Building TUM in a Day

Timo Class Arda Nacar Han Wu
Computer Vision Group
TUM
timo.class@tum.de arda.nacar@tum.de hann.wu@tum.de

Abstract

We present a metric 3D reconstruction pipeline that leverages recent transformer-based models for dense reconstruction of large-scale real-world environments. Building upon VGGT, our method generates a 3D scene reconstruction from monocular RGB videos and addresses key challenges in scalability, metric consistency, and dense surface recovery. To achieve this, we introduce strategies for large-scale alignment across video chunks, integrate MAST3R for scale estimation, and employ NKSR and CityGaussian for dense reconstructions. We evaluate our pipeline on a custom dataset captured at the TUM Mathematics and Informatics building, which features reflective glass facades, a transition from outside to inside, and an extended atrium-like hall. Our results demonstrate that the proposed approach preserves local accuracy, highlighting both the strengths and limitations of current transformer-based 3D reconstruction methods in practical deployment scenarios.

1. Introduction

Three-dimensional (3D) reconstruction has long been a central problem in computer vision, aiming to recover the geometry and appearance of scenes from visual data. Classical methods such as Structure-from-Motion (SfM), Multi-View Stereo (MVS), and SLAM-based systems have laid the foundations of this field, powering applications in Robotics, Mapping, and Augmented Reality. However, these approaches often struggle in challenging scenarios involving large-scale environments, reflective surfaces, low-texture regions, or transitions between indoor and outdoor settings. Recent transformer-based networks, such as VGGT [27], have advanced the state of the art by predicting point clouds, depth maps, and camera parameters directly from image sequences. Although efficient and accurate, VGGT operates in a normalized scale, limiting its ability to produce metrically consistent reconstructions required for real-world applications.

In this work, we develop a metric 3D reconstruction pipeline that leverages VGGT for dense reconstruction of the TUM Mathematics and Informatics building. This setting presents challenging characteristics, including reflective glass facades, a transition from outside to inside, and an extended atrium-like hall. To address these, our pipeline combines VGGT with strategies for large-scale alignment, metric scaling, and dense surface recovery.

Our contributions are: (1) a unified pipeline for scalable, metrically accurate 3D reconstructions, (2) adaptations for robustly handling outdoor-indoor transitions, and (3) a real-world case study that highlights both the capabilities and limitations of current transformer-based reconstruction systems.

2. Related work

3D reconstruction encompasses a wide range of methods aimed at recovering the geometric structure and appearance of scenes and objects from sensor data. Since different applications impose varying requirements on scale, accuracy, and completeness, the task can be divided into several sub-domains. For this work, we consider four categories: Point Cloud Reconstruction, Large-scale Reconstruction, Metric Reconstruction and Dense Reconstruction. Each sub-domain addresses distinct challenges and contributes complementary perspectives to the problem.

Point Cloud Reconstruction focuses on producing sets of 3D points that capture the geometry of a scene or object. Over the years, a wide variety of approaches have been proposed, reflecting different assumptions about the input data and the desired level of detail. Classical methods such as Structure from Motion (SfM) [7, 15, 22], Multi-View Stereo (MVS) [8, 23] and SLAM-based systems [3, 18, 19] form the foundation of this field, relying on robust feature detection and correspondence matching across images to recover camera poses and triangulate 3D points [10] . These methods remain widely used and effective, particularly in highly textured environments. However, their performance often degrades in scenarios with repetitive structures, low-texture regions, or wide

viewpoint variations.

Early learning-based methods focused on monocular depth estimation, which aims to infer dense depth maps from a single RGB image. Approaches such as Monodepth2 [9], ZoeDepth [2] and DepthAnythingV2 [29] demonstrated that deep networks can generalize geometric priors across diverse scenes and datasets. These works established the feasibility of feed-forward inference for geometry, setting the stage for more general reconstruction pipelines.

Recent advances in deep learning have substantially expanded the capabilities of point cloud reconstruction [14, 20, 26, 31]. Data-driven methods leverage learned priors to recover plausible geometry even under challenging conditions. Transformer-based architectures [25], in particular, have advanced the field by modeling global scene context and long-range dependencies across views [4, 6, 28]. VGGT [27] has emerged as a state-of-the-art model capable of inferring point clouds, depth maps, camera parameters, and tracking points in a feed-forward manner. By avoiding explicit geometric optimization, such models combine efficiency with high accuracy, making them especially suitable for large-scale reconstruction tasks.

Large-scale Reconstruction addresses the challenge of extending 3D reconstruction methods to expansive environments such as entire buildings, large indoor spaces, city blocks or outdoor landscapes. Unlike small-scale reconstruction, where accuracy and detail are the primary focus, large-scale tasks introduce challenges in scalability, computational efficiency and robustness. Algorithms must process large amounts of data, while preserving geometric continuity across extended spaces. A common strategy is to partition the environment into smaller chunks, reconstruct them locally, and perform local or global optimization to ensure consistency across all chunks. Well-known examples include systems like *Build Rome in a Day* [1], which demonstrates the feasibility of reconstructing entire cities from large-scale photo collections, and more recent models such as *Vggt-Long* [5], which extend transformer-based approaches to efficiently handle long input sequences.

Metric Reconstruction extends the 3D reconstruction by predicting geometries at real-world scale. One line of work solves this as a monocular depth estimation problem [11, 30]. However, depth inference from a single image remains challenging, as it requires knowledge of both the object's true scale and the camera's focal length. Metric3Dv2 [11] addresses this challenge by applying a canonical transformation to the input image, effectively normalizing it to a predefined focal length. Depth is then predicted in canonical space, and the final metric estimates are obtained by applying the inverse (de-canonical) transformation to the predicted depth map. On the other hand, MAST3R [14] predicts a metric point cloud from image pairs by allowing stereo information to be utilized by a cross attention mechanism and training on metric ground truth data. It predicts point clouds for each image, expressed in the coordinate frame of the first image, and employs a global alignment step to generalize to arbitrarily long sequences.

