

GauU-Scene: A Scene Reconstruction Benchmark on Large Scale 3D Reconstruction Dataset Using Gaussian Splatting

Butian Xiong

The Chinese University of Hong Kong, Shenzhen
School of Science and Engineering

butianxiong@link.cuhk.edu.cn

Abstract

We introduce a novel large-scale scene reconstruction benchmark using the newly developed 3D representation approach, Gaussian Splatting, on our expansive U-Scene dataset. U-Scene encompasses over one and a half square kilometres, featuring a comprehensive RGB dataset coupled with LiDAR ground truth. For data acquisition, we employed the Matrix 300 drone equipped with the high-accuracy Zenmuse L1 LiDAR, enabling precise rooftop data collection. This dataset, developed under the auspices of the Chinese University of Hong Kong and Shenzhen MSU-BIT University, offers a unique blend of urban and academic environments for advanced spatial analysis. Our evaluation of U-Scene with Gaussian Splatting includes a detailed analysis across various novel viewpoints. We also juxtapose these results with those derived from our accurate point cloud dataset, highlighting significant differences that underscore the importance and innovation of our work.

1. Introduction

3D reconstruction is a transformative technology, enabling the conversion of real-world scenes into digital three-dimensional models. This technology, often involving the transformation of multiple 2D images into 3D models, fosters applications in urban planning, virtual reality (VR), and augmented reality (AR). Various techniques have been employed to enhance the accuracy and efficiency of 3D reconstruction. Among them, Structure from Motion (SfM) has been notably prominent, as extensively studied by [10]. This photogrammetric technique uses 2D images captured from different angles to reconstruct three-dimensional structures.

A recent innovation in 3D reconstruction is the advent of Neural Radiance Fields (NeRF), which predicts volumetric scene representations from sparse 2D images using a fully connected neural network [8]. Despite ini-

tial challenges such as training difficulty and limitations in scale and complexity, rapid advancements have been made. Meta-learning techniques [3], sparsity exploitation [18], data structure integration [5, 7], and eigenvalue usage [2] have significantly improved NeRF’s performance. The current state-of-the-art method in addressing aliasing issues has also been proposed [1].

NeRF models have been successful in synthesizing novel views of scenes, but the Gaussian Splatting Representation [6] has emerged as a more recent 3D representation, combining rasterization with novel view synthesis for large-scale scenarios. While effective from a distance, Gaussian Splatting exhibits limitations such as blurring at closer inspection, which we analyze in detail in this study. We also explore the discrepancies between ground truth point cloud data and novel views generated by Gaussian Splatting.

Neural fields have rapidly developed beyond NeRF, with significant advancements in alternatives such as [4, 12, 16]. However, these are not the focus of our current work and will not be discussed further.

Adapting NeRF models to large-scale environments, such as cityscapes, has been an emerging research area [11, 13]. Compared to previous indoor scene reconstructions, these adaptations target extensive urban areas, challenging the conventional scopes of smaller scenes.

The challenge of city-scale reconstruction extends beyond the model to the dataset. Existing city-scale datasets [11, 13, 15] have limitations, either lacking accurate ground truth data or being focused on scene understanding rather than reconstruction. To address this, we employ the DJI Matrix 300 drone with Zenmuse L1 Lidar for capturing highly accurate 3D RGB point clouds. This approach enables us to venture beyond traditional indoor scenes, undertaking large city-scale outdoor scene reconstructions. Our dataset covers an area larger than 2.4 km^2 , surpassing the scale of existing datasets like UrbanBIS [15] and comparable to CityNeRF’s RGB dataset [13].

Our contributions are threefold:



Figure 1. Our dataset is divided into three main parts. The first part is the top part of this graph. We call it CUHKSZ(The Chinese University of Hong Kong, Shenzhen) lower campus, and the bottom left corner shows the upper campus of CUHKSZ, and the bottom right corner shows the SMBU(Shenzhen MSU-BIT University) Campus. We use highly accurate lidar to collect the dataset and the range we cover is more than 1.5 km^2 .

