Surveys on feed-forward 3R methods for high-resolution photogrammetric images via image divide-and-conquer strategy

Zhe Shen, Mengmeng Shu, Guanbo Wang, Yifei Yu*, Zongqian Zhan, Xin Wang

School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
(shenzhe, mengmengshu, wangguanbo, yfyu2020)@whu.edu.cn, (zqzhan, xwang)@sgg.whu.edu.cn

Keywords: 3R methods, 3D Reconstruction, High Resolution Image, Divide-and-Conquer

Abstract

Recently, data-driven feed-forward 3D reconstruction methods, such as DUSt3R, MASt3R, Fast3R and VGGT, have gained widespread attention due to their superior end-to-end processing capabilities across various geometric 3D vision tasks. However, heavy reliance on GPU hardwares limits the applicability of these 3R methods to only single-image pairs or small-scale datasets, making them challenging to handle large-scale high-resolution photogrammetric images. In this work, we conduct a survey on these 3R methods and employ a divide-and-conquer framework that divides the entire image dataset into several overlapping sub-blocks, reconstructs each sub-block separately using 3R methods, and then merges them per 3D similarity transformations. Experimental results demonstrate that our method effectively expands the number of images that the aforementioned feed-forward 3R methods can handle. Furthermore, a comprehensive experiment on photogrammetric data is carried out by comparing the processing time, GPU memory usage, and accuracy to explore the possibility of applying these novel feed-forward 3R methods to high-resolution photogrammetric datasets. Project web: https://sh1nzzz.github.io/3R-methods-via-divide-and-conquer-strategy.github.io/.

1. Introduction

Nowadays, learning-based and data-driven methods have achieved considerable results in the field of 3D reconstruction and have gradually become a research hotspot. Traditional geometry-based photogrammetry methods typically involve several very complex processes, dividing the entire 3D reconstruction task into many subtasks, including feature extraction and matching (Lowe, 2004; Wang et al., 2024a; Hou et al., 2023), image relative orientation (Nistér, 2004), triangulation (Hartley and Zisserman, 2003), reconstruction of sparse scenes (Snavely et al., 2006; Schonberger and Frahm, 2016), dense matching (Bleyer et al., 2011; Yao et al., 2018), and so on. However, this step-by-step approach can lead to some limitations, such as error accumulation between processing that severely impacts the accuracy of the results and collapse in challenging cases with sparse viewpoints (Long et al., 2022) and weak texture (Yang and Jiang, 2021; Wang et al., 2023).

In contrast, recent data-driven feed-forward 3D reconstruction methods, such as DUSt3R (Wang et al., 2024b), MASt3R (Leroy et al., 2024), Fast3R (Yang et al., 2025), and VGGT (Wang et al., 2025), have emerged to show fair performance in estimating camera poses and dense point clouds. These methods leverage end-to-end learning pipelines to effectively overcome the limitations of traditional approaches, carrying out robust and high-quality 3D reconstruction. However, according to our tests on a computer with an RTX 3080 (10GB VRAM), DUSt3R can handle up to 14 images at most, while MASt3R can process about 40 images at most. The Fig. 1 shows their GPU memory and time consumption. It can be observed that once the GPU memory is exceeded and computation switches to the CPU, the inference speed becomes extremely slow. Recent works such as Fast3R and VGGT have significantly improved runtime efficiency and, to some extent, increased the number of images that can be processed. But on the whole,

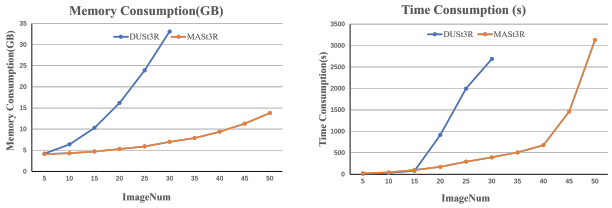

Figure 1. GPU memory and time consumption of DUSt3R and Mast3R.

current feed-forward 3R methods still cannot carry out reconstruction on personal devices (a single RTX 4090 GPU) when the number of images is relatively large.

To address this limitation and further explore these feed-forward methods, we make two key contributions in this paper: first, we propose an image divide-and-conquer framework that divides the entire scene into several overlapping sub-blocks using the ”clustering block expansion method”. The sub-blocks are first reconstructed using the aforementioned data-driven methods, and then the sub-blocks are merged into a complete model using ”3D similarity transformation” based on their overlapping relationships. Based on this, high-resolution UAV images can be successfully processed by feed-forward 3R methods using limited computing resources; second, this work integrates the proposed image divide-and-conquer framework with these feed-forward 3R methods and conducts extensive experiments on various photogrammetric datasets. By comparing their processing time, GPU memory usage, and accuracy, the possibility of applying these SOTA feed-forward 3R methods in the field of photogrammetry is explored.

2. Related Work

3D reconstruction is a long-standing research topic in computer vision and photogrammetry that has been extensively stud-

* Corresponding Author

ied over the past few decades (Hartley and Zisserman, 2003; Stathopoulou and Remondino, 2023). This section reviews some 3D reconstruction methods, which are mainly divided into two parts: the traditional geometry-based 3D reconstruction methods and the feed-forward 3R methods.

2.1 Traditional geometry-based 3D Reconstruction Methods

Traditional geometry-based 3D reconstruction pipelines typically begin with Structure-from-Motion (SfM) to estimate camera poses and recover a sparse point cloud, after which Multi-View Stereo (MVS) is employed to generate a dense reconstruction.

