# An Input Data-Oriented Optimization Strategy for Using VGGT on Low-Compute Edge Devices

Abhijat Bharadwaj, Parth Bansal, Animesh Kumar, Prof. Amit Sethi
*Department of Electrical Engineering*
*Indian Institute of Technology Bombay*

***Abstract*—Large transformer-based 3D reconstruction models offer strong accuracy but remain difficult to deploy on low-compute devices due to heavy memory and processing demands. This work explores the 3D reconstruction and spatial understanding capabilities of Visual Geometry Grounded Transformer (VGGT) and demonstrate that the VGGT can maintain reliable scene reconstruction even when provided with far fewer input images than originally required or captured. Building on this observation, we explore an input data–oriented approach that selects only the most informative views while discarding redundant frames and evaluate the approach with four subset selection strategies. Experiments on unseen ETH3D scenes demonstrate that substantial reductions in input size still preserve camera pose accuracy and point cloud quality, enabling practical and efficient use of the model in resource-limited environments.**

***Index Terms*—3D-reconstruction, DINO, Point cloud, Subset selection, Transformers, VGGT**

## I. Introduction

Recent advances in deep learning techniques have greatly benefit the research in 3D reconstruction by shifting from traditionally complex and computationally expensive geometric methods to data-driven learning models. The Visual Geometry Grounded Transformer (VGGT) [1] is a powerful 3D-view reconstruction model that demonstrates very promising results in scene reconstruction, one-shot generation, and novel view synthesis. VGGT directly predicts camera parameters, dense depth maps, 3D point clouds, and point tracking while eliminating the need for intermediate post-processing and heavy optimization steps characteristic of classical pipelines [1]. This streamlined approach enables highly accurate and efficient reconstruction in under a second per scene, a performance level that surpasses prior task-specific models and foundational methods like DUSt3R [2] and MASt3R [3].

Despite its predictive power and multi-task versatility, VGGT's large model size, approximately 1.2 billion parameters [1], places high demands on memory and computation. This limits its deployment to high-capacity hardware, making it unsuitable for many edge or low-power devices commonly used in practical scenarios such as mobile robotics, augmented reality, and embedded vision systems. Moreover, as edge devices may capture unrefined, low quality, redundant or crowd-sourced data, the computation resources turn into a bigger hindrance. Consequently, effective optimization and compression strategies become essential to enable VGGT's broad adoption in resource-constrained environments [4], [5].

**Our Observations.** Although VGGT achieves its best performance with extensive input data, we observe that it does not require redundant information for reliable generation. Aided by strong zero-shot single-view generation and novel-view synthesis capabilities, VGGT exhibits a well-structured understanding of spatial consistency, enabling coherent scene reconstruction and maintaining geometric plausibility across generated views. Figure 1 demonstrates that 3D views of object in a scene generation can be generated to a comparable degree of accuracy by only using 16.7% of the original data. The subset was created by uniformly skipping every $6^{th}$ frame. Figure 2 depicts the 3D view of the scene generated by using 30 frames and a subset of 5 frames. Corresponding parts of Figure 3 shows the ground-truth and predicted camera trajectory, projected on the XY plane. We can clearly see the consistency in camera trajectory predictions and 3D scene reconstruction.

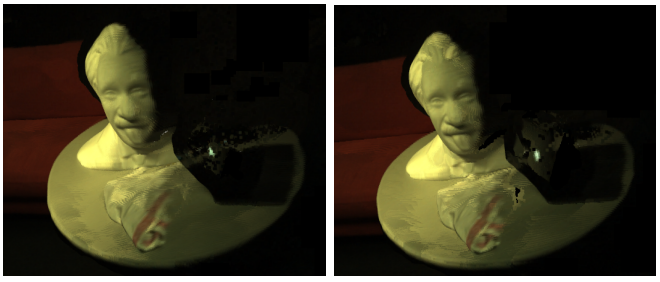

Fig. 1. 3D point-cloud reconstruction of scene objects using 30 input frames (left) vs. a subset of 5 input frames (right). Regions beyond the reconstructed points are rendered in black for visual clarity.