Dense Reconstruction aims to recover continuous, high-fidelity geometry and appearance beyond sparse point representations. Current approaches largely follow two complementary directions: surface-first recovery from point clouds and radiance/primitive-based optimization from images. On the point cloud side, NKSR learns a neural kernel field over large, sparse and noisy point clouds. It solves a gradient-fitting objective with compactly supported kernels, enabling sparse solvers, out-of-core scaling, and robust watertight surface generation [12].

On the image side, CityGaussian extends 3D Gaussian Splatting [13] to city-scale with divide-and-conquer training, fusion with a global prior, and block-wise level-of-detail scheduling for stable real-time rendering [16]. CityGaussianV2 improves geometric accuracy and efficiency via decomposed-gradient densification with depth regression, an elongation filter to prevent Gaussian explosion, and parallel train-compress scheduling for large scenes [17].

3. Method

Our proposed method processes monocular RGB videos to reconstruct 3D scenes, represented either as meshes or as 3D Gaussians. We formulate the problem in terms of four sub-tasks: Point Cloud Reconstruction, Large-scale Reconstruction, Metric Reconstruction and Dense Reconstruction. Our method preserves the local accuracy of VGGT, while remaining robust in challenging scenes, including transitions from outdoor to indoor environments.

3.1. Point Cloud Reconstruction

This task aims at creating an intermediate, local 3D reconstruction of the scene that can be used for further processing. We employ the recent transformer-based model VGGT to obtain an accurate 3D point cloud P_i from an RGB input sequence $(I_i)_{i=1}^{K}$. In our approach, instead of directly using the predicted 3D point maps by VGGT, we unproject the predicted depth maps D_i into 3D space using the camera parameters g_i yielding a higher accuracy.

3.2. Large-scale Reconstruction

This task focuses on the computational feasibility of reconstructing large scenes. It generates a large intermediate point cloud of the whole scene given spatially ordered RGB videos $(V_i)_{i=1}^{M}$. This process is divided into the following

tasks: Chunking and Local Alignment.

Chunking. To enable detailed and computationally feasible processing of large scenes, we chunk the scene into spatially overlapping batches of 2 videos $(B_i)_{i=1}^{M-1}$ with $B_i = \{V_i, V_{i+1}\}$. The batch size is limited by the available GPU memory, which indirectly determines the maximum number of images N that can be processed at once. Using the largest possible batch size for a given GPU allows the model to maximize detail and fully utilize the available computational resources. In a first step, we uniformly oversample frame proposals for each video. Given a video with T frames $V = \{I_0, I_1, \dots, I_{T-1}\}$, we generate an overcomplete set of $L > \frac{N}{2}$ proposals by uniformly selecting frame indices

$$t_k = \left\lfloor \frac{k}{L-1}(T-1) \right\rfloor, \quad k = 0, 1, \dots, L-1. \tag{1}$$

The corresponding proposal set of frames is

$$\mathcal{V} = \{I_{t_k} \mid k = 0, 1, \dots, L-1\}. \tag{2}$$

To increase the informativeness of the images, we filter for blurriness. Given a (grayscale) image $I \in \mathbb{R}^{H \times W}$, we compute its discrete Laplacian

$$\Delta I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}, \tag{3}$$

and define the sharpness as the variance of the Laplacian $\sigma_L^2 = Var_{(x,y)}(\Delta I(x, y))$. An image is considered not blurry if $\sigma_L^2 > \tau$ where τ is experimentally determined such that it excludes low-textured outlier frames [21]. The batches of images $\mathcal{B} = \{I_0, ..., I_{N-1}\}_{i=1}^{M-1}$ are then created by uniformly sample $\frac{N}{2}$ frames per video from the filtered frames.

Local Alignment. Each batch of images $B_i = \{I_0, ..., I_{N-1}\}$ is processed by VGGT as in Sec. 3.1, resulting in $M-1$ point clouds $\{P_i\}_{i=1}^{M-1}$. The ordering of the sequential input allows to create a cumulative transformation chain. Consecutive batches overlap in one video segment, such that the relative transformation between two batches can be directly computed from the camera parameters predicted by VGGT. Let $H_b^a \in SE(3)$ denote the rigid transformation that maps points from the coordinate frame b into the coordinate frame a. For two consecutive batches i and $i+1$, we obtain $\frac{N}{2}$ candidate estimates of the relative transformation. Specifically, for each overlapping frame $j \in \{1, ..., \frac{N}{2}\}$, VGGT provides its extrinsics in the coordinate system of batch i, $H_i^{j,i}$, and once in that of batch $i+1$, $H_{i+1}^{j,i+1}$. By comparing these two representations of the same frame, we obtain an estimate of the relative batch-to-batch transformation by computing

$$H_{i+1}^i = (H_i^{j,i})^{-1} H_{i+1}^{j,i+1}. \tag{4}$$

Collecting all estimates yields the set

$$\mathcal{H}_{i+1}^i = \{(H_i^{1,i})^{-1} I, (H_i^{j,i})^{-1} H_{i+1}^{j,i+1}, ...\} \tag{5}$$

where the first element, involving the identity I, corresponds to the alignment of the reference frame (defined as the first frame passed to VGGT) and $j \in \{2, ..., \frac{N}{2}\}$.
To obtain an accurate transformation, we average the matrices by mapping the transformation matrices from Lie group $H \in SE(3)$ to Lie algebra $\xi \in \mathfrak{se}(3)$, compute the weighted average of the corresponding twist vectors

