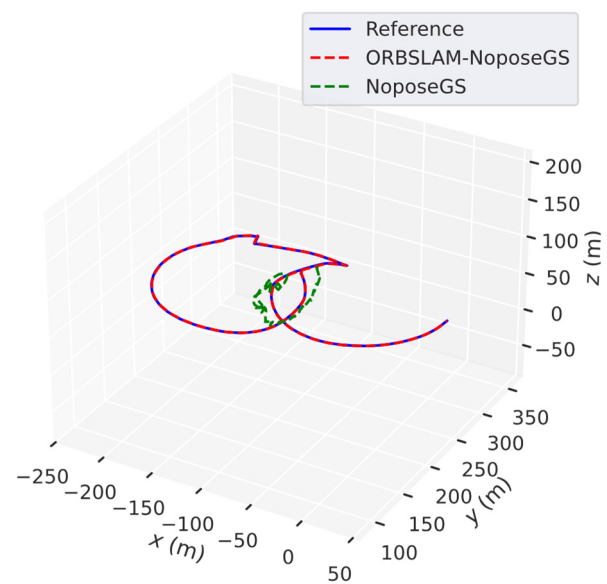
Table 3 contains averaged quantitative results for the *JB3D* dataset.

#### 4.2.1 Scene Quality

COLMAP+3DGS (Kerbl et al., 2023) produces the highest quality results. From the set of approaches using no or noisy initial trajectories, ORBSLAM-NoposeGS produces the most competitive results compared to the original 3DGS by Kerbl et al. (2023). Sensor-DPT fails to pick up some details which ORBSLAM-NoposeGS manages to capture but comes out ahead of CF-3DGS and NoposeGS. The approach is presumably held back by the lack of a good point cloud for initialising the Gaussians and noise in the GPS data. Figures 2 and 3 contain renderings comparing the different approaches on the set of evaluated scenes.

For the other evaluated baseline approaches, the quality of the reconstructed scenes heavily depends on the complexity and length of the trajectory. While both approaches converge on simple scenes such as *Main Stage I*, NoposeGS (Schmidt et al., 2024) does not converge on *Main Stage II*, where the drone flies in an outwards spiral. CF-3DGS (Fu et al., 2024) on the other hand does not finish reconstructing the scene on longer scenes like *Main Stage II* or *Entrance I*. This is caused by the high computational requirements of CF-3DGS. The Nvidia L40 GPU on which the experiment was run does not have enough VRAM for CF-3DGS on long scenes.

#### 4.2.2 Trajectories

**Figure 1** Comparison of the Trajectories of ORBSLAM-NoposeGS and NoposeGS on *Entrance I*

The quality of the trajectories is evaluated by aligning them to and comparing them with metric COLMAP trajectories.

Of the evaluated approaches, ORBSLAM-NoposeGS has the most accurate trajectory after pose refinement. While the trajectory reconstructed using sensor data has a lower absolute position error, the relative error between adjacent camera frames is higher. This indicates that the sensor data varies more between images but does not suffer from drift as ORB-SLAM3 (Campos et al., 2021) does. The drift in turn leads to a higher absolute error, because it leads to less



**Figure 2** Renderings of test cameras. From left to right: *Main Stage I*, *Main Stage II*, *Entrance I*, *TV Tower*, *Gardens II*. Some images are missing due to CF-3DGS running out of resources during scene reconstruction.

overlap between the estimated trajectory and the reference trajectory.

On all scenes, trajectories estimated by CF-3DGS and NoposeGS have significantly worse quality. Figure 1 plots a trajectory estimated by NoposeGS against the refined

ORB-SLAM-NoposeGS and Reference (COLMAP by Schönberger & Frahm (2016)) trajectories.

On the *Silo* and *Area 10m* scenes, ORB-SLAM3 loses tracking. This is an unavoidable drawback when dealing with SLAM approaches. We are therefore only able to

reconstruct the scenes using the keyframes from the second part of the videos.

### 4.2.3 Speed-Up

As mentioned above, both approaches presented in this article are consistently faster than the baseline approaches. Between these, Sensor-DPT is often faster than ORBSLAM-NoposeGS. The differences in speed between our approaches is mostly caused by the differences in pose and point cloud reconstruction time. Thus, the magnitude of the speed-up depends on the length of the scene, with longer scenes seeing a larger difference.

