TC-SfM: Robust Track-Community-Based Structure-From-Motion
Lei Wang, Linlin Ge, Shan Luo, Zihan Yan, Zhaopeng Cui, and Jieqing Feng

Abstract—Structure-from-Motion (SfM) aims to recover 3D scene structures and camera poses based on the correspondences between input images, and thus the ambiguity caused by duplicate structures (i.e., different structures with strong visual resemblance) always results in incorrect camera poses and 3D structures. To deal with the ambiguity, most existing studies resort to additional constraint information or implicit inference by analyzing two-view geometries or feature points. In this paper, we propose to exploit high-level information in the scene, i.e., the spatial contextual information of local regions, to guide the reconstruction. Specifically, a novel structure is proposed, namely, track-community, in which each community consists of a group of tracks and represents a local segment in the scene. A community detection algorithm is performed on the trackgraph to partition the scene into segments. Then, the potential ambiguous segments are detected by analyzing the neighborhood of tracks and corrected by checking the pose consistency. Finally, we perform partial reconstruction on each segment and align them with a novel bidirectional consistency cost function which considers both 3D-3D correspondences and pairwise relative camera poses. Experimental results demonstrate that our approach can robustly alleviate reconstruction failure resulting from visually indistinguishable structures and accurately merge the partial reconstructions.

Index Terms—Structure-from-motion, image-based reconstruction, ambiguous structures, track-community.

I. INTRODUCTION

STRUCTURE-FROM-MOTION is designed to recover camera motions and sparse 3D structures from image collections [1], [2]. This technique has been applied to various scenarios, such as indoor-outdoor 3D reconstructions [3], [4], natural environment monitoring [5], cultural heritage digitization [6], and recent neural rendering [7], [8]. The typical steps of SfM consist of feature detection, feature matching, camera poses estimation, and 3D structure reconstruction [9].

Although SfM methods have achieved impressive performance across numerous tasks, the existing methods still struggle to reconstruct the scene with duplicate structures accurately, which are common in the real world, such as the repetitive facades and decorations in buildings. The reasons lie in the image feature matching. If different instances share a highly similar appearance, their local features tend to be falsely matched, which leads to the incorrect pose estimation as well as the final 3D reconstructions like superimposed and phantom structures.

Existing work often deals with ambiguity by analyzing the feature points or epipolar geometries (EGs) between two views to explore consistent constraints. For example, some work attempted to remove inconsistent EGs by adding additional geometric consistency constraints between the views, such as loop constraints [10]. However, the effectiveness of these approaches is limited due to the accumulated geometric errors. Considering that the incorrect geometric relations stem from mismatched correspondences, a more fundamental solution is to analyze the visibility of points based on the correspondences, such as missing correspondences [11] and visibility graph [12]. The co-occurrence of feature correspondences can provide additional inference about ambiguous structures. Nevertheless, these methods are based on the local triplets or each feature point individually, and are prone to remove many positive correspondences. Since ambiguous structures usually exist in regions of the scene, the mismatched correspondences caused by ambiguous structures have a large correlation, which is ignored by previous methods. Solving the ambiguity at structure level can not only drastically simplify the task, but also exploit high-level information (i.e., scene structures), which is naturally exploited in human perception.

Based on these insights, we exploit the underlying spatial contextual information of the scene structures by grouping related tracks into the same cluster. Unlike the prior work that considers each point equally and ignores the underlying scene structures, our method considers the spatial relationship between the structures. In this case, the ambiguous structure, which usually belongs to a local region in the scene, can be directly detected at the region level.

To this end, we propose a novel track-community structure to partition the scene into several parts without reconstructing the scene in advance. This structure is obtained by analyzing the adjacency of the tracks and performing a community detection algorithm. Specifically, a track is defined as a set of matched 2D feature points from different views and corresponds to a 3D point in the real world. Accordingly, tracks can encode the visibility of the 3D points in each view, and a track-community refers to the local region of an object.
or several adjacent objects in the scene, namely, a segment of the scene, as shown in Fig. 1(c). Once the track-community is established, we can distinguish ambiguous structures via two steps, i.e., ambiguity detection and correction. In ambiguity detection, the diversity analysis of the tracks is used to determine whether a track-community potentially contains erroneous tracks caused by ambiguous structures. In ambiguity correction, the camera pose confliction checking is used to remove erroneous correspondences from ambiguous segments, resulting in several ambiguity-free segments.