$$\bar{\xi} = \frac{2\,\xi_1 + \sum_{j=2}^{\frac{N}{2}-1} \xi_j}{\frac{N}{2} + 1} \tag{6}$$

and map the results back to $H \in SE(3)$. Note that here the first twist is weighted by two since its transformation matrix only consists of one estimate and the identity (see first element in Eq. (5)). We obtain a chain of transformations between the batches

$$\mathcal{H} = \{H_2^1, H_3^2, ..., H_{M-1}^{M-2}\} \tag{7}$$

which can be cumulatively applied to obtain the point clouds ${}^i p$ with respect to the same reference frame

$$\{{}^1 p\} = \left\{ \left(\prod_{k=1}^{i} H_{k+1}^k \right) {}^i p \right\}_{i=1}^{M-1}. \tag{8}$$

Since each batch is processed independently by VGGT, the resulting point clouds vary in scale. While their orientations are aligned, the translation and scale require correction. Given two batches, we know the exact overlap of the respective point clouds. Using the weighted Umeyama algorithm [24] solving for

$$\min_{R^*, t^*, s^*} \frac{1}{N} \sum_{j=0}^{N} w_j ||x_{i,j} - SR x_{i+1,j} + t||^2 \tag{9}$$

which aligns the next point cloud x_{i+1} to the current point cloud x_i, weighted by the predicted confidence $w_j = \sqrt{c_{x_{i+1,j}} c_{x_{i,j}}}$, results in the closed-form-solution for the optimal, relative $SIM(3)$ transformations. To robustify the local alignment, we only consider the most confident points of the overlapping point clouds by filtering for a confidence threshold $\tau_{align} \in (0, 1)$. The corrected point clouds are obtained in the same way as explained in Eq. (8).

3.3. Metric Reconstruction

Achieving metric 3D reconstruction cannot rely on VGGT alone, as it only produces normalized 3D predictions. In contrast, MAST3R leverages metric ground truth during training, resulting in improved metric accuracy. Nevertheless, VGGT cannot simply be replaced by MAST3R

within our pipeline, since VGGT provides more consistent and geometrically accurate predictions. To combine the strengths of both models, our approach employs MAST3R solely to recover a global scale, which is then applied to the normalized predictions of VGGT. Based on this design, we implement the following pipeline.

Sub-batch Extraction. We subdivide each batch B_i in the reconstruction pipeline into three equal-sized sub-batches. The corresponding frame indices are

$$K_n = \{0\} \cup \{\, i \mid 3 \leq i < L_i \ \wedge \ i \equiv n-1 \pmod 3\} \tag{10}$$

where L_i is the length of B_i and $n \in \{1, 2, 3\}$. Then, each sub-batch is further down sampled to L_{sub} frames analogous to Eq. (1). We call these down sampled indices $K_n^{'}$ and define the ultimate sub-batches as

$$B_i^n = \{I_{i,k} \mid k \in K_n^{'}\}. \tag{11}$$

It is reasonable to select a small L_{sub}, since the purpose of the scale estimation step is not to generate the final 3D reconstruction. The computational overhead of this procedure should remain negligble compared to that of VGGT. In practice, we set $L_{sub} = 4$.

Scale Estimation. For each sub-batch B_i^n, we infer a metric 3D point cloud Q_i^n using MAST3R. To estimate the relative scale between the predictions of MAST3R and VGGT, the corresponding point clouds must be aligned. Specifically, the MAST3R prediction for sub-batch B_i^n is aligned with the VGGT prediction P_i for the full batch B_i. Since both Q_i^n and P_i are represented in the same key frame, their relative rotation is the identity. Consequently, the alignment problem reduces to minimizing

$$L(s,t) = \frac{1}{N}\sum_{j=1}^{N} ||s x_j + t - y_j|| \tag{12}$$

with $N = |Q_i^n|$, $x_j \in P_i$ and $y_j \in Q_i^n$. Note that $|Q_i^n| \neq |P_i|$. So, we apply closest-point matching to obtain correspondences (x_j, y_j). Given these pairs, the energy term $L(s,t)$ can be minimized in closed form as

$$s_i^n = \frac{cov(X,Y)}{\sigma_X^2} \tag{13}$$

and

$$t_{i,n} = \bar{y} - s_i^n \bar{x} \tag{14}$$

where $\bar{x}$ and $\bar{y}$ are are the centroids of X and Y respectively. Since the initial point correspondences may be inaccurate, equation Eq. (12) can be considered as a function of the correspondences (x_j, y_j) and minimized iteratively in an ICP-like fashion.

After computing s_i^n for all pairs (Q_i^n, P_i) we obtain three relative scale estimates (scale corrections) for each point cloud P_i. The variance of these three estimates is used as a confidence measure for the scale estimation on batch B_i. Finally the global scaling factor for the whole scene is computed as

$$s^* = \frac{1}{3}\sum_{n=1}^{3} s_{i^*}^n \tag{15}$$

with $i^* = \arg\min_i(var(\{s_i^n\}_{n=1}^3))$. Note that this estimate only considers the most confident batch. Alternatively, a confidence-weighted average scale of all the batches could be used.