Structure-from-Motion(SfM) (Schonberger and Frahm, 2016; Wang et al., 2019; Zhan et al., 2025) aims to reconstruct a sparse point cloud and estimate camera parameters simultaneously from a set of images. Traditional SfM pipeline typically consists of several key steps: image feature extraction and matching, pose estimation, triangulation, and bundle adjustment. Recently, many learning-based methods have been integrated into the SfM pipeline to improve specific steps, achieving remarkable progress—particularly in feature extraction (Dusmanu et al., 2019; Gleize et al., 2023) and matching (Chen et al., 2021, 2022). Despite these improvements, the sequential structure of the SfM pipeline remains unchanged, which leads to the accumulation of errors between subtasks as well as the decline of overall robustness.

Mutli-view Stereo(MVS) refers to the task of performing dense 3D reconstruction from multiple overlapping images with known camera parameters. MVS algorithms are typically categorized into conventional methods based on semi-global matching (SGM) (Poggi and Mattoccia, 2016; Rothermel et al., 2012) and learning-based approaches (Kang et al., 2019, 2024). Since MVS is not the primary focus of this study, more detailed information can be found in the following references (Stathopoulou and Remondino, 2023; Wang et al., 2024a). However, most of these methods rely heavily on accurate camera parameters, and their performance can degrade significantly if the input poses are imprecise. In contrast, recent feed-forward 3D reconstruction methods effectively address these limitations in both MVS and SfM.

2.2 Data-driven Feed-forward 3R Method

Recently, with the growing volume of data and advances in computational power, a variety of data-driven 3D reconstruction methods have emerged. DUSt3R is the first fully data-driven method for multi-view geometry (Wang et al., 2024b). It adopts an end-to-end framework that formulates pairwise reconstruction as a regression problem over pointmaps, and further proposes a complete 3D reconstruction pipeline for unconstrained images. Unlike traditional SfM-MVS pipelines, DUSt3R directly infers pointmaps from input images without relying on any prior camera information, marking a significant paradigm shift in 3D reconstruction. However, it is worth noting that when the number of input images exceeds two, DUSt3R requires an expensive global alignment process to obtain a complete reconstruction (both in time and memory consumption).

In the past year, many subsequent works based on DUSt3R have appeared, such as MASt3R (Leroy et al., 2024), which obtains better matching results via a 3D perceptual matching method based on the DUSt3R framework, significantly improving the accuracy of its pose estimation. Also, MASt3R provides a sparse global alignment method, which reduces its GPU memory consumption, allowing the processing of a larger number of images. However, similar to DUSt3R, MASt3R still relies on pairwise inference followed by global alignment, which limits its ability to leverage the interaction information across all input images. As a result, it suffers from slower processing speed and substantial memory consumption during the global alignment phase.

More recently, Fast3R (Yang et al., 2025) and VGGT (Wang et al., 2025) have made improvements in network architectures, enabling direct multi-view inference and reconstruction without additional global alignment optimization, which significantly reduces memory and time consumption. Nevertheless, when the number of input images becomes relatively large, current feed-forward 3R methods still fail to perform reconstruction on personal devices (e.g., a single NVIDIA RTX 4090 GPU). In this paper, we propose a framework based on image divide-and-conquer to solve this problem.

3. Preliminary

In this section, a brief explanation of the four surveyed feed-forward methods (DUSt3R, MASt3R, Fast3R, and VGGT) is provided.

3.1 DUSt3R

Network architecture. The architecture of DUSt3R, inspired by CroCo (Weinzaepfel et al., 2022), consists of two identical branches (one for each input image), each comprising an image encoder, a decoder, and a regression head. As shown in Fig. 3, the two images are encoded using a shared-weight Vision Transformer (ViT) encoder (Dosovitskiy et al., 2020) to produce features *F1* and *F2*, which are then fused through self-attention and cross-attention mechanisms in the decoder. The aggregated features are passed to a regression head that outputs a pointmap and a confidence map for each image.

Loss function. DUSt3R employs the Euclidean distance between predicted and ground-truth pointmaps as its regression loss. For an image pair $v \in \{1, 2\}$, let $\bar{X}^{1,1}$ and $\bar{X}^{2,1}$ denote the ground-truth pointmaps and D^1 , D^2 represent the sets of valid pixels. The regression loss for the *i*-th pixel in image *v* is defined as Eq. (1):

$$\ell_{\text{regr}}(v, i) = \left\| \frac{1}{z} X_i^{v,1} - \frac{1}{\bar{z}} \bar{X}_i^{v,1} \right\|. \tag{1}$$

Where $z = \text{norm}(X^{1,1}, X^{2,1})$ and $\bar{z} = \text{norm}(X^{1,1}, X^{2,1})$ are scaling factors introduced to resolve scale ambiguity, computed as the average distance of all valid 3D points, as shown in Eq. (2):

$$\text{norm}(X^1, X^2) = \frac{1}{|\mathcal{D}^1| + |\mathcal{D}^2|} \sum_{v \in \{1,2\}} \sum_{i \in \mathcal{D}^v} \|X_i^v\|. \tag{2}$$

Additionally, DUSt3R takes the predicted confidence of each pointmap into account, which is used to reweight the regression loss. The final training objective is the confidence-weighted regression loss, as shown in Eq. (3):

$$\mathcal{L}_{\text{conf}} = \sum_{v \in \{1,2\}} \sum_{i \in \mathcal{D}^v} C_i^{v,1} \ell_{\text{regr}}(v, i) - \alpha \log C_i^{v,1} \tag{3}$$

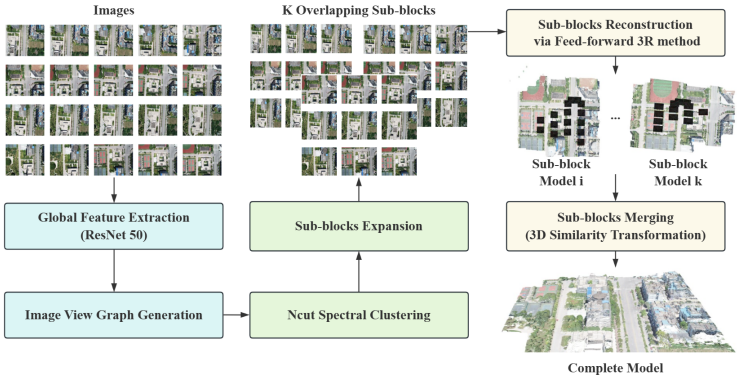

Figure 2. Pipeline of the proposed image divide-and-conquer framework.