**Our Contributions.** This work focuses on input data-oriented optimization strategies specifically tailored to VGGT. By carefully selecting only the minimal required information, as per the constraint, through techniques that consider the unique characteristics of the input and architecture of VGGT, the goal is to achieve efficient inference without sacrificing the quality of the reconstruction. We formalize our problem statement as a that of a subset selection, and critically review four strategies to achieve the minimal viable input subset.

## II. Background and Related Works

### A. The Visual Geometry Grounded Transformer (VGGT)

VGGT [1] is a large feed-forward architecture tailored for multi-view 3D reconstruction. Given a sequence of RGB

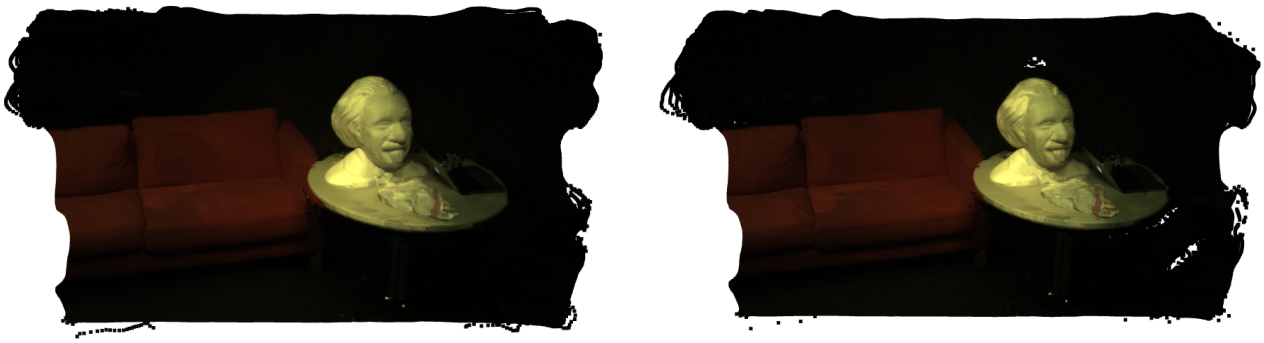

Fig. 2. 3D point-cloud reconstruction of a scene (same as Figure 1) using 30 input frames (left) vs. a subset of 5 input frames (right).