3.4. Dense Reconstruction

Triangle Mesh Extraction with NKSR. We employ NKSR [12] as one of our dense reconstruction methods. NKSR predicts a sparse voxel hierarchy and a kernel function for each voxel, given an input point cloud. These kernel functions can then be used to compute a signed distance function (SDF), from which the zero iso-surface can be extracted.
The model includes a tunable voxel size hyperparameter, defined as the "voxel size of the finest level in the sparse voxel hierarchy". The authors recommend setting this hyperparameter approximately to the standard deviation of the noise in the input point cloud. In our case the standard deviation of the noise level is unknown. Therefore, we implement a noise estimation procedure.
Due to memory constraints, we utilize the "chunked mode" of NKSR, which divides the point cloud into overlapping chunks and predicts an SDF for each chunk. These per-chunk predictions are subsequently merged. Since this mode does not allow direct specification of the voxel size, we scale our input point cloud by $0.1/\textit{voxel\_size}$ following the guidelines in the official repository.
At a given 3D point x, the noise level can be estimated as the spread of its neighboring points about the normal plane of x. However this is only accurate for x that lie on planar surfaces. So we need to identify neighborhoods of points lying on approximately planar regions. To this end, we first downsample the point cloud by a factor of 0.01, yielding P_{down}, to reduce computational cost. For each point $x_i \in P_{down}$, we compute a normal vector based on its K nearest neighbors. Once the normals are computed, the *curvature* at a point x_i is evaluated as the deviation of the mean of its neighboring normals $n_{i,k}$ from its normal n_i. Formally,

$$curve(x_i) = deg\left(arccos\left(\frac{1}{K}\sum_{k=1}^{K} n_{i,k} \cdot n_i\right)\right). \tag{16}$$

The standard deviation of the data noise is calculated as

$$\sigma = mean\left(std\left(\{D_i | curve(x_i) < 5\}\right)\right), \tag{17}$$

where

$$D_i = \{(x_i - x_{i,k}) \cdot n_i\} \tag{18}$$

is the set of deviations of neighboring points $x_{i,k}$ around the surface defined by (x_i, n_i).

CityGaussian-Based Dense Reconstruction. CityGaussian is a Gaussian-splatting framework tailored for scalable city-scale 3D reconstruction. We employ CityGaussianV2 in our pipeline, initializing each block's Gaussians from the VGGT point cloud: means $\boldsymbol{\mu}_k \leftarrow x_k \in P_i$, initial $\boldsymbol{\Sigma}_k$ from local covariance, α_k from point density, and SH colors from per-view RGB. After SIM(3) alignment and global fusion, we build the LoD hierarchy and fine-tune, producing a city-scale, dense radiance model that preserves VGGT accuracy and remains real-time renderable.

4. Experiments

To reconstruct a TUM building, we collected data tailored to the requirements of our pipeline. Our focus was the Mathematics and Informatics (MI) department building, which is characterized by an extended atrium-like hall with multiple side corridors. In particular, we targeted the exterior entrance and the front section of the entrance hall.
This task is particularly challenging due to several factors. First, the entrance features a large glass facade, which complicates depth estimation and makes it difficult to separate interior from exterior regions. Reflective metal surfaces, such as those on the lecture hall wall, introduce further ambiguity. The transition from outdoor and indoor spaces poses an additional challenge, as the illumination and appearance differ drastically between these domains. Furthermore, the data distribution differs substantially from that of the VGGT training set, which limits the model's ability to generalize. In particular, the length of the atrium-like hall leads to distorted views, where similar perspectives can produce inconsistent depth estimates, resulting in warped or fragmented point clouds.

4.1. Data Collection Method

To ensure a high-quality reconstruction, we divided the building into three spatial regions and adapted our data collection strategy accordingly: (1) the exterior, (2) the transition zone (entrance), and (3) the interior. For each part, we applied a tailored approach while maintaining sufficient overlap between adjacent regions to support a robust reconstruction.

Outside
For the exterior, we focused on the entrance view, which includes the side wall of one corridor section, the main entrance, and the reflective lecture hall facade (Fig. 2, Top left). Data was collected using static videos covering approximately 180 degrees, ensuring complete coverage of the front area from fixed viewpoints.

Transition
The entrance consists of an automatic sliding glass door framed by a red box. Its fully transparent surface makes capturing a smooth exterior-interior transition particularly challenging (Fig. 2, Top right). Data was collected using moving video sequences, starting from a static viewpoint outside and ending at a static viewpoint inside. During the sequences, the camera performs a 180-degree rotation to ensure alignment between the initial and final orientations.

Inside
While it may seem intuitive to collect interior data in the same way as the exterior, experiments revealed that this approach does not yield accurate and consistent point clouds. The main issue arises from large rotations along the entrance hall, which produce views with highly varying depths. As a result, VGGT's accuracy degrades, and the 3D points reconstructed from certain views fail to align consistently with those from others within the same batch (see Fig. 1). We attribute this discrepancy to limited generalization of VGGT's training data to our specific dataset. To mitigate this issue, we simplify the task by adopting a different data collection strategy for the interior zone. The space was partitioned into vertical surface segments, which were recorded sequentially from static viewpoints, moving from the bottom to the top. To ensure robust reconstruction, adjacent segments are recorded with overlap.

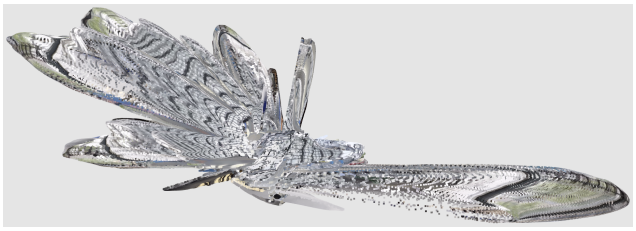


Figure 1. **Misalignment of interior point cloud reconstruction** due to varying depths along the entrance hall.