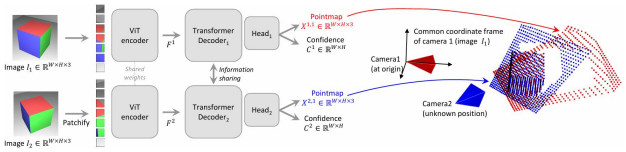

Figure 3. Network Architecture of DUSt3R.

Where $C_i^{v,1}$ is the confidence score for pixel i, and α is a hyper-parameter controlling the regularization term (Wan et al., 2018).

Training datasets. DUSt3R trains the network using a mixture of 8 datasets: Habitat, MegaDepth, ARKitScenes, MegaDepth, Static Scenes 3D, Blended MVS, ScanNet++, CO3D-v2 and Waymo. These datasets feature diverse scenes types: indoor, outdoor, synthetic, real-world, object-centric, etc. DUSt3R totally extracts 8.5M pairs to tarin. For details, please refer to the paper of DUSt3R (Wang et al., 2024b).

3.2 MASt3R

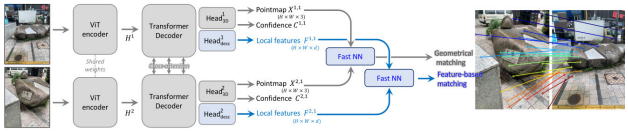

Figure 4. Network Architecture of MASt3R.

Network architecture. As shown in Fig. 4, based on the DUSt3R network, MASt3R introduces an additional head for each branch to generate local features. These features are then combined with the 3D pointmaps and fed into a fast nearest-neighbor (NN) matcher to generate robust correspondences. As a result, compared to DUSt3R, MASt3R achieves higher accuracy in both matching and camera pose estimation. However, the enhanced performance is accompanied by a longer inference time.

Loss function. The loss function of MASt3R is also modified on the basis of DUSt3R. To address use-cases where scale invariance is undesirable (such as map-free visual localization), MASt3R modifies the normalization factors z and $\bar{z}$ used in DUSt3R. Specifically, when ground-truth pointmaps are given in metric scale, MASt3R sets $z := \bar{z}$, thus avoiding normalization of the predicted pointmaps. As a result, the regression loss $\ell_{\text{regr}}(v,i)$ becomes the form shown in Eq. (4), while the remaining components of the final confidence-aware regression loss remain consistent with DUSt3R, as defined in Eq. (3):

$$\ell_{regr}(v,i) = \frac{\|X_i^{v,1} - \bar{X}_i^{v,1}\|}{\bar{z}} \tag{4}$$

Moreover, to achieve high-precision image matching, MASt3R introduces a matching head that produces two dense feature maps $D^1, D^2 \in \mathbb{R}^{H\times W\times d}$ of dimensionality d, and applies the InfoNCE (Oord et al., 2018) loss over the set of ground-truth correspondences $\hat{\mathcal{M}} = \{(i,j)|\hat{X}_i^{1,1} = \hat{X}_j^{2,1}\}$, as shown in Eq. (5):

$$\mathcal{L}_{\text{match}} = -\sum_{(i,j)\in\hat{\mathcal{M}}} \log \frac{s_\tau(i,j)}{\sum_{k\in\mathcal{P}^1} s_\tau(k,j)} + \log \frac{s_\tau(i,j)}{\sum_{k\in\mathcal{P}^2} s_\tau(i,k)}$$
$$\text{with} s_\tau(i,j) = \exp\left[-\tau D_i^{1\top} D_j^2\right] \tag{5}$$

Here, $\mathcal{P}^1 = \{i|(i,j)\in\hat{\mathcal{M}}\}$ and $\mathcal{P}^2 = \{j|(i,j)\in\hat{\mathcal{M}}\}$ denote the subset of considered pixels in each image and τ is a temperature hyper-parameter. Since MASt3R wants to encourage each pixel to match at most one pixel in the other image as the same 3D point in the scene, the network is rewarded only when the the correct pixel is matched, rather than a nearby pixel. Finally, the final training objective is formulated as a sum of the revised regression loss and the matching loss, as shown in Eq. (6):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{conf}} + \beta\mathcal{L}_{\text{match}} \tag{6}$$

Training datasets. MASt3R trains the network with a mixture of 14 datasets. In addition to the dataset used by DUSt3R, it has expanded the Mapfree, WildRgb, VirtualKitti, Unreal4K, TartanAir and an internal dataset. For details, please refer to the paper of MASt3R (Leroy et al., 2024).

3.3 Fast3R

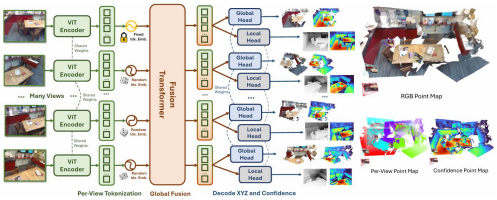

Figure 5. Network Architecture of Fast3R.