For camera pose and 3D structure estimation, an incremental SfM approach is selected in our method because of its accuracy and robustness [9]. However, this approach usually suffers from the drift problem in large-scale reconstruction [13], [14]. To overcome this drawback, many SfM methods adopt the strategy of first distributingly registering the cameras and then merging them [13], [15]. Considering that the whole scene has been divided into several parts in the previous disambiguation step, we also utilize the partitioning registration strategy for SfM. To accurately merge all local reconstructions into a global frame, we refine the similarity transformations between them by a new cost function, which takes both 3D-3D correspondences and pairwise EGs into account.

In summary, the main contributions are listed as follows:

- We present a robust SfM method, i.e., TC-SfM, which explores the scene contextual information from track-communities to mitigate the problem of reconstruction failure caused by ambiguous structures.
- We propose a novel approach for detecting and correcting ambiguous structures by dividing the scene into segments and checking the pose consistency among segments.
- A new bidirectional consistency cost function is proposed to refine the similarity transformation between two partial reconstructions.

We conduct experiments on various datasets, and the results show that TC-SfM can robustly alleviate reconstruction failure resulting from ambiguity and achieve superior performance.

II. RELATED WORK

Many existing studies devote to improving the performance of SfM in the presence of ambiguous structures. These methods can be divided into three categories: refining view-graph construction, improving camera registration strategy, and post-processing reconstruction results.

A. View-Graph Refinement

The view-graph, where the nodes are images from different views and the edges are pairwise EGs between them, is an indispensable component in many SfM pipelines. The reconstruction performance will be greatly influenced by the contaminative view-graph.

One solution to the view-graph refinement is to directly remove the incorrect edges in the view-graph. Zach et al. [11] assumed that all three images of a triplet that share sufficient and well-distributed correspondences suggest the correct EG among all view pairs. The absence of enough correspondences (i.e., missing correspondences) provides strong evidence of the presence of an erroneous EG in them. Cui and Tan [16] also applied missing correspondence analysis between image pairs to filter EGs with large errors. Zach et al. [10] enforced the loop consistency of geometric relations estimated from the input. They detected conflicting relations in a Bayesian framework to infer the set of likely false-positive geometric relations. Roberts et al. [17] identified mismatched pairs based on an expectation-maximization framework that incorporates image timestamp cues with missing correspondence cues. However, the timestamp information in an unordered dataset is difficult to obtain, thereby limiting its usage. Wilson and Snavely [12] assumed that two observations visible on one view are also visible on other views and utilized the bipartite local clustering coefficient over the visibility graph to measure such consistency. The latest work for disambiguation is a learning-based approach proposed in [18].

The visual disambiguation problem is formulated as a binary classification task on image pairs by training a classifier that can learn both positive and negative evidence from data. These methods easily result in over-segment or incomplete models because all edges scoring below the threshold are directly deleted from the view graph.

Another solution is to find an optimal subgraph from the full view-graph, which can be regarded as a reliable input for the reconstruction to ensure the accuracy and completeness. Shen et al. [19] incrementally expanded the minimum spanning tree to form locally consistent strong triplets. They enforced loop consistency checking both locally and globally to guarantee the consistency of the matching set. Yan et al. [20] first introduced a geodesic consistency measure by selecting a set of iconic images. Correspondences that are connected in the visibility network but disconnected based on visual propagation along the path network are geodetically inconsistent.

The methods based on the analysis of the points or EGs aim to improve the quality of the view-graph. Due to the lack of higher contextual information, these methods usually remove a large number of positive edges, which tends to reduce the completeness of the reconstruction. In contrast, our method explores the spatial information among the regions in the scene to filter the correspondences from the ambiguous structures rather than directly remove the erroneous edges in the view-graph. This approach robustly detects ambiguous structures and improves the completeness of the reconstructed scene.

B. Camera Registration Strategy

Based on feature correspondences and EGs in the view-graph, registration is to determine the camera pose of each view. The structure of the scene is subsequently recovered based on camera poses. The performance of the reconstruction is also affected by the registration strategy, such as the registration order of view.

Although full view-graph contains many erroneous edges, some studies concentrate on improving camera registration strategy so that the cameras registered correctly are dominant. Cui et al. [21] introduced camera prioritization. A priori camera rotations are estimated accurately via rotation averaging on orthogonal maximum spanning trees of the view-graph and as weak supervision. Cameras whose rotation is consistent with
Fig. 1. Pipeline of the TC-SfM. Our method takes an image collection as input, and adopts the partition strategy for disambiguation and reconstruction. (a) View-graph construction. (b) Track sampling in superpixels. (c) Track-graph construction and community detection. (d) Image clusters corresponding to the segments. (e) Image clusters after disambiguation. (f) 3D models of partial reconstructions. (g) Merged 3D models in a uniform frame.

a priori rotation are preferentially registered and inconsistent cameras are subsequently registered. Chen et al. [15] proposed an image clustering algorithm and performed reconstruction in a divide-and-conquer manner. Some wrong EGs are discarded in clusters to prevent individual reconstructions from being affected by the wrong matches.

