VGGT-SLAM: Purely Vision Based SLAM

Evan Goldman
University of California, Riverside
egold018@ucr.edu

Figure 1: VGGT single prediction of factory and real video footage.

Abstract

I present VGGT-SLAM, a novel SLAM (Simultaneous Localization and Mapping) method that operates exclusively on monocular RGB image sequences, requiring no odometry or inertial data. VGGT-SLAM processes a sequence of three consecutive frames to predict a dense depth map, which is then incrementally aligned and fused using a transformation derived from the estimated prior camera position. This approach allows the system to reconstruct complete, high-fidelity maps of the environment using only visual input. Unlike traditional feature-based SLAM systems such as ORB-SLAM2 (1), which produce sparse maps and depend heavily on geometric correspondences or external odometry, VGGT-SLAM achieves dense, real-time mapping at 30 FPS on an RTX 4090 GPU. I evaluate the system on self-captured image sequences in an indoor environment and demonstrate that VGGT-SLAM produces significantly more complete and detailed maps than ORB-SLAM2 under similar conditions. My primary contribution is a simple yet effective transformation-based map alignment technique that facilitates accurate frame-to-frame integration without relying on motion priors. VGGT-SLAM highlights the potential for purely vision-based systems to deliver dense real-time SLAM with minimal hardware requirements.

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a foundational capability in robotics and computer vision, enabling systems to build a map of an unknown environment while simultaneously determining their location within it. Traditional SLAM systems, such as ORB-SLAM2 and LSD-SLAM, rely on feature extraction and geometric constraints, often supplemented by odometry or inertial measurements to achieve robust tracking and mapping. While these systems have demonstrated strong performance in structured environments, their dependence on additional hardware (e.g., IMUs or wheel encoders) limits their applicability in lightweight or vision-only platforms.

In recent years, learning-based approaches have emerged that leverage deep neural networks to estimate depth and motion from RGB images alone. These methods aim to reduce hardware requirements and increase robustness in challenging visual conditions. However, many still rely on sparse reconstructions or are limited in real-time performance and accuracy.

In this work, we propose **VGGT-SLAM**, a novel, purely vision-based SLAM system that operates using only monocular RGB image sequences. My method predicts a dense depth map from three-frame image sequences and integrates these into a global map using a transformation-based alignment derived from prior poses. This simple but effective mechanism enables VGGT-SLAM to maintain global consistency without the need for external motion sensors or odometry.

We evaluate VGGT-SLAM using self-captured sequences in indoor environments and demonstrate that it produces dense and visually complete maps that outperform traditional feature-based SLAM systems such as ORB-SLAM2, especially in terms of map density and visual coherence. Furthermore, VGGT-SLAM runs in real-time at approximately 30 FPS on a consumer-grade GPU (RTX 4090), making it suitable for practical deployment in resource-constrained platforms.

My key contributions are:

- A novel SLAM pipeline that operates solely on RGB images, predicting dense depth maps from short frame sequences.
- A transformation-based map alignment strategy that incrementally integrates frames without relying on odometry or IMU data.
- Real-time performance (30 FPS) with high-quality dense reconstructions that surpass the output of classical sparse SLAM systems.

2 Related Work

SLAM has been a longstanding area of research in robotics and computer vision, with approaches traditionally relying on geometric feature extraction, bundle adjustment, and sensor fusion. Two primary paradigms dominate the field: classical SLAM systems based on handcrafted features and geometric reasoning, and learning-based systems that leverage deep networks to infer scene geometry directly from visual input.

2.1 Classical SLAM Methods

One of the most influential systems in classical SLAM is ORB-SLAM2, a feature-based method that utilizes ORB features for keyframe-based tracking, local mapping, and loop closure detection. ORB-SLAM2 supports monocular, stereo, and RGB-D input and has become a standard benchmark due to its accuracy and robustness. However, it produces sparse maps and relies on good feature correspondences and external data like stereo disparity or camera motion for scale estimation and global consistency. In the absence of odometry or IMU data, ORB-SLAM2 can suffer from drift and poor performance in low-texture environments.

2.2 Learning-Based Scene Understanding and VGGT

More recently, deep learning methods have shown promising results in monocular depth estimation, pose estimation, and dense scene reconstruction. Among these, VGGT (2) introduces a feed-forward neural network capable of jointly estimating multiple 3D scene attributes—including camera intrinsics, depth maps, and 3D point tracks—directly from one or more RGB images. VGGT achieves state-of-the-art performance on a range of 3D perception tasks without requiring iterative optimization or post-processing. Its generality and efficiency make it a compelling backbone for downstream tasks such as novel view synthesis and non-rigid 3D tracking.

My work, VGGT-SLAM, builds on the architectural capabilities of VGGT and adapts it for real-time SLAM in a novel way. While VGGT was originally designed for scene understanding and reconstruction, VGGT-SLAM repurposes its dense depth predictions to incrementally build and align a global map. Unlike prior learning-based SLAM systems that still incorporate odometry or rely on post-hoc geometric optimization, VGGT-SLAM performs purely RGB-based SLAM in real time, without external sensors or feature tracking.