- Provision of a large dataset covering over 1.5 km^2 with a dense point cloud using Lidar as illustrated in the 1.
- Establishment of a concrete benchmark on our dataset using state-of-the-art Gaussian Splatting.
- Identification of Gaussian Splatting’s drawbacks and the gap between matrices generated from novel RGB views and ground truth point clouds.

In subsequent sections, we discuss related work in 3D reconstruction, Neural Field, and Volumetric Rendering. After that we will introduce our measurement matrix, and detail our data collection methodology and some preliminaries of Gaussian Splatting. Finally, we discuss future work and conclusions.

2. Related Work

In this section, we will introduce related work about the neural field. [14] has already provided an all-around overview of the neural field and beyond, in this text, I will just summarize the important parts of them and give it a summary.

As what you can see from the 2 it provides the following steps to reconstruct a 3D model, first, we use a multi-layer Perceptron to approximate an underlying data distribution. This data distribution can be anything, in the current work, we introduce two different situations, radiance field or signed distance field

- Radiance Field: For radiance field, the input is a coordinate in R^3 and an angle in R^2 . And the output will be a property along that ray. After that, we will try to do a rendering of the neural radiance field to get the rendered 2D image. We usually call it as the volumetric rendering. In the introduction section, we will introduce this part in a more detailed way.
- Sign Distance Field: The basic idea is that we thing the object is converted by a surface. When the input point in R^3 is in one side of the surface, the sign will be negative. And when the input point is on the surface, the sign will be zero. And when the point is on the other side of the surface, the sign will be positive. And the magnitude of the output value should represent the distance to the surface. The way we render it is by using the Sphere Tracing.

Now we introduce NeRF and Volumetric in detail.

2.1. Neural Radiance Field

Neural Radiance Fields (NeRF) represent a sophisticated approach to 3D scene reconstruction that stands on the shoulders of deep learning. The essence of NeRF is to synthesize novel views of complex scenes with unprecedented detail and realism through the use of a neural network to model the volumetric scene implicitly. At its core, NeRF employs a fully connected deep neural network to map 5D coordinates (Spatial location $X = (x, y, z)$, and view direction which is a 2D angle $d = (\theta, \phi)$) to a volumetric scene

function. This function outputs both the volume density and the color (as RGB values) for any point in space. The fundamental premise is that for any given ray emanating from the camera position, one can predict the color at the camera sensor by accumulating the contributions of all points along that ray. Here’s a step-by-step breakdown of the process:

- **Sampling:** NeRF begins by casting rays through each pixel of the provided 2D images and samples points along these rays in the scene space.
- **Prediction:** For each sampled point, the NeRF model takes its 3D position and 2D viewing direction as input and predicts two quantities through a neural network: the RGB color and the differential volume density.
- **Volume Rendering:** The predicted colors and densities along each ray are composed using the classical volume rendering technique known as ray marching, specifically via numerical quadrature. This involves approximating the integral of radiance along the ray to compute the final pixel colour. It relies on the rendering equation, which models the light transport to create images.
- **Optimization:** NeRF optimizes the neural network parameters by minimizing the difference between the rendered colors $C(r)$ from the rays and the corresponding observed pixel colours from the input images.

2.2. Volumetric Rendering

Here we first introduce volume rendering equations in a continuous form like this:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(t)c(t)dt$$

The colour $C(r)$ of a ray r that we actually observe is computed via the volume rendering integral. Where t is the bounds of the integral along the ray and $T(t)$ is the accumulated transmittance defined as:

$$T(t) = \exp(-\int \sigma(s)ds)$$

In practice, the continuous integral is approximated using numerical quadrature. This involves sampling N points $\{t_i\}_{i=1}^N$ along the ray and computing a Riemann sum:

$$C(r) \approx \sum_{i=1}^N T_i(1 - \exp(-\sigma(t_i)\Delta t_i))c_{t_i}$$

Where Δt_i is the distance between adjacent sample points, and T_i is the discrete transmittance computed as:

$$T_i = \exp(-\sum_{j=1}^{i-1} \sigma(t_j)\Delta t_j)$$

The term $1 - \exp(-\sigma(t_i)\Delta t_i)$ is the probability that light is scattered at t_i .