NoposeGS is the next-fastest approach. Notably, 3DGS using COLMAP poses is still faster on long scenes than CF-3DGS on short ones.

## 5 Discussion

The two presented approaches introduce novel initialisations of both camera poses and Gaussians for use in 3DGS scene reconstruction. The evaluation shows that both ORBSLAM3 (Campos et al., 2021) and the use of sensor data is promising for initialising 3DGS reconstruction at a lower computational cost than COLMAP (Schönberger & Frahm, 2016). While using sensor data currently leads to lower quality results, the approach still outperforms several prior works and highlights the possible time savings.

This section aims to discuss different aspects of the results as well as potential limitations.

### 5.1 Scene Initialisation for Sensor Trajectory

Both the original 3DGS (Kerbl et al., 2023) and NoposeGS (Schmidt et al., 2024) initialise the Gaussian scene by placing a Gaussian at each point in a point cloud. When reconstructing the camera trajectory from sensor data, no point cloud is available. Thus, an additional approach must be used to create the point cloud from which the Gaussians are initialised.

Sensor-DPT uses DPT (Ranftl et al., 2021) for unprojecting monocular depth into the scene. While this enables us to quickly generate an initialisation for scene reconstruction with NoposeGS, monocular depth is not multi-view consistent. Moreover, as no exact scene bounds are known, learning rates cannot be scaled with the scene size. Instead, the initial camera trajectory is scaled to fit inside the unit cube. This could possibly be solved with newer monocular depth estimators such as MoGe (Wang et al. 2025b) or Video Depth Anything (Chen et al., 2025).

Together with the local noise in the sensor data, this results in Sensor-DPT producing slightly worse results than ORBSLAM-NoposeGS. Nonetheless, the approach remains promising as it is not prone to losing tracking or camera drift as is the case for trajectories reconstructed with ORBSLAM3 (Campos et al., 2021).

### 5.2 Choice of Input Images

For the evaluation, all scenes were reconstructed using the keyframes chosen by ORBSLAM3 (Campos et al., 2021).