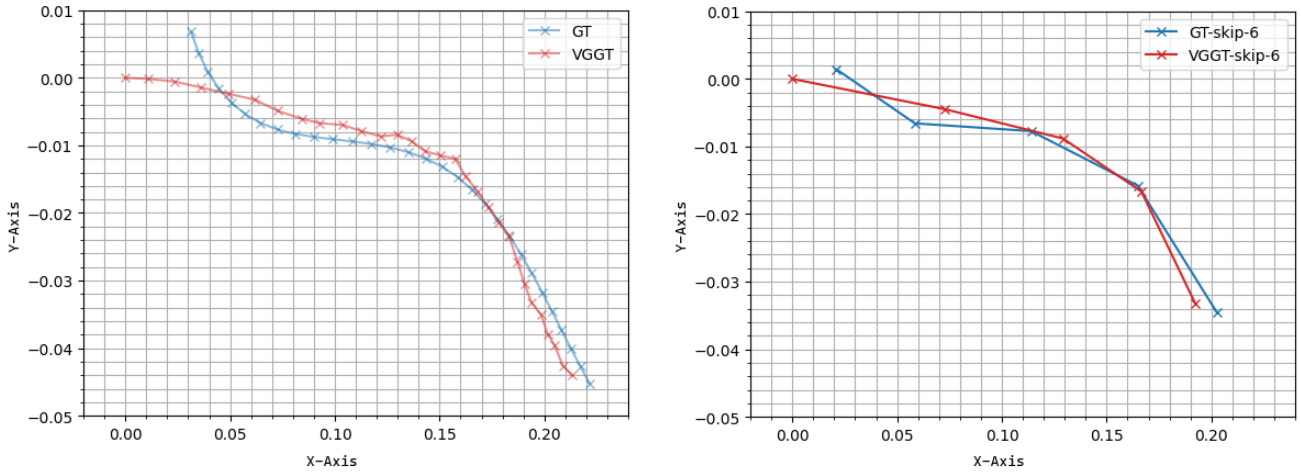


Fig. 3. Ground truth (blue) and predicted (red) camera trajectory for a scene (same as Figure 1) using 30 input frames (left) vs. a subset of 5 input frames (right), projected on XY plane for visualization. Skip-6 subset was created by skipping every $6^{th}$ frame of the original scene.

images $\{I_i\}_{i=1}^N$ of some scene, it directly predicts per-frame camera parameters ($g_i \in \mathbb{R}^k$), depth maps ($D_i \in \mathbb{R}^{H\times W}$), point maps ($P_i \in \mathbb{R}^{3\times H\times W}$), and dense tracking features ($T_i \in \mathbb{R}^{C\times H\times W}$). The model uses a transformer backbone with Alternating-Attention (AA) [1], switching between frame-wise and global self-attention to balance intra-frame normalization with cross-view aggregation. Images are patchified using DINO-v2 [6], augmented with camera and register tokens, and processed by $L$ AA layers. DPT-based [7] heads and convolutional modules produce structured geometric fields and uncertainty estimates from the resulting tokens.

VGGT's output space is intentionally over-complete: although depth, point maps, and camera parameters are geometrically coupled, supervising all tasks jointly improves consistency and accuracy. Cameras are parameterized by quaternion rotation ($q \in \mathbb{R}^4$), translation ($t \in \mathbb{R}^3$), and field of view ($f \in \mathbb{R}^2$), with the first frame fixed as reference and the remainder treated permutation-equivariantly. The tracking features integrate seamlessly with modules such as CoTracker [8], [9], enabling robust point correspondence, tracking, view synthesis, and large-scale scene reconstruction, often surpassing optimized geometric systems in both speed and accuracy.

### *B. Model Quantization for Post-training Optimization*

The combination of accuracy and speed positions VGGT as a strong candidate for real-time 3D perception and a natural foundation for low-compute optimization strategies [4], [10]. Yet its 1.2B parameters pose clear deployment challenges: the memory footprint and compute demands exceed the capabilities of most low-power devices despite outperforming several task-specific models. These limitations motivate systematic compression and optimization, with recent work emphasizing input-level strategies such as token pruning, patch selection, efficient geometric encoding, and quantization.

Model compression, particularly quantization, has shown strong potential in 2D vision transformers and large language models but are still under exploration for large-scale 3D architectures like VGGT [10]. Quantization methods broadly fall into *Quantization-Aware Training* (QAT), which preserves accuracy under aggressive bit reductions via retraining, and *Post-Training Quantization* (PTQ), which relies on lightweight

calibration and demands careful design to mitigate accuracy degradation [11], [12]. Recent PTQ methods such as BRECQ, QDrop, SmoothQuant, and QuaRot improve robustness through block reconstruction, stochastic quantization, and activation-outlier suppression [13]–[16]. Feng *et al.* [10] claim to have achieved 4-bit quantization of VGGT.

## III. Methodology

In this section, we presents four main strategies followed in this study for selecting a compact and representative subset of images from a large dataset using their feature embeddings. The goal is to reduce input redundancy while preserving the underlying distribution of features, enabling efficient VGGT inference on low-compute devices. Let the dataset $X \in \mathbb{R}^{N\times H\times W\times 3}$ contain $N$ of RGB images of size $H \times W$, each mapped to a $D$-dimensional embedding using a pretrained backbone. Since DINO [6] also serves as the feature backbone of VGGT, we utilize DINO as our feature extractor. The embeddings form a matrix

$$\hat{X} \in \mathbb{R}^{N\times D},$$

where the $i$-th row $\mathbf{x}_i$ denotes the embedding of image $i$. For some given constraints, let us assume that we wish to select a subset of length $N_s$. We briefly describe the subset selection and sampling strategies below:

### A. Uniform Sampling

Uniform Sampling constructs a subset by selecting indices evenly spaced across the full index range of the dataset. Given the total number of images $N$ and the desired subset size $N_s$, the sampled indices are

$$u_k = \left\lfloor \frac{k(N-1)}{N_s - 1} \right\rfloor, \qquad k = 0, 1, \ldots, N_s - 1.$$

The resulting uniform subset is

$$S = \{\, u_0,\, u_1,\, \ldots,\, u_{N_s-1} \,\}.$$

If the redundancy is greater than the images skipped between two samples, uniform sampling is computationally inexpensive and provides a statistically fair chance of covering the scene, assuming all images are also drawn from a uniform sample of 2D projections of the scene. However, as the camera may move with a varying speed and its trajectory is often chaotic, the assumption is weakened.

### B. Random Sampling

Another candidate for this study is random sampling. Random Sampling selects $N_s$ distinct images uniformly at random from the dataset indices $\{0, 1, \ldots, N-1\}$. Formally, the selected set is drawn without replacement:

$$S = \mathrm{RandomSubset}(N,\, N_s),$$

meaning that every subset of size $N_s$ has equal probability of being chosen. Assuming no prior information on the images, random sampling is the most statistically fair subset selection strategy.

### C. Utilizing DINO Features

However, since we know that the DINO backbone is used in VGGT, we can utilize the features extracted from DINO to guide our subset selector. Since our objective is to select the $N_s$ best quality samples that cover the maximum amount of non-redundant information, we utilize DINO to decide what qualifies as *information* in an image. As similar images cluster together in the feature space of DINO [6], our sampling strategy resolves to that of a clustering problem.

Specifically, we focus on two algorithms: $K$-Means [17], [18] and Metric $K$-Center [19].

*1) K-Means Clustering:* $K$-Means [17], [18] strategy tends to cluster similar images together in the given feature space and reports the mean representative image. We briefly describe the strategy below:

*a) Clustering Objective:*

$$\min_{\{\boldsymbol{\mu}_j\}_{j=1}^{K}} \sum_{j=1}^{K} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2.$$

*b) Cluster Assignment:*

$$\mathrm{cluster}(i) = \arg \min_{j\in\{1,\ldots,K\}} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2.$$

*c) Representative Selection:*

$$\mathbf{x}_j^* = \arg \min_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2.$$

where, $\forall j \in \{1, ..., K\}$, $C_j$ represents $j^{th}$ cluster with mean $\boldsymbol{\mu}_j$. $\mathbf{x}_i \in C_j$ are images in the cluster $C_j$, and $\mathbf{x}_j^*$ is the actual image closest to $\boldsymbol{\mu}_j$ in the feature space.

*2) Metric $K$-Center:* Metric $K$-Center [19] aims to cluster images in the feature space such that it minimizes the maximum distance from any image to its nearest cluster *center*. In contrast to the $K$-Means objective, it emphasizes coverage and diversity in the feature space, rather than compactness [20].

*a) Clustering Objective:*

$$\min_{\{\mathbf{c}_j\}_{j=1}^{K}} ; \max_{i\in\{1,\ldots,N\}} ; \min_{j\in\{1,\ldots,K\}} \|\mathbf{x}_i - \mathbf{c}_j\|_2.$$

*b) Greedy Center Selection (Farthest-First Traversal):*

$$\mathbf{c}_t = \arg \max_{\mathbf{x}_i}, \min_{j<t} \|\mathbf{x}_i - \mathbf{c}_j\|_2, \qquad t = 2, \ldots, K.$$

*c) Cluster Assignment:*

$$\mathrm{cluster}(i) = \arg \min_{j\in 1,\ldots,K} \|\mathbf{x}_i - \mathbf{c}_j\|_2.$$

*d) Representative Centers:*

$$\mathbf{x}_j^* = \mathbf{c}_j, \qquad \mathbf{x}_j^* \in C_j.$$

where, $\forall j \in \{1, \ldots, K\}$, $C_j$ denotes the $j^{\text{th}}$ cluster induced by center $\mathbf{c}_j$. Each $\mathbf{x}_i \in C_j$ is a data point assigned to cluster $C_j$, and $\mathbf{x}_j^* = \mathbf{c}_j$ is the selected representative (point-center) chosen by the farthest-first traversal.