4.2. Transition - Inside vs. Outside

The Transition from outside to inside data introduces another challenge. In addition to the difficult scene characteristics caused by large windows and reflective walls, there is a difference between the outside and inside data distribution which is for example visible in illumination. As a result, a high distribution shift can cause the predicted point cloud to split into two entrances with a translational offset: one aligned with outside views and another with inside views as in Fig. 3. We attribute this discrepancy to VGGT's limited generalization to our dataset, particularly for outside-to-inside transitions. To mitigate this issue, we reduce the video distance (i.e. the displacement within a single video) while simultaneously increasing the data den-

Figure 2. **Example images from our dataset.** Top left: exterior view of the Mathematics and Informatics building. Top right: main entrance with glass facade. Bottom left: entrance hall captured from the entry zone. Bottom right: reflective surface of the lecture hall.

Figure 3. **Point clouds of transition scenes.** Left: Example showing distribution differences between inside and outside data, leading to a translational offset. Right: Improved reconstruction after reducing video distance and increasing data density in transition scenes.

sity for transition scenes. This reduces the distribution gap within batches, simplifying the point cloud reconstruction task and effectively decreases the translational offset.

4.3. Extended Large-scale Reconstruction

To test the limits of the local alignment, we extended the scene to encompass the entire building. This experiment revealed that the errors in the local alignments accumulate. As a result, the global point cloud alignment becomes increasingly distorted over distance, as further illustrated in Appendix A. These findings motivate the use of global optimization methods with loop closure to mitigate such drift.

4.4. Scale Estimation

Although MAST3R was trained for metric predictions, the complexity of the MI building may exceed its training domain. To evaluate its metric accuracy under simpler conditions, we additionally tested a door within the building. The same door was cropped from multiple frames to generate several batches with varying crops. Results for these batches are provided in Appendix B.

4.5. Parametrization for Dense reconstruction

NKSR. To investigate the effect of the *voxel_size* parameter, we add a tunable parameter α_{scale} that scales the *voxel_size* calculated using the estimated standard deviation. Intuitively, *voxel_size* should correlate inversely with the amount of detail in the reconstruction. However, an optimal *voxel_size* needs to be chosen to balance detail and noise. To qualitatively evaluate the optimality of setting $voxel\_size = std(\epsilon_{pcd})$, we conducted experiments by varying α_{scale} (see Appendix C).

CityGaussian. *Block Partitioning.* We partition the scene on the ground plane into non-uniform blocks to balance load across heterogeneous content. Concretely, we use a coarse tiling adapted to content density: a finer 3×3 tiling for indoor floors where geometry is dense, and a 2×2 tiling for outdoor areas to cover larger, sparser extents.

View Assignment Threshold. For assigning training views to blocks, we adopt an SSIM-based threshold ε (computed over projected patches) and a coverage prior: a view is assigned to block b if either $\mathrm{SSIM}(b, \mathrm{view}) > \varepsilon$ or its ray coverage within b exceeds 5%. We found

$$\varepsilon_{\text{indoor}} = 0.12, \qquad \varepsilon_{\text{outdoor}} = 0.085$$

to work well.

Training Schedule. Each block is initialized from a global coarse prior and fine-tuned for N_{ft} steps. For building-scale scenes we use

$$N_{\text{ft}}^{\text{indoor}} = 6 \times 24{,}000, \qquad N_{\text{ft}}^{\text{outdoor}} = 4 \times 30{,}000$$

with early stopping if the validation PSNR plateaus for 3×10^4 steps.

Densification, Elongation Control, and Depth Guidance. We employ decomposed-gradient densification on photometric residuals; a Gaussian g_k is split/inserted if

$$\left\|\nabla_{\boldsymbol{\mu}_k} \mathcal{L}_{\text{phot}}\right\|_2 > \tau_\mu \quad \text{or} \quad \left\|\nabla_{\boldsymbol{\Sigma}_k} \mathcal{L}_{\text{phot}}\right\|_F > \tau_\Sigma,$$

with (τ_μ, τ_Σ) annealed by block iteration. To suppress degenerate elongated splats, we cap the covariance eigenvalue ratio $\kappa_k = \lambda_{\max}/\lambda_{\min}$ with

$$\kappa_{\max}^{\text{indoor}} = 8, \qquad \kappa_{\max}^{\text{outdoor}} = 12.$$

Optimizer and Learning Rates. We decouple learning rates for appearance and geometry; in dense phases:

$$\eta_{\text{color}} = 2.0 \times 10^{-3}, \qquad \eta_{\text{geometry}} = 4.0 \times 10^{-4}.$$

5. Results

Our proposed method reconstructs both the interior and exterior regions of the TUM Mathematics and Informatics building with geometrical accuracy, despite the complexity of the building (Fig. 4). The results demonstrate robust local alignment across batches.

Figure 4. **Comparison of reconstruction methods.** For three representative views of the Mathematics and Informatics building (left column), the corresponding visualizations are shown: point cloud, NKSR mesh reconstruction, and CityGaussian rendering.

5.1. Completeness

The reconstructed scene captures the primary structural components, including the entrance, facade, and interior hall, while preserving their geometric accuracy. Nevertheless, certain regions remain incomplete due to insufficient observations in the dataset. In particular, the back of the red entrance box (Fig. 4 Top), the ceiling of the lecture hall, and the rear balcony sections exhibit missing or distorted geometry. For example, the ceiling of the lecture hall drifts toward the background (Fig. 5), an artifact that originates not from the reconstruction pipeline itself but from the absence of viewpoints covering the upper part of the hall. These results indicate that the limitations in completeness are primarily determined by the coverage of the dataset rather than the reconstruction pipeline.