Network architecture. To overcome the limitation that DUSt3R and MASt3R can only process a single pair of images in a single pass, Fast3R, inspired by DUSt3R, introduces a novel network architecture that includes three components: image encoding, fusion transformer, and pointmap decoding, as illustrated in Fig. 5. Similar to DUSt3R, Fast3R also employs a CroCo ViT (Weinzaepfel et al., 2022) as image feature encoder. After that, positional embeddings are added with one-

dimensional image index embeddings. These index embeddings enable the fusion transformer to identify which patches come from the same image and to distinguish the first image, which is used to define the global coordinate frame. Subsequently, these concatenated encoded patches from all views are fed into a fusion transformer that performs all-to-all self-attention, providing Fast3R with full context from all views. Finally, two separate DPT-L (Ranftl et al., 2021) decoder heads are used to predict both local and global pointmaps $(\mathbf{X}_L, \mathbf{X}_G)$ and their corresponding confidence maps $(\mathbf{\Sigma}_L, \mathbf{\Sigma}_G)$.

Loss function. The total loss function of Fast3R is the combination of pointmap loss for the local and global pointmaps, as shown in Eq. (7):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{X}_\text{G}} + \mathcal{L}_{\text{X}_\text{L}} \tag{7}$$

where $\mathcal{L}_{\text{X}_\text{G}}$ and $\mathcal{L}_{\text{X}_\text{L}}$ are confidence-weighted versions of the normalized 3D pointwise regression loss, which is similar to that of DUSt3R, as shown in Eq. (8):

$$\mathcal{L}_{\mathbf{X}}(\hat{\mathbf{\Sigma}}, \hat{\mathbf{X}}, \mathbf{X}) = \frac{1}{|\mathbf{X}|} \sum \hat{\mathbf{\Sigma}}_+ \cdot \ell_{\text{regr}}(\hat{\mathbf{X}}, \mathbf{X}) + \alpha \log(\hat{\mathbf{\Sigma}}_+) \tag{8}$$

Since the log term requires the confidence scores to be strictly positive, here set $\hat{\Sigma}_+ = 1 + \exp(\hat{\Sigma})$.

Training datasets. Fast3R trains the network with a mixture of only 6 datasets. These datasets are subsets of the training datasets in DUSt3R and do not include CO3D-v2 and Waymo. For details, please refer to the paper of Fast3R (Yang et al., 2025).

3.4 VGGT

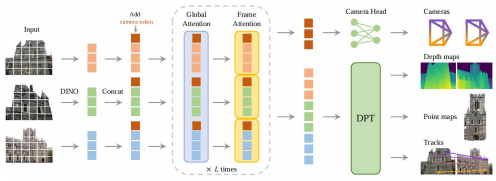

Figure 6. Network Architecture of VGGT.

Network architecture. VGGT, close to Fast3R, is capable of inferring multiple images in a single pass. Compared with the other three methods, VGGT can simultaneously infer multiple geometric results, including camera poses, depth maps, pointmaps, and keypoint tracks. As shown in Fig. 6, VGGT first tokenizes the input images via DINO and then camera tokens are added for camera pose prediction. Subsequently, the tokens from all images are combined and processed using alternating frame-wise and global self-attention, allowing the transformer to alternately attend to local image features and global context. Finally, a camera head is used to predict the camera parameters, while a DPT(Dense Prediction Transformer) head (Ranftl et al., 2021) produces the depth maps, pointmaps, and keypoint tracks.

Loss function. To jointly account for the four different outputs (cameras, depth maps, pointmaps, tracks), VGGT is trained using a multi-tasks loss, as shown in Eq. (9):

$$\mathcal{L} = \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}} + \lambda \mathcal{L}_{\text{track}} \tag{9}$$

The camera loss $\mathcal{L}_{\text{camera}}$ measures the discrepancy between the predicted and ground-truth camera parameters using the Huber loss function. The depth loss $\mathcal{L}_{\text{depth}}$ adopts an aleatoric uncertainty-weighted formulation, where the difference between the predicted and ground-truth depth maps is weighted by the predicted uncertainty map, and further enhanced by incorporating depth gradient information. The pointmap loss $\mathcal{L}_{\text{pmap}}$ follows a similar design to the depth loss but applies the uncertainty weighting to the point map instead. Lastly, the tracking loss $\mathcal{L}_{\text{track}}$ quantifies the difference between the predicted query points and their corresponding ground-truth locations. Please refer to the original paper (Wang et al., 2025) for more details.

Training datasets. VGGT trains the network with a mixture of 17 datasets. These datasets use 6 from the training dataset of MASt3R and extend 11 more: DL3DV, Kubric, ScanNet, HyperSim, Mapillary, Replica, MVS-Synth, PointOdyssey, Aria Synthetic, Aria Digital Twin, Objaverse-like. The combination of its datasets is broadly comparable to those of MASt3R in size and diversity. For details, please refer to the paper of VGGT (Wang et al., 2025).

4. Methodology

To increase the number of photogrammetric images that feed-forward methods are capable of handling, we propose an image divide-and-conquer framework, as illustrated in Fig. 2. The pipeline primarily consists of the following steps: image view-graph generation, sub-blocks division, sub-blocks reconstruction via feed-forward 3R methods and sub-blocks merging.

4.1 Image View-Graph Generation

Take all images of the given scene as input, a pre-trained ResNet 50 model is employed to extract the global features f_i for each image. Subsequently, a weighted image view-graph $G = (V, E, W)$ is generated based on these extracted features. In this view-graph, vertices V represent all input images, and edges $e_{ij} \in E$ indicate the matching relationships between image pairs $\{v_i, v_j\} \in V$. W represents the set of edge weights w, with larger weights corresponding to higher similarity between image pairs. Specifically, the weights are computed using a Gaussian kernel function, as shown in Eq. (10), where the parameter γ is empirical and in this paper it is set to 0.00001.

$$w_{e_{ij}} = \exp\left(-\gamma * \|f_i - f_j\|_2^2\right) \tag{10}$$

4.2 Sub-blocks Division

Based on the weighted image view-graph $G = (V, E, W)$, the graph can be partitioned into k unconnected subgraphs, ensuring that these subgraphs $G_1, G_2, ..., G_k$ satisfy $G_i \cap G_j = \emptyset$ and $G_1 \cup G_2 \cup ... \cup G_k = G$. To achieve this, Normalized Cut (Shi and Malik, 2000) is employed for spectral clustering, resulting in k mutually exclusive sub-blocks $G^i_{div} = \left(V^i_{div}, E^i_{div}, W^i_{div}\right), i = 1, ..., k$.