However, due to the influence of wrong matches, SfM methods based on improving the registration strategy usually struggle to recover the correct reconstruction when the scene structures share a strong visual resemblance. Although local reconstructions are correct, the merging step is still affected by outliers. In our work, we partition the scene into several segments and perform reconstruction individually. Benefiting from checking the pose consistency with the distinct structures, the ambiguous structures are detected so that both local reconstruction and merging are consistent.

C. Post-Processing

For visually indistinguishable structures, some studies correct ambiguities based on a reconstructed 3D model with erroneous elements. Such methods assume that a priori knowledge of the ambiguous structure is not available at registration time, thereby resulting in reconstruction failure. Heinly et al. [22] proposed the informative measure of conflicting observations to identify the incorrectly placed unique scene structures. Later, Heinly et al. [23] presented another post-processing pipeline to split an incorrect reconstruction into error-free models by exploiting the co-occurrence information in the scene geometry with local clustering coefficients. As the post-processing methods take reconstructions as inputs, they work well on many challenging datasets due to more useful information about ambiguous structures. Obviously, complete reconstructions are indispensable for these methods, thereby introducing additional computation costs.

Currently, most existing works focus on analyzing the EGs or feature matches to handle the ambiguity, but do not well exploit the underlying high-level contextual information before reconstruction. Nevertheless, our method explores the spatial relationship among the regions of the scene based on track-communities to detect and correct ambiguous structures.

III. TRACK-COMMUNITY-BASED SfM

To correct the ambiguous structures, we propose a track-community-based SfM method (i.e., TC-SfM). Fig. 1 illustrates the pipeline of the proposed SfM method. This method first performs feature extraction and matching, geometric verification, and EG estimation to construct the view-graph as the conventional pipeline. To partition the scene according to the scene structure, a track-graph is constructed. Subsequently, the tracks are divided into groups by a community detection algorithm. Each track-community is regarded as a scene segment, which roughly refers to the local region of the scene. Then, potential erroneous tracks are detected by diversity analysis on the track-graph. The segment that contains large erroneous tracks is potentially identified as ambiguous structures and corrected by checking the pose consistency with the help of other distinct segments. In this way, multiple ambiguity-free segments are obtained. Each segment is reconstructed via the standard incremental SfM. Finally, all partial reconstructions are aligned into a unified
framework by a bidirectional consistency constraint. A final BA is performed to minimize the global reprojection error of the whole reconstruction. The proposed method is described in detail in the following.

A. Track-Graph and Track-Community Construction

Some previous methods [12], [20] have demonstrated that tracks can be used to distinguish the ambiguities, as the co-occurrence of feature correspondences can provide additional inference before reconstruction. However, ambiguities often exist in regions of the scene, while previous methods analyze each track independently, ignoring the large correlation between tracks. In this section, we exploit the dependencies among the tracks to explore the underlying scene structures by constructing the track-graph and track-community.

Firstly, a full view-graph is constructed. Based on the conclusion in previous works [20], [24], the image that has the most matches with the given image is more likely to be the correct match. For a view pair \( (C_i, C_j) \), we calculate the ratio of the number of the common 2D observations to the total observations in each view, which is denoted by \( r_{ij} \) and \( r_{ij} \), respectively. The weight of an edge in the view-graph is defined as the average of two ratios (i.e., \( w_{ij} = \frac{r_{ij} + r_{ji}}{2} \)).

Ideally, a track corresponds to a 3D point in the real world. If two 3D points are on the same object and are close to each other, their 2D projections in the view are typically also close. Therefore, the neighborhood relations of the tracks are utilized to explore the contextual information of the scene. For improving efficiency, the full tracks must be simplified. Intuitively, features in close proximity on a surface usually exhibit similar observational behavior, so we utilize superpixel segmentation to sample the tracks. We perform Simple Linear Iterative Clustering [25] on each image to generate superpixels. Each track represents the local scene corresponding to the superpixel. If the superpixel region contains tracks, then the longest track will be selected because longer tracks can show the distribution of the same object on each view more fully so that the connectivity between the local scene can be captured more fully. More discussion of track selection strategy is provided in the supplementary material. Note that the correspondences in EGs with weights less than \( \tau_w \) are considered unreliable and ignored when searching correspondences for track sampling. All sampled tracks are regarded as nodes of the track-graph. If two tracks are visible in the same view and their 2D points are located in adjacent superpixels, then they are connected by an edge. The constructed track-graph can clearly display the surrounding information of a track.