Figure 5. **Lecture hall - inside view.** The reconstructed ceiling drifts toward the background due to missing upper-view observations.

5.2. Transition

Another evaluation criterion is the Transition between the outside scene and the inside of the building, as shown in Fig. 6. The transition region may appear to exhibit a translational offset. However, this assumption is incorrect for two reasons. First, the alignment of the red box is correct, as the gap occurs due to point removal for visualization reasons explained in Sec. 5.3.

Second, two window facades are visible - One corresponding to the exterior and one to the interior. The real distance between the edge of the red box and the window facade is noticeably larger for the exterior than for the interior. Therefore, the facade of the exterior is modeled more accurately. Generally speaking, the displacements are difficult to reconstruct for VGGT, as depth estimation is particularly challenging for transparent regions. This also explains the absence of displacement on the interior side. The actual displacement between the red box edge and the interior window facade is too small for VGGT to estimate reliably, resulting in no reconstructed displacement.

Furthermore, it is important to emphasize that the exterior window facade is reconstructed more consistently and accurately than the interior one. We attribute this to the relative ease of depth estimation when transitioning from the outside view to the inside view. Two factors contribute to this: (i) when viewed from the outside, the darker interior makes the windows appear less transparent in the images, thereby

Figure 6. **Transition: Outside - Inside** Visualization of the entrance region showing aligned interior and exterior point clouds.

simplifying depth estimation for VGGT and (ii) viewing inside reveals a finite and well-defined depth, while looking outwards corresponds to much larger depths or even infinite depth in the case of the sky.
This phenomenon is also reflected in the reconstructed interior window facade. The transparent surfaces exhibit a wavy, inconsistent depth profile, whereas the metal bars, which represent structures with a well-defined finite depth, remain consistent and serve to stabilize the reconstruction.

5.3. Visualization and Transition Filtering

For improved visualization, particularly with respect to computational feasibility and noise reduction, we apply filtering using a dedicated confidence threshold τ_{visu}. This threshold is chosen independently of the confidence threshold τ_{align} used for local alignment.
Since the batches focus on different regions, the optimal confidence threshold for noise suppression varies, particularly when comparing exterior and interior reconstructions with transition reconstructions. Transition videos introduce substantial noise, primarily as a result of the moving-camera data collection process. To generate a noise-free point cloud, we include the functionality to filter batch-wise for a confidence threshold $\tau_{visu,i}$ and improve the visualization by removing non-confident points. In our setup, we apply a confidence threshold of 1.0 for the transition scenes, which effectively removes the point clouds entirely. While this removes the heavy noise and thereby visually refines the point cloud, it also eliminates the transition data.
Remarkably, even though the transition scenes contain considerable noise, the overall alignment across the batches remains robust. This demonstrates the stability of the reconstruction pipeline despite challenging input conditions.

5.4. Scaling

As no ground truth data for the MI building exists, we exploit the fact that the parabola slides descend from a height of 13 meters. Based on this, we approximate the building's height to be 17 meters on the lower (front) side.
Despite using variance-based confidence estimation for the scale, the reconstructions produced by our pipeline are generally only a few meters high. This discrepancy is likely due to high-variance predictions and the limited generalization capability of MAST3R.

5.5. CityGaussian: Visualization & Metrics

To quantitatively evaluate the quality of our CityGaussian-based reconstruction on the MI Building dataset, we report three widely used image-level metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS); see Table 1. For qualitative assessment, we visualize both an exterior facade and an interior corridor reconstructed by our method. For more detail, we refer to the renderings in Appendix D, Fig. 11.

Method	**PSNR** ↑	**SSIM** ↑	**LPIPS** ↓
Mean(indoor)	21.8791	0.7263	0.3416
Mean(outdoor)	26.3668	0.8550	0.1838

Table 1. **Quantitative results of our method** averaged over Indoor/Outdoor Test Images.

6. Conclusion

In this work, we presented a metric 3D reconstruction pipeline for large-scale environments, demonstrated on the challenging TUM MI building. Our approach combines transformer-based point cloud prediction, local alignment techniques for large-scale processing, scale estimation, and dense reconstruction to address reflective surfaces, outdoor-indoor transitions, and complex interior structures. The results highlight both the potential and current limitations of transformer-based 3D reconstruction in real-world settings.
Looking ahead, several improvements remain open. While our local alignment strategy ensures consistency across overlapping batches, a global alignment procedure could further improve stability and enable error correction across multiple batches. Incorporating whole-view images within each batch may also support global consistency. Moreover, scaling remains a major challenge: although MAST3R provides metric cues, its performance degrades in complex environments. A dedicated pipeline for reliable scale estimation, potentially leveraging architectural priors such as door frames, could significantly enhance metric accuracy.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE*

12th International Conference on Computer Vision, pages 72–79, 2009. 2

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. 2

[3] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6): 1874–1890, 2021. 1

[4] Zerui Chen, Rolandos Alexandros Potamias, Shizhe Chen, and Cordelia Schmid. Hort: Monocular hand-held objects reconstruction with transformers, 2025. 2

[5] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it – pushing vggt's limits on kilometer-scale long rgb sequences, 2025. 2

[6] Yu Feng, Xing Shi, Mengli Cheng, and Yun Xiong. Diff-point: Single and multi-view point cloud reconstruction with vit based diffusion model, 2024. 2

[7] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building rome on a cloudless day. In *Computer Vision – ECCV 2010*, pages 368–381, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 1

[8] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 1

[9] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation, 2019. 2

[10] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 1

[11] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024. 2

[12] Jiahui Huang, Kangxue Mei, Yiqun Wang, Yinda Chen, Andrea Tagliasacchi, Yaran Zhang, and Zhen Huang. Neural kernel surface reconstruction. In *CVPR*, 2023. CVPR 2023 Highlight. 2, 4