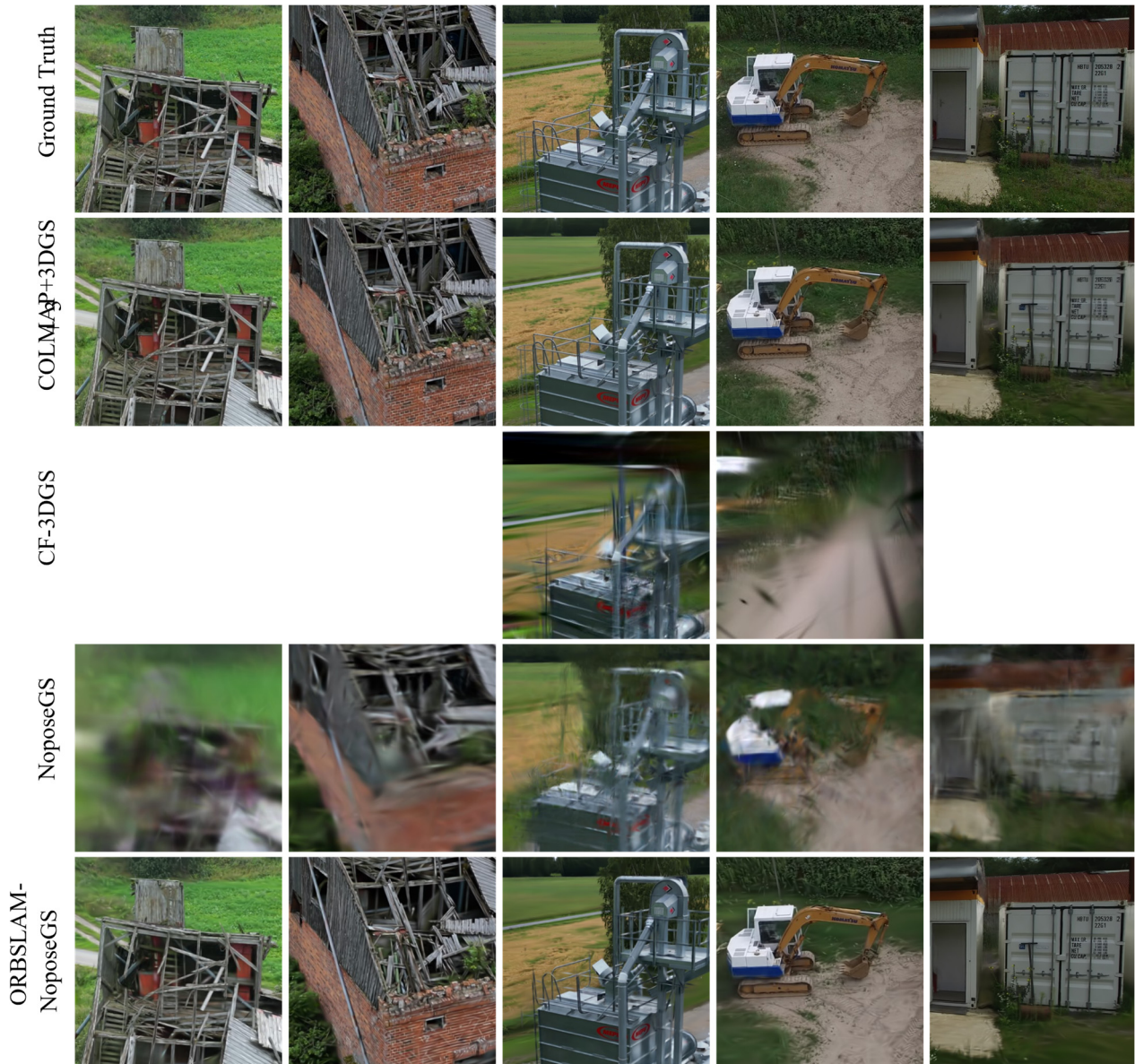
The downside of this is that it shares information gathered by ORBSLAM3 with other approaches. If the algorithm used by ORBSLAM3 to select keyframes discards frames without new 3D information, less frames are selected when the camera is not moving. In such scenarios, using only the keyframes would result in lower end-to-end reconstruction times for other approaches compared to a naive implementation, e.g. selecting every 10th frame.

On the other hand, if the camera moves rapidly in low-texture environments, ORBSLAM3 might not be able to track the camera accurately, selecting more keyframes to account for the rapid camera movement. In this scenario, a naive metric might choose fewer images, leading to slower reconstruction times in the evaluation.

In the end, using the same images for all approaches was deemed superior as it results in a better comparability of the quantitative results. It enables all approaches to use the same set of training and testing images, enabling direct comparison of image metrics such as SSIM or PSNR. Moreover, it provides a comparison on the runtime of the different approaches in relation to the number of images used.

### 5.3 Precomputation of Camera Intrinsic

Current approaches for joint pose refinement and scene reconstruction require estimated camera intrinsics. This includes NoposeGS, which is used to reconstruct scenes in this article, and ORBSLAM3, which is used to reconstruct the camera poses in the first approach.



**Figure 3** Renderings of test cameras. From left to right: *Ruin Roof*, *Ruin*, *Silo*, *Area 10m*, *Container 2m*. Some images are missing due to CF-3DGS running out of resources during scene reconstruction

The camera intrinsics for the evaluation were estimated using COLMAP. The time this takes is not included in the measurements (see Table 3). The decision to not include the camera intrinsics computation in the timings was made to emphasize the improvement in the parts of the reconstruction pipeline which were modified in our proposed approaches.

Nonetheless, faster estimation of camera intrinsics would have a positive impact on the feasibility of fast reconstruction of scenes using 3DGS. Until then, a solution

is to estimate the intrinsic camera parameters only once for each UAV.

All in all, estimating camera intrinsics remains a difficult problem, especially when using camera models with additional distortion parameters. In the future, approaches such as VGGT (Wang et al., 2025a) or Depth Anything 3 (H. Lin et al., 2025) might address this, but as of now, they are not accurate enough to be used with 3DGS.