3. Preliminaries

In this section, we introduce the testing method Gaussian Splatting we used in the current work.

Gaussian Distribution

A Gaussian distribution, often referred to as a normal distribution, is a continuous probability distribution characterized by a bell-shaped curve. It is completely determined by its mean μ and variance σ^2 (or standard deviation σ). The probability density function (PDF) of a Gaussian distribution is given by:

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right). \quad (1)$$

In the multivariate case, the Gaussian distribution can be extended to accommodate a vector of means and a covariance matrix. The multivariate Gaussian PDF with mean vector θ and covariance matrix Σ is:

$$p(\mathbf{y}|\theta, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \theta)^T \Sigma^{-1}(\mathbf{y} - \theta)\right), \quad (2)$$

where $|\Sigma|$ denotes the determinant of Σ , and k is the dimensionality of the vector \mathbf{y} .

Covariance Matrix

The covariance matrix Σ is a symmetric positive semi-definite matrix that contains the covariances between pairs of elements of the random vector. The covariance between two random variables X and Y is calculated as:

$$\sigma_{XY} = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}), \quad (3)$$

where \bar{X} and \bar{Y} are the sample means of X and Y , respectively.

Symmetric Property and Optimization of the Covariance Matrix

The eigenvalues and eigenvectors of the covariance matrix play a crucial role in understanding the shape and orientation of the distribution. The eigenvector corresponding to the largest eigenvalue points in the direction of the greatest variance (the major axis of the ellipse in the case of a bivariate Gaussian), with the eigenvalue indicating the magnitude of this variance. To maintain the positive semi-definite property of the covariance matrix during optimization, it is essential to parameterize the matrix in such a way that this property is not violated.

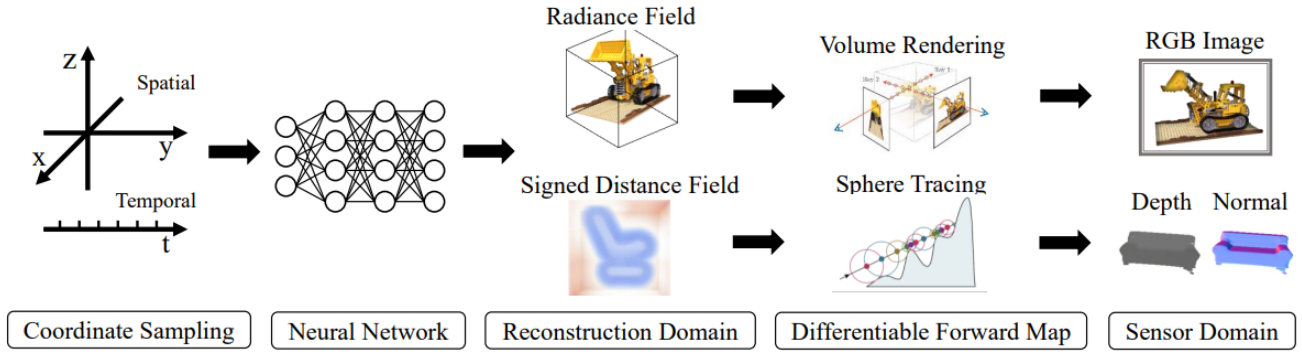


Figure 2. This is the caption of the figure.: A typical feed-forward neural field algorithm. Spatiotemporal coordinates are fed into a neural network which predicts values in the reconstruct a domain. Then, this domain is mapped to the sensor domain where sensor measurements are available as supervision. Figures adapted from [8] [9]

Proofs and Properties

Additional proofs and properties of the Gaussian distribution and its covariance matrix are beyond the scope of this preliminary section, we provide the proof it in the appendix 6

3.1. 3D Representation

In the context of 3D graphics, Gaussian Splatting is a technique used for rendering purposes, which involves representing points in space with Gaussian functions. Each point is represented as a Gaussian splat with several attributes that define its appearance in the rendered image. These attributes include:

- \mathbf{m} : the mean vector, which defines the center of the splat in the 3D space.
- σ : the covariance matrix, which determines the spread of the splat in different directions.
- α : the opacity, which defines the transparency level of the splat.
- \mathbf{h} : the color, represented using Harmonic Spheres to encode the color variation within the splat.