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4), 2023. 2

[14] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 2

[15] Shaohui Liu, Yidan Gao, Tianyi Zhang, Rémi Pautrat, Johannes L. Schönberger, Viktor Larsson, and Marc Pollefeys. Robust incremental structure-from-motion with hybrid features, 2024. 1

[16] Yang Liu, He Guan, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *ECCV*, 2024. 2

[17] Yang Liu, He Guan, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes, 2024. 2

[18] Raul Mur-Artal and Juan D. Tardos. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[19] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[20] Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. *arXiv preprint*, 2024. 2

[21] J.L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, pages 314–317 vol.3, 2000. 3

[22] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[23] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[24] Chitturi Sidhartha, Lalit Manam, and Venu Madhav Govindu. Adaptive annealing for robust geometric estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21929–21939, 2023. 3

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2

[26] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21686–21697, 2024. 2

[27] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2

[28] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass, 2025. 2

[29] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2, 2024. 2

[30] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image, 2023. 2

[31] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. Geomvsnet: Learning multi-view stereo with geometry perception. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21508–21518, 2023. 2

Appendices

A. Extended Large-scale experiment

To further illustrate the limitations of local alignment, we present an extended point cloud reconstruction of the entire building (Fig. 7). While local alignments remain effective over short distances, small errors in rotation, translation, and scale accumulate as the reconstruction spans larger areas. This drift results in noticeable distortions in the global structure of the point cloud.

The top view in Fig. 7 shows a downsampled point cloud reconstruction for visualization purposes with real-appearance coloring. To highlight the accumulated error and make the misalignments between consecutive segments more apparent, the bottom view assigns a distinct color to each batch. These results demonstrate the importance of global optimization techniques with loop closure to mitigate drift and ensure global consistency in large-scale reconstructions.

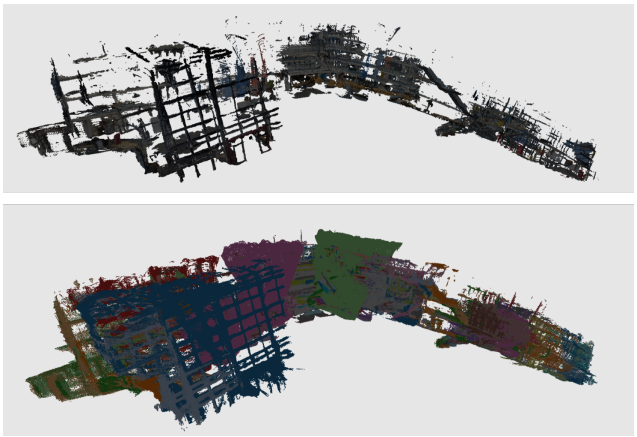


Figure 7. **Point cloud reconstruction of the extended building scene.** The visualization highlights the accumulation of local alignment errors over large scene reconstructions. Top: point cloud rendered with real-appearance coloring. Bottom: point cloud with each batch assigned a distinct color.

B. MAST3R Predictions on a Door

As illustrated in Fig. 8, MAST3R's predictions are not metrically consistent among different crops of the same scene from our dataset. Importantly for our pipeline, this also implies that MAST3R does not always produce reconstructions with reliable metric scales.

Figure 8. **MAST3R Predictions for a Door.** Predictions for different crops of the same door, highlighting variations in predicted scale.

C. Effect of α on NKSR Reconstruction

In order to assess the optimality of the noise based voxel size parameter, we incorporate a scaling factor α that scales the estimated σ_{noise}. Intuitively, voxel size (σ_{noise}) should correlate inversely with the reconstruction resolution. Decreasing α should increase the reconstruction resolution. Fig. 9 demonstrates that this actually is the case. However, experiments show that the α parameter also has an effect on the completeness in the reconstructions.

Specifically, increasing the α is observed to result in larger holes forming in the reconstruction. Even though lowering