Figure 4 shows an instance of frames sampled using various strategies for a scene taken from ETH3D-SLAM [21]. 'Farthest Point' refers to the $K$-Center algorithm

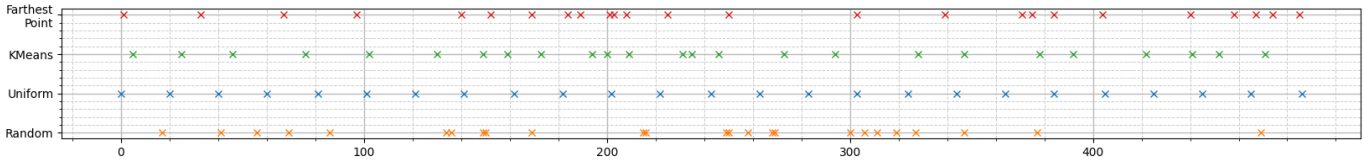


Fig. 4. Example of selected frames for the einstein_1 input scene, farthest point refers to k-centre sampling technique

### D. Weight-Only Quantization

We tested a streaming, sensitivity-aware INT8 post-training quantization scheme for large vision–geometry transformer models. All parameters are processed independently to ensure memory safety.

**(1) Precision Assignment.** Each tensor is classified as FP32 or INT8 based on (a) name-based sensitivity (patch embeddings, positional encodings, LayerNorm parameters, depth/pose heads, and biases) and (b) statistical criteria including parameter count, variance, and outlier ratio. Sensitive tensors are retained in FP32.

**(2) Channel-Axis Detection.** For tensors selected for INT8 quantization, the algorithm automatically determines an appropriate channel axis—the output dimension for 2D matrices or the largest dimension for higher-order kernels—to enable per-channel quantization.

**(3) Mean–Residual Decomposition.** Weights are decomposed as $W = \mu + R$. If subtracting either a per-channel or scalar mean significantly reduces residual variance, the mean $\mu$ is stored in FP32 and only the residual $R$ is quantized. This stabilizes quantization for tensors with strong biases or heavy-tailed distributions.

**(4) INT8 Quantization.** Residuals are quantized using symmetric INT8 with per-channel or per-tensor scaling:

$$q = \text{round}(R/s), \qquad s = \frac{\max|R|}{127}.$$

Both $q$ and $s$ are stored, enabling exact dequantization.

**(5) Streaming Storage and Reconstruction.** Each tensor is quantized and saved in a separate file, avoiding construction of large global dictionaries. During loading, tensors are reconstructed one-by-one via

$$\hat{W} = q \cdot s + \mu,$$

assigning FP32 precision to sensitive components and FP16 to remaining layers.

## IV. Experiments and Results

### A. Experimental Setup

To evaluate the various data selection strategies, we evaluate the VGGT model and the subset selection models quantitatively on camera pose estimation and qualitatively on point cloud generation. We use ETH3D-SLAM [21] and ETH3D-Stereo [22] benchmark datasets, as they are unseen to VGGT during training [1]. To simulate a resource limited environment, all experiments are performed on 15GB VRAM T4-GPU available on free-tier Google Colab with 12.7 GB RAM.

### B. Camera Pose Estimation

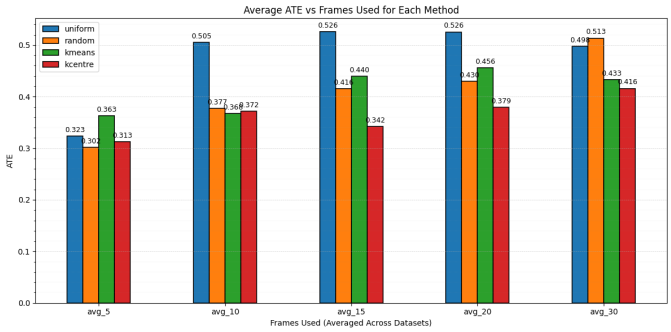


Fig. 5. Average trajectory similarity error for different number of frames over different techniques.