To ensure successful merge of these sub-blocks in subsequent steps, a sufficient number of common images between adjacent sub-blocks is essential. Therefore, it is necessary to further expand each sub-block to include additional common images. Based on the connection relationships provided by the global image view-graph $G = (V, E, W)$ before partitioning, the edges that originally connected two adjacent sub-blocks

G^i_{div} and G^j_{div} in the global graph are used to introduce common images into both sub-blocks. The process is iteratively performed until each sub-block contains at least 10 images shared with other sub-blocks. As a result, the expanded sub-blocks are denoted as $G^i_{exp} = \left(V^i_{exp}, E^i_{exp}, W^i_{exp}\right), i = 1, ..., k$.

4.3 Sub-blocks feed-forward Reconstruction

After obtaining the expanded sub-blocks, each sub-block is individually processed using the feed-forward methods (DUSt3R, MASt3R, Fast3R, VGGT), producing the reconstructed dense point clouds and estimated image poses.

4.4 Sub-blocks Merging

Finally, to obtain a complete reconstruction of the scene, all the individually reconstructed sub-blocks need to be merged into a unified model. For two adjacent sub-blocks G^i_{exp}, G^j_{exp} with common images, the coordinates of these common images in their respective coordinate frames are denoted as (X^i, Y^i, Z^i) and (X^j, Y^j, Z^j). When the number of common images between two sub-blocks reaches 3, a 3D similarity transformation algorithm (Wang et al., 2019) is employed to compute the transformation matrix between their coordinate systems based on these common images (X^i, Y^i, Z^i), (X^j, Y^j, Z^j). The transformation matrix $B = \lambda(R|T)$ is composed of a scaling factor λ, a 3*3 rotation matrix R, and a 3*1 translation vector T.

To ensure the robustness of the results, a gold standard outlier rejection method, RANSAC, is used to estimate the transformation matrix. In each iteration, three pairs of corresponding coordinates are randomly selected to compute a candidate transformation. The remaining correspondences are then used to evaluate the transformation error. The number of RANSAC iterations is set to 500, and the transformation matrix B that yields the lowest error is selected as the optimal solution.

In order to transform all sub-blocks into a unified coordinate system, a weighted sub-block connection graph is constructed based on their overlapping relationships, where the weight of each edge is defined as the negative number of common images. A minimum spanning tree (MST) is then generated from this graph, with the coordinate system of its root node chosen as the reference. Subsequently, each sub-block is transformed into this reference system along the MST paths. Finally, the point clouds of all sub-blocks, now aligned in the same reference coordinate system, can be merged together to produce a complete 3D dense point cloud of the entire scene.

5. Experiments

To verify the effectiveness of the proposed framework and explore the possibility of applying these novel feed-forward 3R methods to high-resolution photogrammetric datasets, four main experiments are run in this section. The first experiment applies the four feed-forward 3R methods to a single sub-block and compares the results with referenced point clouds to evaluate their performance. The second experiment evaluates the feasibility of the proposed divide-and-conquer framework by comparing the Colmap results reconstructed from partitioned sub-blocks with those obtained from the entire dateset. The third experiment studies the influence of different numbers of blocks in partition to identify an optimal strategy within the framework. Finally, we apply the optimal strategy to four 3R methods, aiming to analyze the strengths and limitations of four 3R methods in the field of photogrammetric reconstruction.

5.1 Datasets

As Tab. 1 shows, three datasets are tested in our experiment: XHSD, YD, and Caffe. Among them, YD consists of UAV images captured with five cameras, XHSD includes UAV images taken only from a downward perspective, and Caffe comprises close-range images collected using three smartphones. Due to the multi-camera acquisition, the YD and Caffe datasets are disordered. Overall, these three datasets contain a relatively large number of high-resolution images (normally exceeding the processing capacity of 3R methods in a single run) and cover diverse photogrammetric situations, making them representative and well-suited for evaluating both the proposed framework and the 3R methods.

5.2 Implementation Details

For the four feed-forward 3R methods, except for the implementation of DUSt3R, which is optimized for GPU memory usage by ourselves, the other three methods are just from the original open-source version. In addition, to compare with the traditional methods, this paper also runs Colmap, and all experiments in this paper are conducted on a single NVIDIA RTX4090 GPU.

5.3 Performance of 3R methods on single sub-block

To evaluate the performance of these 3R methods within one single inference, a single sub-block from caffee dataset is tested and their results are compared with the ground-truth point clouds from LiDAR, as shown in Fig. 7. It can be found that, for individual sub-block, all four methods generally achieve fair results, and VGGT generates the highest quality reconstruction, followed by DUSt3R and MASt3R, while Fast3R yields the poorest results.

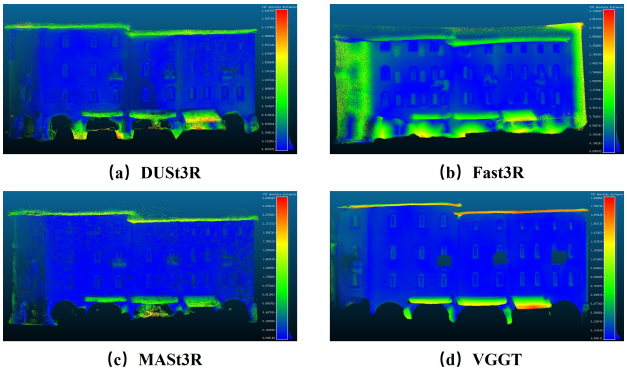

Figure 7. Individual sub-block reconstruction results.