A Gaussian splat can thus be defined by the tuple $\langle \mathbf{m}, \sigma, \alpha, \mathbf{h} \rangle$. The equation for a Gaussian splat incorporating these attributes is given by:

$$G(\mathbf{m}, \sigma, \alpha, \mathbf{h}) \quad (4)$$

when we do the rendering, we will integrate through the Gaussian Splatting in the 3D space using alpha blendering

The color function $\mathbf{h}(\mathbf{x})$ is defined on the Harmonic Spheres, which allow for a smooth transition of colors across the surface of the splat, adding a rich visual texture to the rendered scene.

4. Methods

In this section, we will mainly introduce how we collect the data, and how we transfer the data into 3D point cloud. What is the scale of the dataset and how we design the measurement matrix.

4.1. Data Collection

The data collection technique mainly contains the following three stages.

- Drone assembly and hardware preparation. The detailed information is hard to describe in purely text, therefore, we provide a link to help our readers to understand the [procedure](#).
- Path Planning: Since we need to obtain all views of each building instead of just the rooftop, we need to set the routing algorithm in DJI-Pilot to the oblique shooting and select Zenmuse L1 and the camera, the detailed information can be examined in the oblique shooting [guidance](#).
- Flying: We need to use our controller to initiate fly according to the routing path we set before. There are something one needs to be very careful about, otherwise, there might be a crashing drone. For example, we need to keep keep distance, and altitude, we need to measure the wind speed and so on, the detailed information can be examined in the [tips](#).

The equipment we use is the DJI Matrix 300 Flying platform together with Zenmuse L1 LiDAR. The detailed parameters are shown in the table 3 and 4 in the appendix

4.2. Data Post-Processing

After we collected the data, we then need to post-process the data utilizing DJI-Terra and we need to merge the data

together to form a new LAS or PLY point cloud. The detailed procedure are shown in this [manual](#)

4.3. Measurement Matrix

In the field of computer graphics reconstruction, the quality of reconstructed images is often assessed using a variety of metrics. The Measurement Matrix includes three key metrics: SSIM, PSNR, and LPIPS.

- **SSIM (Structural Similarity Index):** The Structural Similarity Index (SSIM) is a metric used to measure the similarity between two images. By calculating the probability attributes of two image and compare the difference we have the following equation:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

- μ_x and μ_y are the average intensities of images x and y ,
- σ_x^2 and σ_y^2 are the variances of x and y ,
- σ_{xy} is the covariance of x and y .

- **PSNR (Peak Signal-to-Noise Ratio):** PSNR is a widely used metric in image processing for measuring the quality of reconstructed or compressed images compared to the original images. It is calculated using the logarithmic scale of the ratio between the maximum possible power of a signal (in this case, the original image) and the power of corrupting noise (differences between the original and the reconstructed image). Higher PSNR values indicate better image quality.
- **LPIPS (Learned Perceptual Image Patch Similarity):** LPIPS [17] is a more recent metric that leverages deep learning to assess image quality. Unlike traditional metrics that focus on pixel-level differences, LPIPS evaluates perceptual similarity between images. It does this by using neural networks trained to mimic human visual perception, thus offering a more accurate assessment of the perceived image quality, especially in cases where traditional metrics may fail to align with human judgment.

The newly developed measurement matrix for point cloud measurement of Gaussian Splatting is also a PSNR. Since point cloud do not have any opacity value and therefore cannot have a one-to-one matching to Gaussian Splatting, we only use the color attributes provided by the Gaussian Splatting Model at a particular position that compares with the real value given by the LiDAR point cloud. The difference between the model prediction and the true value is regarded as noise and the peak signal is the maximum RGB vector which is $< 255, 255, 255 >$.



Figure 3. The figure shown here is one small portion in Gaussian Splatting Reconstructed Scene From Top View

The 3D reconstruction method used here is Gaussian Splatting [6]. And the result are shown in this [graph 3](#)

5. Result

The qualitative results can be examined on the [project page](#), and the content will be updated shortly with additional visualization details. A small image of the Gaussian is available in [Figure 3](#). The quantitative results are presented in [Table 1](#).