We use all frames of the scenes from ETH3D-SLAM [21] and derive subsets of sizes varied from 5 to 30 using the strategies described in Section III. We extract the camera trajectories estimated by VGGT using the sampled points and align them with the corresponding ground-truth camera trajectories using Umeyama [23] algorithm. The Absolute Trajectory Error (ATE) is then calculated between the aligned predicted camera trajectory $\hat{t}$ and the ground-truth camera trajectory $t_0$ as follows:

$$ATE = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|\hat{t}_i - t_{0,i}\|^2}$$

**Inferences:** Across four input scenes (large_loop_1, einstein_1, sofa_1, and mannequin_1) with frame counts ranging from 467 to 1511, we evaluated subsets of 5, 10, 15, 20, and 30 frames selected using the described sampling strategies. On average, the k-centre method consistently outperforms the others (Figure 5), largely due to its ability to maximize coverage and diversity in the embedding space, thereby capturing a wider range of stereographic viewpoints.

As the number of selected frames increase, absolute deviations from the ground-truth trajectory also increase, leading to higher ATE values overall. For sufficiently large subsets (greater than 5 frames), k-means typically performs better than uniform interval sampling, while random sampling remains unstable and can yield either strong or poor ATE depending on the scene. However, after a certain number of samples, ATE hits a maxima and the overall trend starts decreasing.

From the visualizations in Figure 4, it is evident that k-centre and k-means tend to pick frames that reflect substantial

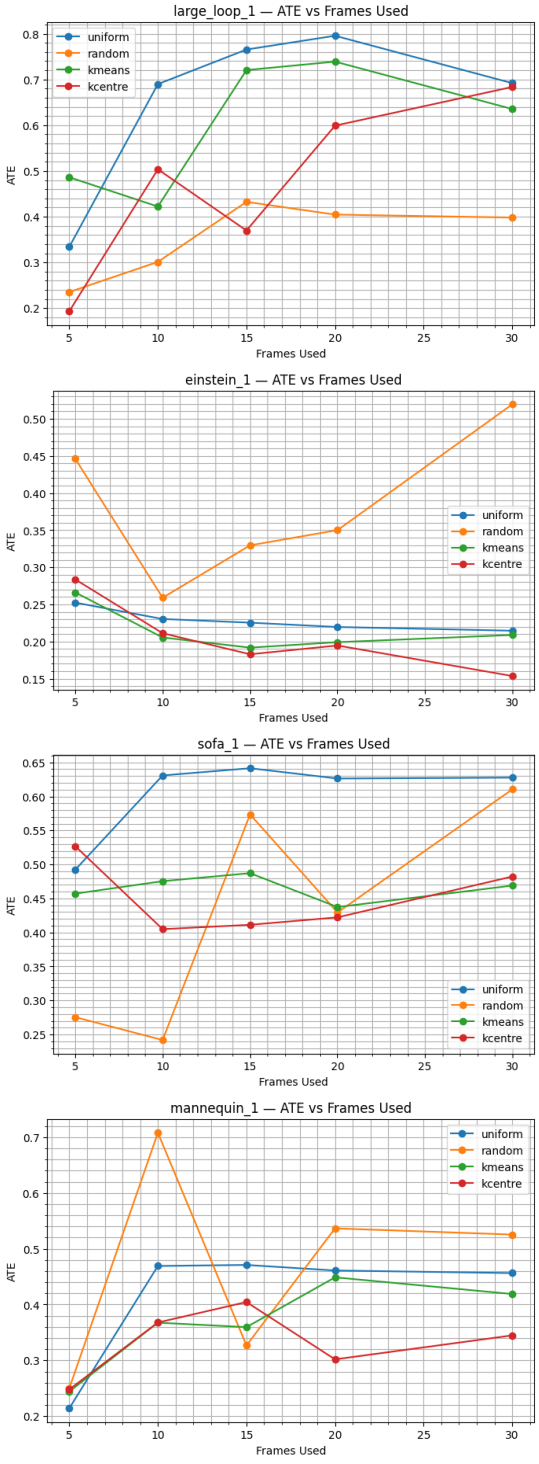


Fig. 6. Trajectory Similarity error vs number of frames used for different input scenes.