5.4 Feasibility of the image divide-and-conquer framework

In order to validate the feasibility of proposed image divide-and-conquer framework, we divide the dataset XHSD into 15 sub-blocks and perform reconstruction using COLMAP within the proposed framework. The reconstruction results obtained from our proposed framework are compared with those from directly applying COLMAP to the entire scene without partitioning. The results are shown in Fig. 8. Experimental results indicate that the reconstruction obtained from the proposed framework exhibits only a minor difference in accuracy compared to result entirely reconstructed from Colmap, demonstrating the feasibility of proposed image divide-and-conquer framework.

Dataset name	Image number	Ordered/Disordered	Number of cameras	Data type	Image resolution
XHSD	400	Ordered	1	UAV	5472*3648
YD	291	Disordered	5	UAV	6000*4000
Caffe	287	Disordered	3	close-range	1920*1980

Table 1. Information of the experimental dataset

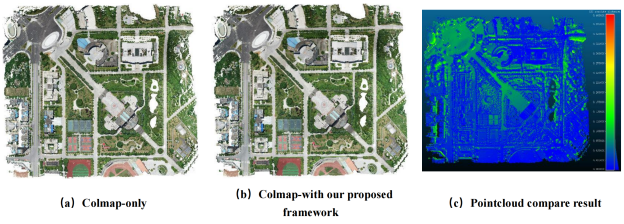

Figure 8. Colmap results with/without our divide-and-conquer strategy.

5.5 Investigation on the impact of different blocks numbers

To explore the impact of different partitioned sub-blocks number, the XHSD dataset is divided into 15, 20, and 25 sub-blocks, respectively. Then they are reconstructed using four 3R methods within the proposed framework in this paper. Taking the reconstruction result of VGGT as an example, as shown in Fig. 9, it can be observed that when the number of sub-blocks increases, the visual reconstruction result shows a higher degree of misalignment. We take the reconstruction result of Colmap as the ground-truth and conduct point cloud comparison with the reconstruction result of VGGT. It can be seen that the reconstruction error of the blocks divided into 15, 20, and 25 sub-blocks gradually increases, which may be attributed to the accumulation of errors during the merging process.

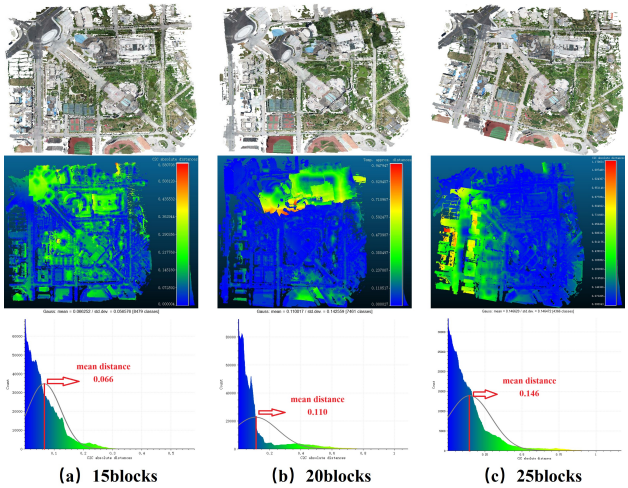

Figure 9. VGGT reconstruction results for different number of blocks.

	DUST3R	MAST3R	FAST3R	VGGT
15blocks	2937.83	6407.41	204.03	276.07
20blocks	2714.01	6009.35	105.69	265.58
25blocks	**2710.41**	**5979.25**	**90.10**	**259.72**

Table 2. Reconstruction Time (s)

	DUST3R	MAST3R	FAST3R	VGGT
15blocks	15.20	6.10	12.00	21.70
20blocks	11.43	5.03	10.66	17.96
25blocks	**8.21**	**5.01**	**10.25**	**16.83**

Table 3. Maximum graphics memory consumption (G)

By comparing the reconstruction time and GPU memory consumption of different partitioning strategies, as shown in the Tab. 2 and Tab. 3, it can be found that for the same reconstruction method, increasing the number of sub-blocks generally leads to a reduction in total reconstruction time, although the difference is relatively small. In terms of GPU memory, a higher number of sub-blocks results in fewer images per sub-block, thereby reducing memory consumption. However, when reconstruction quality is the primary concern, the optimal partitioning strategy under the proposed framework is: the smaller the number of sub-blocks, the better, as long as the GPU memory can support it.

5.6 Evaluation of 3R methods on image partitioning framework

Based on the optimal partitioning strategy in Sec. 5.5, we divided the three datasets XHSD, YD, and Caffe into 15, 12, and 10 sub-blocks, respectively. Using four feed-forward 3R methods, experiments are conducted under the proposed framework, and the reconstruction results of XHSD, YD, and Caffe are shown in Fig. 10. To evaluate the reconstruction results of the four 3R methods, the reconstructions generated by the traditional method COLMAP (well-known for its accuracy and robustness) are selected as the ground-truth for XHSD, YD and Caffe, comparing the 3R method results with the reference point clouds from COLMAP, the 3D spatial accuracy of the four 3R methods is shown in Fig. 11.

The results indicate that among the four 3R methods, VGGT achieves the best performance, followed by MASt3R, while DUSt3R and Fast3R perform poorly and can be considered reconstruction failures. This may be attributed to the fact that merging sub-blocks requires highly accurate image poses. VGGT benefits from incorporating a large amount of global information from all input images and employs a dedicated camera head to estimate high-precision image poses. Similarly, MASt3R leverages an additional head specifically designed to produce accurate pose information. In contrast, DUSt3R and Fast3R lack such components, which likely contributes to their lower performance.