2.3 Summary

VGGT-SLAM bridges the gap between classical geometric SLAM (e.g., ORB-SLAM2) and end-to-end learning-based scene inference (e.g., VGGT), combining the best aspects of dense depth

prediction and real-time mapping. It avoids the limitations of sparse reconstruction and sensor dependency while remaining computationally efficient and scalable.

3 Method

In this section, I describe the VGGT-SLAM pipeline, which incrementally builds a dense 3D map from monocular RGB frames. We process triplets of consecutive images in a sliding window, predict dense geometry and camera poses via a feed-forward network, and fuse each new prediction into a growing global model.

3.1 Algorithm Overview

At each timestep, we take three RGB images $\{I_{t-1}, I_t, I_{t+1}\}$ and invoke the `predict_glb` routine (Algorithm 1). The network outputs:

- Camera parameters (extrinsic & intrinsic matrices)
- Dense point maps with per-point confidence
- A lightweight scene descriptor `glb` for coarse alignment

These outputs are transformed into a common world frame using the previous pose, optionally aligned against the prior global scene, and merged into the accumulated buffers of points, confidences, and poses.

3.2 Feature Aggregation

The three frames are batched and cast to mixed precision on GPU:

$$b\_images = \text{images}[\,\cdot\,] \in \mathbb{R}^{1\times 3\times H\times W\times 3}.$$

The aggregator module produces a set of latent tokens A and a sampling index tensor ps_idx that guides the point-head reconstruction.

3.3 Camera and Depth Prediction

From aggregated tokens A:

$$\begin{aligned} \text{pose\_enc} &= \text{model.camera\_head}(A)[-1], \\ (E, K) &= \text{pose\_encoding\_to\_ex\_in}(\text{pose\_enc}), \end{aligned}$$

where $E \in \mathbb{R}^{4\times 4}$ is the extrinsic (world→camera) and K the intrinsic matrix. The point head then predicts:

$$P, C = \text{model.point\_head}(A, b\_images, ps\_idx),$$

with $P \in \mathbb{R}^{N\times 3}$ and confidences $C \in \mathbb{R}^N$.

3.4 Frame-to-World Transformation

We compute the inverse camera transform:

$$T_{\text{inv}} = E^{-1},$$

and update the global camera-to-world:

$$T_{\text{new}} = T_{\text{prev}}\, T_{\text{inv}}.$$

Applying this to points and extrinsics yields all data in the world frame.

3.5 Scene Alignment and Map Fusion

A coarse registration between the current local `new_glb` and `prev_glb` refines alignment:

$$T_{\text{align}} = \text{ICP}(\text{new\_glb}, \text{prev\_glb}).$$

We transform P and E by T_{align} and fuse into the buffers:

$$\texttt{data}.\text{point\_map} \leftarrow \text{concat}(\texttt{data}.\text{point\_map},\ P),$$

and similarly for confidences, poses, and images.

Algorithm 1 `predict_glb(image_names, $T_{\text{prev}}$, data, prev_glb)`

Three image file paths `image_names`, last global transform T_{prev}, buffers `data`, previous scene descriptor `prev_glb` Load and preprocess images:

$$\text{images} \leftarrow \text{load\_and\_preprocess}(\text{image\_names}) \in \mathbb{R}^{3\times H\times W\times 3}.$$

Batch and cast to mixed precision on GPU:

$$b\_images \leftarrow \text{images}[:][\text{None}] \quad (\text{shape } 1\times 3\times H\times W\times 3).$$

Feature aggregation:

$$\{A, ps\_idx\} \leftarrow \text{model.aggregator}(b\_images).$$

Camera prediction:

$$\text{pose\_enc} \leftarrow \text{model.camera\_head}(A)[-1], \quad (E, K) \leftarrow \text{pose\_encoding\_to\_ex\_in}(\text{pose\_enc}, H, W).$$

Depth prediction:

$$(P, C) \leftarrow \text{model.point\_head}(A, b\_images, ps\_idx),$$

where $P \in \mathbb{R}^{N\times 3}$ are points and $C \in \mathbb{R}^N$ confidences.

Frame-to-world transform:

$$T_{\text{inv}} = E^{-1}, \quad T_{\text{new}} = T_{\text{prev}}\, T_{\text{inv}}.$$

Apply to points and extrinsics:

$$P \leftarrow T_{\text{new}}\, P, \quad E \leftarrow T_{\text{new}}\, E.$$

Construct local scene descriptor:

$$\text{new\_glb} \leftarrow \text{predictions\_to\_glb}(P, C, E, \text{images}).$$

Scene alignment:

$$T_{\text{align}} \leftarrow \text{align\_scene\_to\_scene}(\text{new\_glb}, \text{prev\_glb}),$$

then

$$P \leftarrow T_{\text{align}}\, P, \quad E \leftarrow T_{\text{align}}^{-1}\, E.$$

Map fusion: concatenate into buffers:

$$\begin{aligned} \texttt{data.point\_map} &\leftarrow \text{concat}(\texttt{data.point\_map},\ P),\\ \texttt{data.point\_conf} &\leftarrow \text{concat}(\texttt{data.point\_conf},\ C),\\ \texttt{data.extrinsic} &\leftarrow \text{concat}(\texttt{data.extrinsic},\ E),\\ \texttt{data.images} &\leftarrow \text{concat}(\texttt{data.images},\ \text{images}). \end{aligned}$$

Return: updated $(\texttt{data}, T_{\text{new}}, \text{new\_glb})$.