The results show consistency across the datasets we collected, as the matrix values are similar. For a fair comparison, we used a single RTX 3090 to train both Mip-NeRF 360 and Gaussian Splatting. Although Gaussian Splatting does not outperform Mip-NeRF 360, the average training time for Gaussian Splatting is significantly faster—approximately 30 minutes, compared to around three days for Mip-NeRF 360.

The dataset we provided here contains more important information such as the altitude and quality analysis result shown here in the [figure 4](#)

The point cloud data is in LAS format and PLY format, and we sampled the final output to be 15cm per dot. The sample output looks like what shown in the [4](#).

In our data set, we also provided a point cloud data set. This point cloud data set can not only provide additional prior to Gaussian Splatting like what DrivingGaussing [19] does, it can also use an additional ground truth that can help us identify how well our Gaussian Splatting is doing. The result is shown in the [table 2](#). As one can see that the LiDAR data shows a better result on the PSNR, which shows the Gaussian is more sensitive to 3D representation instead of the final rendering result of 2D images.

6. Conclusion

In this work, we present three significant contributions to the field of computer vision and dataset processing. First,

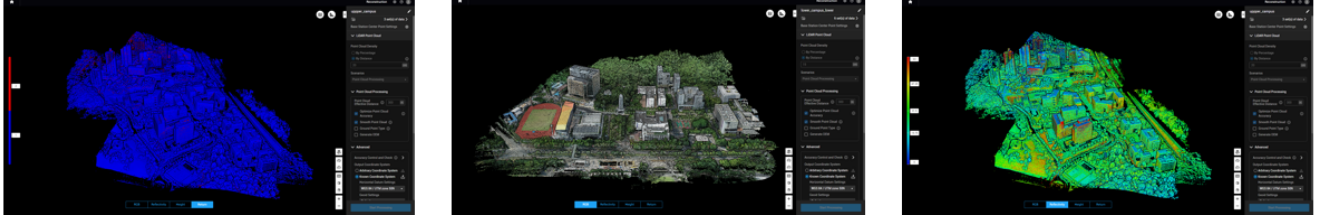


Figure 4. The left one is the quality of the point if the point is blue, then the quality is ok, otherwise it is red. The right-hand side picture shows the point’s altitude, and the middle one is the RGB point cloud

Table 1. U-Scene Test Result on Mip-NeRF 360 and Gaussian Splatting

Dataset	CUHKSZ LOWER			CUHKSZ UPPER			SMBU		
Metric	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
M-NeRF360	0.792	27.69	0.237	0.786	27.22	0.254	0.763	27.40	0.255
3D-GS	0.887	23.43	0.234	0.884	23.14	0.235	0.879	22.22	0.257

Table 2. U-Scene Test Result on Gaussian Splatting Using LiDAR Data

Dataset	PSNR LiDAR	PSNR_IMG
CUHKSZ_LOWER	34.86	23.43
CUHKSZ_UPPER	33.73	23.14
SMBU	34.78	22.22

we provide a comprehensive dataset that spans more than 1.5 square kilometers, meticulously detailed with a dense Lidar-generated point cloud, as depicted in Figure 1. Second, we establish a robust benchmark using state-of-the-art Gaussian Splatting techniques applied to our extensive dataset. Finally, we critically analyze the limitations of Gaussian Splatting by comparing the matrices derived from novel RGB views against the ground truth data provided by point clouds. This analysis not only highlights the current shortcomings but also pinpoints the disparities that future research can aim to resolve. There are several things that we will finish shortly. The first is that for the large scene reconstruction, we need to eliminate the edge effect. That is the edge of different Gaussian is blur and hard to merge. The second would be testing the result of combining the LiDAR prior as the initial of Gaussian Splatting and test the result.