In addition to reconstruction quality, the time efficiency and GPU memory consumption of these four methods are also compared, and the result is shown in the Tab. 4 and Tab. 5. With the results, we can find that all four 3R methods outperform COLMAP in terms of reconstruction speed, especially Fast3R and VGGT, which achieve reconstruction times tens of times shorter than COLMAP. In terms of GPU memory usage,

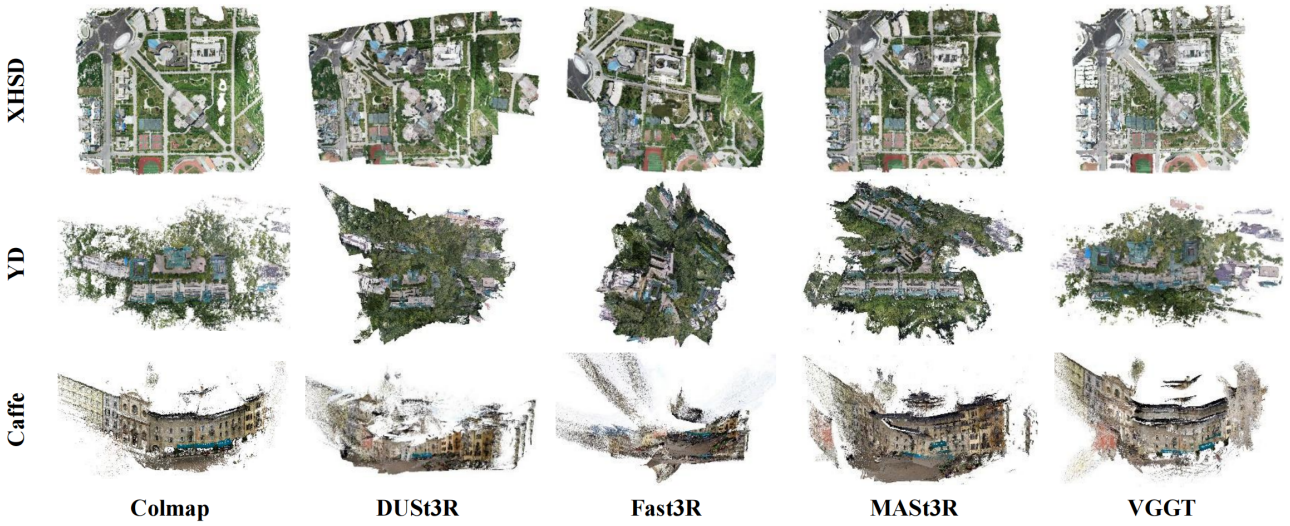

Figure 10. Reconstruction results of XHSD, YD and Caffe.

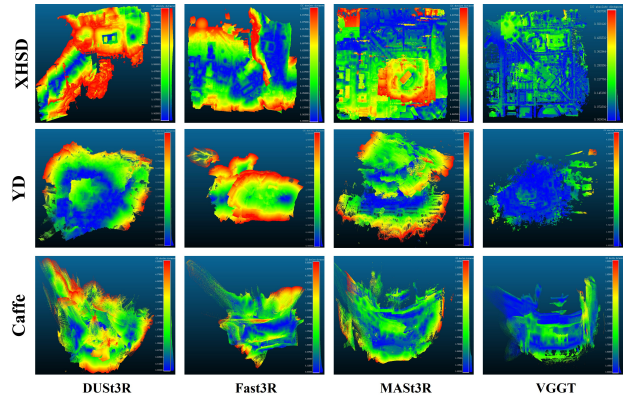

Figure 11. 3D spatial accuracy of the four 3R methods.

COLMAP has relatively low memory consumption, primarily during feature extraction and dense matching. Among the 3R methods, VGGT has the largest memory consumption, followed by DUSt3R and Fast3R, while MAST3R demonstrates the lowest memory usage.

	XHSD	YD	Caffe
COLMAP	14520.00	9720.00	11820.00
DUST3R	2937.83	1898.34	1882.79
MAST3R	6407.41	6120.22	3889.46
FAST3R	**204.03**	**102.32**	82.39
VGGT	276.07	261.46	**73.65**

Table 4. Reconstruction Time (s)

	XHSD	YD	Caffe
COLMAP	**5.80**	5.60	**5.60**
DUST3R	15.20	14.30	15.00
MAST3R	6.10	**5.30**	6.90
FAST3R	12.00	11.10	12.00
VGGT	21.70	19.40	21.70

Table 5. Maximum GPUs memory consumption (G)

Considering overall reconstruction quality, speed, and GPU memory usage, VGGT emerges as the most effective method under the framework proposed in this paper. It achieves the highest reconstruction quality among the four 3R methods, while also maintaining a fast processing speed, ranking in the top tier alongside Fast3R, and performing dozens of times faster than the traditional method COLMAP. Although VGGT has the highest GPU memory consumption, it is capable of efficiently handling dense reconstruction of several hundred high-resolution photogrammetric images on a single RTX 4090 GPU, while still maintaining a reliable reconstruction accuracy and quality under the proposed framework.

6. Conclusion

This paper extensively surveys four SOTA feed-forward 3R methods (DUSt3R, MASt3R, Fast3R, and VGGT) and employs a corresponding divide-and-conquer framework to make these 3R methods feasible on dealing with a large number of high-resolution photogrammetric images. Our comprehensive experimental results demonstrate that, via the proposed framework, among these 3R methods, VGGT can successfully perform fast and dense reconstruction for several hundred high-resolution photogrammetric images on a single RTX 4090 GPU. Although there remains a certain gap in 3D accuracy compared to the traditional COLMAP, the reconstruction speed is several tens of times faster. Therefore, this approach shows promising practical possibility in scenarios where time efficiency is prioritized over the requirement of very high 3D accuracy. Nevertheless, the performance of our divide-and-conquer strategy is heavily reliant on block partitioning and the accuracy of camera pose estimation. In the future, we would like to explore a better block partition as well as a more robust estimator for merging. Also, a fine-tuning solution of 3R method that is tailored for photogrammetric images is of great interest to further investigate.