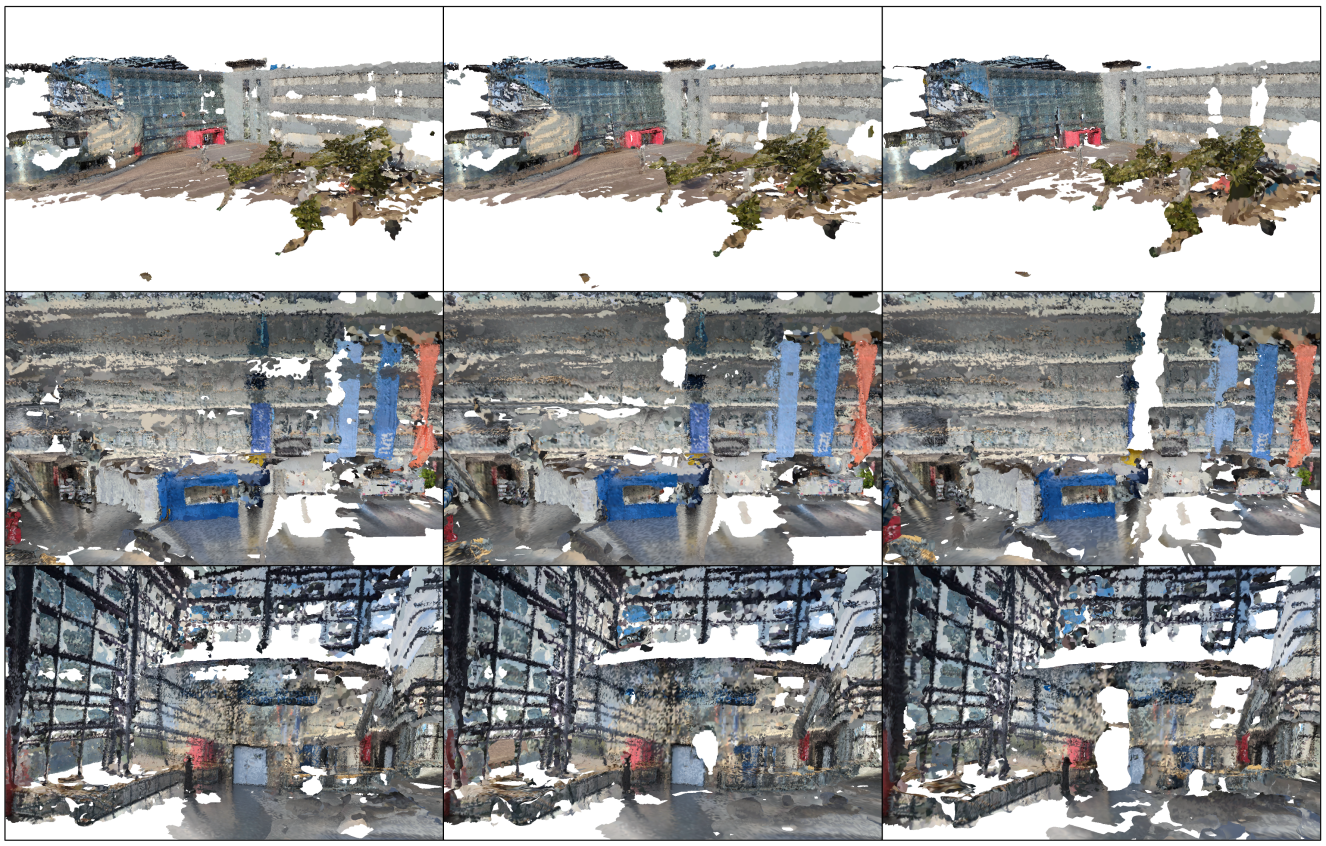


Figure 9. **NKSR results** from three different views with varying α. Left: $\alpha = 0.75$; Middle: $\alpha = 1$; Right: $\alpha = 1.25$.

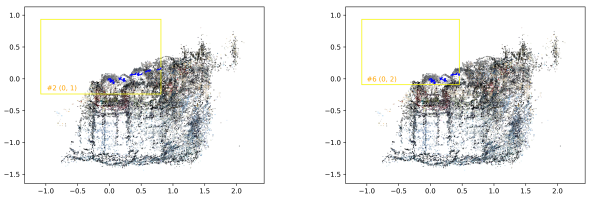
α seems to prevent large holes in high fidelity areas, observations suggest that choosing a too low α might also cause holes in flat regions of the reconstruction.
Finally, it can be concluded that despite different α values resulting in better local reconstructions at different regions, $\alpha = 1$ seem to be a good tradeoff between resolution and noise.

D. Effect of Block Count on Outdoor Effective Points

In outdoor scenes, the choice of block partitioning directly affects how many effective points (i.e., 3D samples with sufficient multi-view support) are available for densification. We define the per-block effective-point count as

$$E_b \;=\; \big|\{\, p \in P \;\big|\; \mathrm{vis}_b(p) \geq m \;\wedge\; p \in \mathrm{Frustum}_b \,\}\big|,$$

where $\mathrm{vis}_b(p)$ counts the number of block-assigned training views in which p is visible (front-facing and within the image bounds), and $m \in \{2, 3\}$ is a minimal multi-view threshold consistent with our view-assignment rule (SSIM/coverage; see main text).
2×2 vs. 3×3. Coarser 2×2 partitions yield larger per-block view pools and wider baselines, typically increasing E_b-



(a) 2×2 partitioning (outdoor) (b) 3×3 partitioning (outdoor)

Figure 10. **Outdoor block partitioning.** Coarser blocks (left) assign more views per block and generally yield higher effective-point counts E_b, improving stability for distant structures. Finer blocks (right) improve locality but may reduce E_b near seams unless compensated by overlap and relaxed view-assignment thresholds.

especially for far-range structures (facades, long streets) where parallax is modest. This improves photometric stability and reduces seam artifacts at block boundaries but raises per-block memory and runtime. Finer 3×3 tilings improve locality and load balance, yet each block receives fewer views and narrower baselines; as a result, E_b can drop for distant geometry and at inter-block seams unless com-

(a) Outdoor facade (CityGaussian). (b) Indoor corridor (CityGaussian).

Figure 11. **Qualitative CityGaussian reconstructions on the MI Building.** Left: outdoor facade; Right: interior corridor.

pensated by overlap or relaxed assignment thresholds.

Implications for CityGaussian. Because our CityGaussian training selects and densifies where multi-view residuals remain high, a reduction in E_b (e.g., with 3×3 outdoors) can postpone or weaken densification in distant regions, and may introduce sparse "holes" near block borders. Practically, for urban outdoor scenes we recommend defaulting to 2×2 (more views per block, stronger cross-view constraints). When memory requires 3×3, we suggest (i) slightly increasing the coverage prior (e.g., from 5% to $8\% \sim 10\%$), (ii) modestly relaxing the SSIM threshold, and (iii) adding a small halo overlap between neighboring blocks to stabilize E_b across seams.

Fig. 11 presents qualitative results for the MI Building: an outdoor facade (left) and an indoor corridor (right). These views complement the quantitative metrics in Tab. 1, showing that our reconstruction preserves large-scale structure outdoors while retaining fine geometric and photometric details indoors.