References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 1
- [2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 1
- [3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 1
- [4] William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM Transactions on Graphics (SIGGRAPH)*, 2023. 1
- [5] Jonathan Granskog, Till N Schnabel, Fabrice Rousselle, and Jan Novák. Neural scene graph rendering. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 1
- [6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 1, 5
- [7] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph. (SIGGRAPH)*, 40(4), 2021. 1
- [8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 4
- [9] Songyou Peng, Boxin Shi, Marc Pollefeys, Zhaopeng Cui, Shaohui Liu, Yinda Zhang. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4
- [10] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 1
- [11] Matthew Tancik, Vincent Casser, Xinchun Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 1
- [12] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R Oswald. Routedfusion: Learning real-time depth map fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4887–4897, 2020. 1
- [13] Yuanbo Xiangli, Lining 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
- [14] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. 2
- [15] Guoqing Yang, Fuyou Xue, Qi Zhang, Ke Xie, Chi-Wing Fu, and Hui Huang. Urbanbis: a large-scale benchmark for fine-grained urban building instance segmentation. In *SIGGRAPH*, pages 1–11, 2023. 1
- [16] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4), jul 2023. 1
- [17] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5
- [18] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5449–5458, June 2022. 1
- [19] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint arXiv:2312.07920*, 2023. 5

Appendix

Proof of Eigenvalue Properties for Covariance Matrix

In the context of the covariance matrix Σ , we consider the eigendecomposition where $\Sigma = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. Here, \mathbf{Q} is the matrix of eigenvectors and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues.

Consider a pair of eigenvalues and eigenvectors $(\mathbf{x}_i, \lambda_i)$ of Σ with the property that $\lambda_i \geq \lambda_j$ for $i < j$. We aim to show that the largest eigenvalue corresponds to the direction of the greatest variance in the data, and subsequent eigenvalues correspond to orthogonal directions of decreasing variance.

Proof for the Longest Axis

Given a unit vector \mathbf{y} , we want to show that it lies within the ellipse defined by the covariance matrix if $\mathbf{y}^T \Sigma \mathbf{y} \leq \lambda_1$, where λ_1 is the largest eigenvalue. To show this, we consider that \mathbf{y} can be expressed as a linear combination of the eigenvectors of Σ :

$$\mathbf{y} = n_1 \mathbf{x}_1 + n_2 \mathbf{x}_2 + \dots + n_k \mathbf{x}_k \quad \text{with} \quad \sum n_i^2 = 1. \quad (5)$$

Then we have:

$$\mathbf{y}^T \Sigma \mathbf{y} = \sum n_i^2 \lambda_i \leq \lambda_1. \quad (6)$$

Proof for Perpendicular Axes

For the second largest eigenvalue λ_2 , we consider the case where the vector corresponding to λ_2 is orthogonal to that of λ_1 . We use the fact that $\mathbf{x}_1^T \Sigma \mathbf{x}_2 = 0$ due to the orthogonality of the eigenvectors. Extending this to the entire set of eigenvectors, we can show that:

$$\mathbf{x}_i^T \Sigma \mathbf{x}_j = 0 \quad \text{for} \quad i \neq j. \quad (7)$$

Hence, we conclude that the eigenvalues of the covariance matrix define the length of the axes of the ellipse (in the case of bivariate Gaussian) or ellipsoid (in the multivariate case), and the eigenvectors define their orientation.