Acknowledgment

This work was jointly supported by the National Natural Science Foundation of China (42301507) and the Natural Science Foundation of Hubei Province, China (2022CFB727).

References

Bleyer, M., Rhemann, C., Rother, C., 2011. Patchmatch stereo—stereo matching with slanted support windows. *BMVC*, 1–11.

Chen, H., Luo, Z., Zhang, J., Zhou, L., Bai, X., Hu, Z., Tai, C.-L., Quan, L., 2021. Learning to match features with seeded graph matching network. *Proceedings of the IEEE/CVF international conference on computer vision*, 6301–6310.

Chen, H., Luo, Z., Zhou, L., Tian, Y., Zhen, M., Fang, T., Mckinnon, D., Tsin, Y., Quan, L., 2022. Aspanformer: Detector-free image matching with adaptive span transformer. *European Conference on Computer Vision*, Springer, 20–36.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T., 2019. D2-net: A trainable cnn for joint description and detection of local features. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 8092–8101.

Gleize, P., Wang, W., Feiszli, M., 2023. Silk: Simple learned keypoints. *Proceedings of the IEEE/CVF international conference on computer vision*, 22499–22508.

Hartley, R., Zisserman, A., 2003. *Multiple view geometry in computer vision*. Cambridge university press.

Hou, Q., Xia, R., Zhang, J., Feng, Y., Zhan, Z., Wang, X., 2023. Learning visual overlapping image pairs for SfM via CNN fine-tuning with photogrammetric geometry information. *International Journal of Applied Earth Observation and Geoinformation*, 116, 103162.

Kang, J., Chen, L., Deng, F., Heipke, C., 2019. Context pyramidal network for stereo matching regularized by disparity gradients. *ISPRS Journal of Photogrammetry and Remote Sensing*, 157, 201–215.

Kang, J., Chen, L., Heipke, C., 2024. EnhancedNet, an End-to-End Network for Dense Disparity Estimation and its Application to Aerial Images. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 92(5), 531–546.

Leroy, V., Cabon, Y., Revaud, J., 2024. Grounding image matching in 3d with mast3r. *European Conference on Computer Vision*, Springer, 71–91.

Long, X., Lin, C., Wang, P., Komura, T., Wang, W., 2022. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. *European Conference on Computer Vision*, Springer, 210–227.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91–110.

Nistér, D., 2004. An efficient solution to the five-point relative pose problem. *IEEE Transactions on pattern analysis and machine intelligence*, 26(6), 756–770.

Oord, A. v. d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Poggi, M., Mattoccia, S., 2016. Learning a general-purpose confidence measure based on o (1) features and a smarter aggregation strategy for semi global matching. *2016 Fourth international conference on 3D vision (3DV)*, IEEE, 509–518.

Ranftl, R., Bochkovskiy, A., Koltun, V., 2021. Vision transformers for dense prediction. *Proceedings of the IEEE/CVF international conference on computer vision*, 12179–12188.

Rothermel, M., Wenzel, K., Fritsch, D., Haala, N., 2012. Sure: Photogrammetric surface reconstruction from imagery. *Proceedings LC3D Workshop, Berlin*, 8number 2.

Schonberger, J. L., Frahm, J.-M., 2016. Structure-from-motion revisited. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), 888–905.

Snavely, N., Seitz, S. M., Szeliski, R., 2006. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (TOG)*, 25number 3, ACM, 835–846.

Stathopoulou, E. K., Remondino, F., 2023. A survey on conventional and learning-based methods for multi-view stereo. *The Photogrammetric Record*, 38(183), 374–407.

Wan, S., Wu, T.-Y., Wong, W. H., Lee, C.-Y., 2018. Confnet: predict with confidence. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2921–2925.

Wang, F., Zhu, Q., Chang, D., Gao, Q., Han, J., Zhang, T., Hartley, R., Pollefeys, M., 2024a. Learning-based multi-view stereo: a survey. *arXiv preprint arXiv:2408.15235*.

Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D., 2025. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*.

Wang, J., He, T., Zhang, X., Wang, Y., Liu, C., Qiao, Y., 2023. Scene Reconstruction Algorithm for Unstructured Weak-Texture Regions Based on Stereo Vision. *Applied Sciences*, 13(11), 6407.

Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J., 2024b. Dust3r: Geometric 3d vision made easy. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Wang, X., Rottensteiner, F., Heipke, C., 2019. Structure from motion for ordered and unordered image sets based on random kd forests and global pose estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147, 19–41.

Weinzaepfel, P., Leroy, V., Lucas, T., Brégier, R., Cabon, Y., Arora, V., Antsfeld, L., Chidlovskii, B., Csurka, G., Revaud, J., 2022. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35, 3502–3516.

Yang, J., Sax, A., Liang, K. J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M., 2025. Fast3R: Towards 3D Reconstruction of 1000+ Images in One Forward Pass. *arXiv preprint arXiv:2501.13928*.

Yang, X., Jiang, G., 2021. A practical 3D reconstruction method for weak texture scenes. *Remote Sensing*, 13(16), 3103.

Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L., 2018. Mvsnet: Depth inference for unstructured multi-view stereo. *Proceedings of the European Conference on Computer Vision (ECCV)*, 767–783.

Zhan, Z., Yu, Y., Xia, R., Gan, W., Xie, H., Perda, G., Morelli, L., Remondino, F., Wang, X., 2025. SfM on-the-fly: A robust near real-time SfM for spatiotemporally disordered high-resolution imagery from multiple agents. *ISPRS Journal of Photogrammetry and Remote Sensing*, 224, 202–221.