viewpoint changes. They cluster more densely in regions where distinct geometric information is present and sparsely where redundancy dominates, which explains their stronger performance compared to uniformly spaced samples.

Figure 6 shows ATE plots for different input scenes over different frame-lengths. It further illustrates that random sampling produces highly variable ATE curves, whereas k-centre achieves the most reliable results across scenes, followed closely by k-means for larger sample sizes.

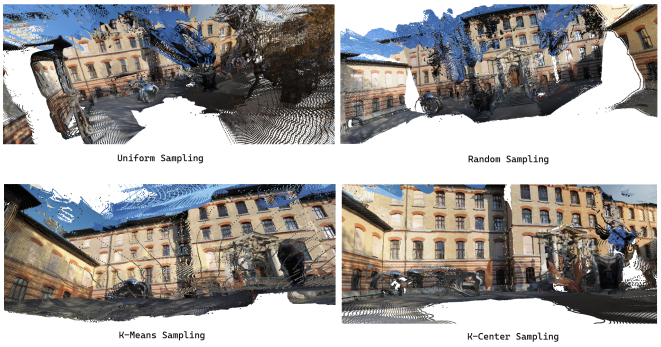


Fig. 7. 3D point cloud generated of DSLR images of a facade. Images from ETH3D-Stereo [22].

### C. 3D Point Cloud Generation

In this section, we present and compare the 3D point clouds generated some scenes from ETH3D-Stereo [22] using our sampling strategies as described in Section III. The 3D point clouds are generated using the best 10 images sampled using the respective strategies as shown in Figures 7 and 8. The point clouds clearly show that a decent amount of generation can be made from a carefully selected subset, and out of all strategies studied, K-Center sampling strategy gives the most reliable generation.

### D. Quantization

After quantization we get total params: 1,256,537,516, INT8 stored: 586,256,384, FP32 stored: 670,281,132 parameters.So about 48% of the weights were quantized. On a single test input of einstein_1 scene, we got comparable ATE result of 0.230307 to that of non-quantised model. The disk space used for scaling values and meta-data along with the new scaled weights was 4675.19 MB.

## V. Conclusion

This work introduces an input data–oriented strategy for optimizing VGGT on devices with limited computational resources by identifying a compact yet highly informative subset of frames. The experiments show that VGGT retains strong performance in 3D reconstruction and camera pose estimation even when the input set is heavily reduced, highlighting the model's robust spatial reasoning and zero-shot generalization. Among the evaluated sampling strategies, the k-center method provides the most effective balance by maximizing input diversity and ensuring broad scene coverage. By lowering both memory usage and computational load, the proposed approach enables more efficient deployment of VGGT for real-time 3D perception in settings such as mobile robotics and augmented reality. This establishes a foundation for future developments in adaptive input selection, multimodal extensions, and model

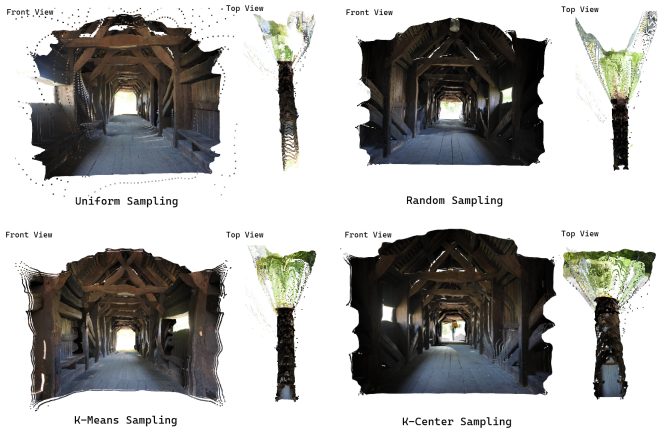


Fig. 8. 3D point cloud generated of DSLR images of a bridge. Images from ETH3D-Stereo [22].

compression techniques aimed at further improving the practicality of VGGT without compromising reconstruction fidelity.