Zenmuse L1 Specifications	
General	
Product Name	Zenmuse L1
Dimensions	152×110×169 mm
Weight	930±10 g
Power	Typical: 30 W; Max: 60 W
IP Rating	IP54
Supported Aircraft	Matrice 300 RTK
Operating Temperature Range	-20° to 50° C (-4° to 122° F) 0° to 50° C (32° to 122° F) (RGB camera)
Storage Temperature Range	-20° to 60° C (-4° to 140° F)
System Performance	
Detection Range 190 m @ 10Point Rate	450 m @ 80 Single return: max. 240,000 pts/s; Multiple return: max. 480,000 pts/s
System Accuracy (RMS 1)	Horizontal: 10 cm @ 50 m; Vertical: 5 cm @ 50 m
Real-time Point Cloud Coloring Modes	Reflectivity, Height, Distance, RGB
Lidar	
Ranging Accuracy (RMS 1)	3 cm @ 100 m
Maximum Returns Supported	3
Scan Modes	Non-repetitive, Repetitive patterns
FOV	Non-repetitive: 70.4° × 77.2°; Repetitive: 70.4° × 4.5°
Laser Safety	Class 1 (IEC 60825-1:2014)
Inertial Navigation System	
IMU Update Frequency	200 Hz
Accelerometer Range	±8 g
Angular Velocity Meter Range	±2000 dps
Yaw Accuracy (RMS 1)	Real-time: 0.3°, Post-processing: 0.15°
Pitch / Roll Accuracy (RMS 1)	Real-time: 0.05°, Post-processing: 0.025°
Auxiliary Positioning Vision Sensor	
Resolution	1280×960
FOV	95°
RGB Mapping Camera	
Sensor Size	1 inch
Effective Pixels	20 MP
Photo Size	Various Formats
Focal Length	8.8 mm / 24 mm (Equivalent)
Shutter Speed	Mechanical: 1/2000 - 8 s; Electronic: 1/8000 - 8 s
ISO	Video: 100 – 3200 (Auto), 100 – 6400 (Manual); Photo: 100 - 3200 (Auto), 100 - 12800 (Manual)
Aperture Range	f/2.8 - f/11
Supported File System	FAT (32 GB) exFAT (≤32 GB)
Photo Format	JPEG
Video Format	MOV, MP4
Video Resolution	H.264, 4K: 3840×2160 30p
Gimbal	
Stabilized System	3-axis (tilt, roll, pan)
Angular Vibration Range	0.01°
Mount	Detachable DJI SKYPORT
Mechanical Range	Tilt: -120° to +30°; Pan: ±320°

Table 3. Zenmuse L1 Specifications

DJI Matrice 300 Specifications	
Aircraft	
Dimensions (Unfolded)	810×670×430 mm (L×W×H)
Dimensions (Folded)	430×420×430 mm (L×W×H)
Diagonal Wheelbase	895 mm
Weight (without batteries)	Approx. 3.6 kg
Weight (with batteries)	Approx. 6.3 kg
Max Payload	930 g
Max Takeoff Weight	9 kg
Operating Frequency	2.4000-2.4835 GHz; 5.725-5.850 GHz
EIRP	Varies with frequency and region
Hovering Accuracy	Varies with mode and condition
RTK Positioning Accuracy	1 cm+1 ppm (Horizontal); 1.5 cm+1 ppm (Vertical)
Max Angular Velocity	Pitch: 300°/s, Yaw: 100°/s
Max Pitch Angle	30° (25° with FVS)
Max Ascent/Descent Speed	6 m/s (Ascent), 5 m/s (Descent)
Max Speed	23 m/s (S mode); 17 m/s (P mode)
Service Ceiling	5000 m / 7000 m (depending on propellers)
Max Wind Resistance	12 m/s
Max Flight Time	55 min
Supported Gimbals	Zenmuse XT2/XT S/Z30/H20/H20T/DJI P1/DJI L1
Gimbal Configurations	Various configurations
Ingress Protection	IP45
GNSS	GPS+GLONASS+BeiDou+Galileo
Operating Temperature	-20°C to 50°C
Remote Controller	
Operating Frequency	2.4000-2.4835 GHz; 5.725-5.850 GHz
Max Transmitting Distance	Varies with region
EIRP	Varies with frequency and region
External Battery	
Type	WB37 Intelligent Battery
Capacity	4920 mAh
Voltage	7.6V
Energy	37.39Wh
Charging Time	70-130 minutes (temperature dependent)
Built-in Battery	
Type	18650 lithium ion battery
Capacity	5000 mAh
Voltage	7.2 V
Charging Power	17 W
Battery Life	Approx. 2.5h (built-in); 4.5h (with external)
USB Power Supply	5 V / 1.5 A
Operating Temperature	-20°C to 40°C
Vision System	
Obstacle Sensing Range	Varies with direction
FOV	Varies with direction
Operating Environment	Surfaces with clear patterns and adequate lighting
Infrared ToF Sensing System	
Obstacle Sensing Range	0.1-8m
FOV	30° (±15°)
Operating Environment	Large, diffuse, and reflective obstacles
FPV Camera	
Resolution	960p
FOV	145°
Frame rate	30 fps

Table 4. Technical Details of the DJI Matrice 300