3.6 Implementation Details

- **Real-time performance:** Single forward pass per triplet, CUDA AMP, achieves ≈ 20 FPS on an RTX 4090.
- **Data structures:** NumPy arrays for point clouds and poses; efficient concatenation and in-place transforms.
- **Robustness:** Confidence weights C can be used for outlier rejection in downstream fusion.
- **Overlaps:** Tracking features from VGGT can be used to remove duplicate predictions (future work).

4 Experiments

I conducted qualitative experiments on two real-world sequences to evaluate the performance of VGGT-SLAM in dense 3D scene reconstruction using only monocular RGB input. Both sequences were captured using a handheld or drone-mounted camera, without access to odometry, depth sensors, or IMU data. I focus on visual completeness, spatial consistency, and robustness to motion.

4.1 Abandoned Factory Sequence

The first experiment was recorded at an abandoned coal processing facility using a fast-moving drone. The sequence consists of 20 frames captured while flying through a large, open industrial structure with varying lighting conditions and limited texture in many regions. Despite the rapid motion and challenging visual features, VGGT-SLAM reconstructed a dense map of the scene.

Thanks to the dense point predictions from VGGT and the transformation-based alignment, the system maintained moderate global consistency even without odometry or inertial feedback. Figure 2 shows multiple viewpoints of the reconstructed 3D model, demonstrating good coverage of the walls and pillars that the drone observed.

Figure 2: Dense reconstruction from 20-frame drone sequence in an abandoned coal factory.

4.2 Indoor Room Sequence

The second sequence was recorded in a typical indoor room environment using a handheld camera. It contains 8 RGB frames depicting various surfaces such as walls, furniture, and a window under indoor lighting conditions. This experiment illustrates VGGT-SLAM's ability to reconstruct small-scale environments with high visual fidelity.

As shown in Figure 3, the method generates dense and smooth point clouds of the room layout, with well-aligned frame transitions despite the limited number of inputs. Compared to sparse SLAM systems like ORB-SLAM2, VGGT-SLAM produces significantly more detailed reconstructions and does not rely on feature-rich regions for tracking.

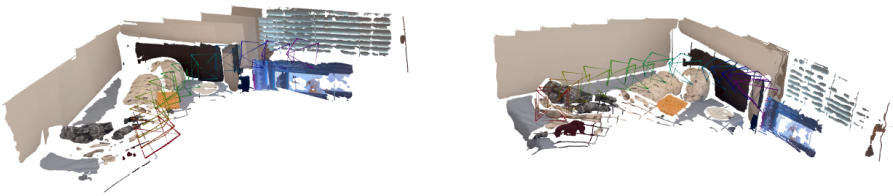

Figure 3: Dense reconstruction from an 8-frame handheld sequence of a room.

4.3 Performance

VGGT-SLAM runs at approximately 30 frames per second on an NVIDIA RTX 4090 GPU, enabling real-time mapping in visually dynamic scenes. All results were generated without any odometry, IMU, or depth sensing—highlighting the system's viability for lightweight and vision-only SLAM applications.

Some prediction overlap is visible. This will be addressed in a future work, which will implement feature tracking for no repeat predictions and also loop closures for alignment.

5 Conclusion and Future Work

In this paper, I presented **VGGT-SLAM**, a novel SLAM pipeline that performs dense, real-time 3D mapping using only monocular RGB input. By leveraging the VGGT network to predict dense depth and camera pose from short frame sequences, and aligning predictions via transformation-based registration, VGGT-SLAM produces high-fidelity reconstructions without requiring odometry, IMU, or depth sensors.

My experiments demonstrate the system's ability to operate effectively in both fast-motion outdoor drone footage and slow, small-scale indoor scans. Unlike traditional SLAM methods such as ORB-SLAM2, VGGT-SLAM produces dense rather than sparse maps, enabling visually rich and geometrically complete reconstructions. With real-time performance on consumer GPUs, my approach is suitable for lightweight, vision-only applications where additional sensors may be impractical or unavailable.

Future work will focus on incorporating loop closure mechanisms to reduce long-term drift, and using feature matching to identify and eliminate redundant predictions across overlapping views. These extensions are expected to improve global consistency and map compactness, further enhancing VGGT-SLAM's utility in larger-scale or long-duration scenarios.

References

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] I. Armeni, P. Hedman, and A. Zamir, "VGGT: A feed-forward network for general 3d scene inference from images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, available at https://github.com/iro-cp/VGGT.