## VI. Future Works

Future work can extend the optimization framework beyond input image selection to jointly consider depth maps, point clouds, and camera parameters. Since VGGT predicts these geometric outputs in a tightly coupled manner, identifying the most informative frames for accurate depth estimation and point cloud reconstruction, together with efficient quantization of both the transformer backbone and the prediction heads, can substantially reduce memory usage and computation. Such a combined approach has the potential to deliver faster and lighter inference on hardware with limited resources. Another direction is to maintain geometric consistency while compressing both the inputs and the model. Evaluating these strategies on diverse real-world datasets and integrating downstream tasks such as dense point tracking will help assess their robustness. These efforts aim to make VGGT practical for real-time 3D perception in robotics, augmented reality, and embedded systems, while preserving reconstruction quality even when the model operates with fewer inputs and a more compact architecture.

## VII. Acknowledgement

We acknowledge the use of OpenAI's ChatGPT for assistance in refining the writing, improving clarity, and organizing certain sections of this report. All conceptual development, experimental design, analysis, and final interpretations were carried out by the authors.

## References

[1] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 5294–5306.

[2] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *CVPR*, 2024.

[3] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," 2024.

[4] R. Scholz, J. Wang, and K. Li, "Efficient vision transformers for edge devices via dynamic token pruning and hybrid quantization," *SSRN Electronic Journal*, 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5319357

[5] S. Zhao and T. Chen, "Enhancing performance of vision transformers on small datasets with local information enhancers," *Pattern Recognition*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0031320324002619

[6] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[7] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 159–12 168.

[8] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker: It is better to track together," in *Proc. ECCV*, 2024.

[9] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker3: Simpler and better point tracking by pseudo-labelling real videos," in *Proc. arXiv:2410.11831*, 2024.

[10] W. Feng, H. Qin, M. Wu, C. Yang, Y. Li, X. Li, Z. An, L. Huang, Y. Zhang, M. Magno, and Y. Xu, "Quantized visual geometry grounded transformer," 2025. [Online]. Available: https://arxiv.org/abs/2509.21302

[11] A. Gholami, S. Yao, S. Venkatakrishnan, B. Wu, K. Keutzer, M. W. Mahoney, K. Hsieh, R. Shao, Z. Zhang, and M. Najibi, "A survey of quantization methods for efficient neural network inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[12] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," 2018.

[13] A. Li, S. Lin, Y. Ma, P. Zhang, K. Yao, S. Ding, P. Hu, and D. Lin, "Brecq: Pushing post-training quantization to 4 bits," in *International Conference on Learning Representations (ICLR)*, 2021.

[14] H. Wei, S. Fang, Y. Du, Z. Yan, H. Wang, and S. Lin, "Qdrop: Randomly dropping quantization for extremely accurate post-training quantization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] T. Xiao, L. Xie, Y. Zhang, X. Peng, K. Lin, Y. Wu, H. Li, S. Han, and Z. Zhang, "Smoothquant: Accurate and efficient post-training quantization for large language models," 2023.

[16] S. Ashkboos, V. Georgiev, A. Samragh, F. Zhang, and D. Song, "Quarot: Quantization-friendly rotations for post-training quantization," in *International Conference on Machine Learning (ICML)*, 2024.

[17] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[18] J. MacQueen, "Multivariate observations," in *Proceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability*, vol. 1, 1967, pp. 281–297.

[19] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Mathematics of Operations Research*, vol. 10, pp. 180–184, 1985.

[20] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations (ICLR)*, 2018.

[21] T. Schöps, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.