## 5.4 Limitations

The applicability of ORBSLAM-NoposeGS is limited mainly by the robustness of ORB-SLAM3. Complex or fast camera movements as well as flights over low texture or highly reflective areas might have a negative impact on the quality on the output of ORB-SLAM3, which would in turn reduce the quality of the reconstructed scene. This can be seen in the evaluation of the *Area 10m* and *Silo* scenes, where ORB-SLAM3 lost tracking and constructed multiple trajectories. Only the last trajectory was used for scene reconstruction in the evaluation.

## 6 Conclusion

While 3DGS (Kerbl et al., 2023) provides high quality NVS results together with low training times, the requirement for accurately initialised camera poses leads to long end-to-end scene reconstruction times. Together with the increased adoption of UAVs for 3D reconstruction tasks, this introduces a need for fast end-to-end reconstruction of environments captured using UAVs. This article proposes to address this by introducing two novel fast approaches for initialising camera poses when reconstructing scenes from aerial imagery using 3DGS.

The first approach, ORBSLAM-NoposeGS, reconstructs camera poses and point clouds using ORB-SLAM3 (Campos et al., 2021). It produces competitive NVS results and accurate trajectories in a fraction of the time needed by the original 3DGS (Kerbl et al., 2023) and other approaches (Schmidt et al., 2024; Fu et al., 2024).

The second approach, Sensor-DPT, uses sensor data commonly provided by UAVs to initialise the trajectory at no computational cost. To initialise the Gaussian scene representation, the approach uses monocular depth estimated using DPT (Ranftl et al., 2021). The trajectories reconstructed from sensor data are not prone to drift or loss of tracking, but suffer from increased local noise, leading to slightly worse overall results when compared to ORBSLAM-NoposeGS. Nonetheless, the potential time savings and other benefits of the sensor data make it a promising approach once the remaining issues with it have been addressed.

For the best performance, it is advised to combine the fast initialisation with other works aiming to improve reconstruction times of 3DGS.

## 7 Outlook

Apart from addressing the limitations raised in the discussion, future work aiming to improve reconstruction times for 3DGS further by addressing the initialisation could work on producing better results with sensor data obtained from UAVs, such as GPS and orientation data. Although it proved to be less precise than ORB-SLAM3 during the evaluation, it has some inherent advantages, such as not being prone to camera drift in longer scenes. Moreover, experimenting with different SLAM approaches could also provide better quality or faster results.

Another promising idea would be to combine the benefits of SLAM approaches and sensor data by using the sensor data during SLAM reconstruction.

Last, more robust pose refinement during 3DGS optimisation would enable the use of even more pose reconstruction methods for initialising Gaussian Splatting.

## References

- Armagan, A., Saà-Garriga, A., Manganelli, B., Nowak, M., & Yucel, M. K. (2025). Trick-GS: A Balanced Bag of Tricks for Efficient Gaussian Splatting. Proc. IEEE Internat. Conference on Acoustics, Speech and Signal Processing, 1–5. <https://doi.org/10.1109/ICASSP49660.2025.10889395>
- Bian, W., Wang, Z., Li, K., & Bian, J.-W. (2023). NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior. Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4160–4169. <https://doi.org/10.1109/CVPR52729.2023.00405>
- Campos, C., Elvira, R., Rodríguez, J. J. G., M. Montiel, J. M., & D. Tardós, J. (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. IEEE Transactions on Robotics, 37(6), 1874–1890. <https://doi.org/10.1109/TRO.2021.3075644>
- Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., & Zhang, G. (2024). PGSR: Planar-based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction. IEEE Transactions on Visualization and Computer Graphics, 1–12. <https://doi.org/10.1109/TVCG.2024.3494046>
- Chen, S., Guo, H., Zhu, S., Zhang, F., Huang, Z., Feng, J., & Kang, B. (2025). Video Depth Anything: Consistent Depth Estimation for Super-Long Videos. Proc. IEEE Conference on Computer Vision and Pattern Recognition 22831–22840. [https://openaccess.thecvf.com/content/CVPR2025/html/Chen\\_Video\\_Depth\\_Anything\\_Consistent\\_Depth](https://openaccess.thecvf.com/content/CVPR2025/html/Chen_Video_Depth_Anything_Consistent_Depth)

- Estimation\_for\_Super-Long\_Videos\_CVPR\_2025\_paper.html
- Diels, L., Vlaminck, M., Philips, W., & Luong, H. (2025). Fast 3D Gaussian Splatting Rendering via Easily Integrable Improvements. *IEEE Signal Processing Letters*, 32, 381–385. <https://doi.org/10.1109/LSP.2024.3521379>
- Durvasula, S., Zhao, A., Chen, F., Liang, R., Sanjaya, P. K., & Vijaykumar, N. (2023). DISTWAR: Fast Differentiable Rendering on Raster-based Rendering Pipelines (No. arXiv:2401.05345). arXiv. <https://doi.org/10.48550/arXiv.2401.05345>
- Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., & Wang, Z. (2024). LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS. *Proc. Advances in Neural Information Processing Systems*, 140138-140158. <https://doi.org/10.48550/arXiv.2311.17245>
- Fu, Y., Misra, I., & Wang, X. (2023). MonoNeRF: Learning Generalizable NeRFs from Monocular Videos without Camera Poses. *Proc. International Conference on Machine Learning*, 10392–10404. <https://proceedings.mlr.press/v202/fu23b.html>
- Fu, Y., Wang, X., Liu, S., Kulkarni, A., Kautz, J., & Efros, A. A. (2024). COLMAP-Free 3D Gaussian Splatting. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 20796–20805. <https://doi.org/10.1109/CVPR52733.2024.01965>
- Girish, S., Gupta, K., & Shrivastava, A. (2024). EAGLES: Efficient Accelerated 3D Gaussians with Lightweight EncodingS. *Proc. European Conference on Computer Vision*, 54-71.
- Grupp, M. (2017). evo: Python package for the evaluation of odometry and SLAM. <https://github.com/MichaelGrupp/evo>
- Guédon, A., & Lepetit, V. (2024). SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 5354–5363. <https://doi.org/10.1109/CVPR52733.2024.00512>
- Ham, Y., Michalkiewicz, M., & Balakrishnan, G. (2024). DRAGON: Drone and Ground Gaussian Splatting for 3D Building Reconstruction. *Proc. International Conference on Computational Photography*, 1–12. <https://doi.org/10.1109/ICCP61108.2024.10644903>
- Huang, B., Yu, Z., Chen, A., Geiger, A., & Gao, S. (2024). 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. *ACM SIGGRAPH Conference papers*, 1–11. <https://doi.org/10.1145/3641519.3657428>
- Kerbl, B., Kopanas, G., Leimkuehler, T., & Drettakis, G. (2023). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4), 139:1-139:14. <https://doi.org/10.1145/3592433>
- Lee, J. C., Rho, D., Sun, X., Ko, J. H., & Park, E. (2024). Compact 3D Gaussian Representation for Radiance Field. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 21719–21728. <https://doi.org/10.1109/CVPR52733.2024.02052>
- Lin, C.-H., Ma, W.-C., Torralba, A., & Lucey, S. (2021). BARF: Bundle-Adjusting Neural Radiance Fields. *Proc. IEEE International Conference on Computer Vision*, 5721–5731. <https://doi.org/10.1109/ICCV48922.2021.00569>
- Lin, H., Chen, S., Liew, J., Chen, D. Y., Li, Z., Shi, G., Feng, J., & Kang, B. (2025). Depth Anything 3: Recovering the Visual Space from Any Views (No. arXiv:2511.10647). arXiv. <https://doi.org/10.48550/arXiv.2511.10647>
- Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., & Yang, W. (2024). VastGaussian: Vast 3D Gaussians for Large Scene Reconstruction. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 5166–5175. <https://doi.org/10.1109/CVPR52733.2024.00494>
- Liu, Y., Guan, H., Luo, C., Fan, L., Wang, N., Peng, J., & Zhang, Z. (2024). CityGaussian: Real-time High-quality Large-Scale Scene Rendering with Gaussians. *Proc. European Conference on Computer Vision*, 265-282. <https://doi.org/10.48550/arXiv.2404.01133>
- Lyu, X., Sun, Y.-T., Huang, Y.-H., Wu, X., Yang, Z., Chen, Y., Pang, J., & Qi, X. (2024). 3DGSr: Implicit Surface Reconstruction with 3D Gaussian Splatting. *ACM Trans. Graph.*, 43(6), 198:1-198:12. <https://doi.org/10.1145/3687952>
- Mallick, S. S., Goel, R., Kerbl, B., Steinberger, M., Carrasco, F. V., & De La Torre, F. (2024). Taming 3DGS: High-Quality Radiance Fields with Limited Resources. *SIGGRAPH Asia Conference Papers*, 1–11. <https://doi.org/10.1145/3680528.3687694>
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). NeRF: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1), 99–106. <https://doi.org/10.1145/3503250>
- Mur-Artal, R., Montiel, J. M. M., & Tardós, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>
- Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262. <https://doi.org/10.1109/TRO.2017.2705103>
- Papantonakis, P., Kopanas, G., Kerbl, B., Lanvin, A., & Drettakis, G. (2024). Reducing the Memory Footprint of 3D Gaussian Splatting. *Proc. ACM Comput. Graph. Interact. Tech.*, 7(1), 16:1-16:17. <https://doi.org/10.1145/3651282>

- Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision Transformers for Dense Prediction. *IEEE International Conference on Computer Vision*, 12159–12168. <https://doi.org/10.1109/ICCV48922.2021.01196>
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Proc. International Conference on Computer Vision*, 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- Schmidt, C., Pickenbrinck, J., & Leibe, B. (2024). Look Gauss, No Pose: Novel View Synthesis using Gaussian Splatting without Accurate Pose Initialization. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 8732–8739. <https://doi.org/10.1109/IROS58592.2024.10801639>
- Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-Motion Revisited. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 4104–4113. <https://doi.org/10.1109/CVPR.2016.445>
- Schönberger, J. L., Zheng, E., Frahm, J.-M., & Pollefeys, M. (2016). Pixelwise View Selection for Unstructured Multi-View Stereo. *Proc. European Conference on Computer Vision*, 501–518. [https://doi.org/10.1007/978-3-319-46487-9\\_31](https://doi.org/10.1007/978-3-319-46487-9_31)
- Shi, D., Cao, S., Fan, L., Wu, B., Guo, J., Chen, R., Liu, L., & Ye, J. (2025). TrackGS: Optimizing COLMAP-Free 3D Gaussian Splatting with Global Track Constraints (No. arXiv:2502.19800). arXiv. <https://doi.org/10.48550/arXiv.2502.19800>
- Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph.*, 25(3), 835–846. <https://doi.org/10.1145/1141911.1141964>
- Wang, J., Chen, M., Karaev, N., Vedaldi, A., Ruppel, C., & Novotny, D. (2025a). VGGT: Visual Geometry Grounded Transformer. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 5294–5306. arXiv. <https://doi.org/10.48550/arXiv.2503.11651>
- Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., & Yang, J. (2025b). MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 5261–5271. [https://openaccess.thecvf.com/content/CVPR2025/html/Wang\\_MoGe\\_Unlocking\\_Accurate\\_Monocular\\_Geometry\\_Estimation\\_for\\_Open-Domain\\_Images\\_with\\_CVPR\\_2025\\_paper.html](https://openaccess.thecvf.com/content/CVPR2025/html/Wang_MoGe_Unlocking_Accurate_Monocular_Geometry_Estimation_for_Open-Domain_Images_with_CVPR_2025_paper.html)
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. <https://doi.org/10.1109/TIP.2003.819861>
- Wang, Z., Wu, S., Xie, W., Chen, M., & Prisacariu, V. A. (2022). NeRF--: Neural Radiance Fields Without Known Camera Parameters (No. arXiv:2102.07064). arXiv. <https://doi.org/10.48550/arXiv.2102.07064>
- Yen-Chen, L., Florence, P., Barron, J. T., Rodriguez, A., Isola, P., & Lin, T.-Y. (2021). iNeRF: Inverting Neural Radiance Fields for Pose Estimation. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1323–1330. <https://doi.org/10.1109/IROS51168.2021.96>
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 586–595. <https://doi.org/10.1109/CVPR.2018.00068>