Since the visual scenes are highly structured, the spatially proximal tracks exhibit strong dependencies, which carry high-level information about the scene’s structure as human perception. Similar to graph-based image segmentation [26], we divide the scene into several parts based on the track-graph. This problem can be formulated as a community detection problem without the need to predefine the number of groups. Community detection often arises in the complex network analysis [27]. A community consists of a set of nodes, which hold dense intra-connections but sparse inter-connections. Many previous studies [19], [28] explore the communities on the view-graph. An image-community consists of images with dense connections and is usually used for distributed scene reconstruction, while a track-community can segment the scene more finely because each track represents a point in the scene. In our method, the track-graph and track-community are used to divide the scene and analyze ambiguous structures. Upon obtaining ambiguity-free segments, the view-graph is used to collect image clusters for subsequent reconstruction and merging. In our implementation, the tracks are grouped by Louvain method [29] for community detection. Fig. 2(a) shows the community detection on the track-graph, which is divided into four communities \( \{TC_1, TC_2, TC_3, TC_4\} \). Each community represents a local segment of the scene, corresponding to an object or a group of adjacent objects within the scene. Fig. 1(c) shows the partitioning results of Books scene [17], where one community is labeled by one color.

B. Ambiguous Structure Detection

This step aims to find potential ambiguous structures of the scene by exploiting the spatial contextual information provided by track-communities. The track containing the mismatched correspondences is considered erroneous. Although such tracks may cover different regions of the real world due to the local similar appearance, their surrounding contents in the scene are usually different. Therefore, this challenge could be addressed by analyzing the surrounding information of a track in each view. The surrounding information of the correct track in each image is relatively consistent, while the erroneous track is more varied. Hence, we introduce Simpson’s Diversity Index to measure the diversity of the surrounding track-communities. Simpson index is often used to quantify the biodiversity of habitat, which considers the number of species
and the relative abundance of each species [30]. The variant of Simpson index, called Gini-Simpson Index (GSI) is typically utilized to measure the diversity [30]. In the track-graph, each track-community TC is treated as species Si. If a track Tj belongs to a species Sj, then its adjacent tracks that belong to other species are regarded as nonexistent. We count the number nj of individuals of each species {Si, i ≠ j} in the adjacent tracks. Therefore, the GSI gsi of a track can be calculated by:

$$gsi = 1 - \sum_{i=1}^{N_{adj}} \left( \frac{n_i}{N_{adj}} \right)^2,$$

where Nadj is the total number of the adjacent tracks belonging to other species; Nj is the number of other species. GSI represents the diversity of the surrounding information of a track across different views. Although GSI cannot find all the erroneous tracks in the scene, such as the tracks located within the inner part of the ambiguous structures, a large number of erroneous tracks can indicate that the community contains ambiguity. Accordingly, GSI is utilized to identify whether a community is ambiguous or distinct. For example, in Fig. 2(a), the dark red node Tj is visible in both Fig. 2(b) and Fig. 2(c), and it has three types of neighboring tracks belonging to other communities. P1 and P2 are the corresponding 2D keypoints in two views. P1 has two adjacent tracks of TC2 and two adjacent tracks of TC3. P2 has two adjacent tracks of TC3. A track with a large gsi is regarded as a potential erroneous track. We set this threshold empirically as rgs = 0.5. Fig. 3 shows detected potential erroneous tracks of Books. A track-community is deemed ambiguous if the ratio of erroneous tracks exceeds a threshold ξ (in our work, ξ is set to 0.2). That is, the corresponding segment contains ambiguous structures. Otherwise, the segment is classified as distinct.

**C. Ambiguous Structure Correction**

In this section, we introduce the method of correcting the ambiguous scene segments detected in the last step. During the incremental registration, 2D keypoints in the next candidate view and existing 3D points will be matched to calculate the camera pose. If all the 2D-3D matches are corrected, the poses estimated by matches from different segments of the candidate view are consistent. However, the existence of ambiguous segments results in inconsistent poses. Based on this observation, we compare the difference between the pose calculated from the distinct segments and the pose calculated from the potential ambiguous segments to extract consistent segments. Then, the correspondences from different segments are filtered during registration. After correction, each segment is reconstructed individually.
The translation error $e_t$ can be expressed by the angle error of two unit translation vectors, computed as $\cos^{-1}\left(\frac{t_1 \cdot t_2}{\|t_1\| \|t_2\|}\right)$, where $t_1$ and $t_2$ are the translation vectors from $\Omega_1$ and $\Omega_2$, respectively. In our work, if two poses have a rotation error exceeding $e_r(\epsilon_r = 0.15)$ or a translation error exceeding $e_t(\epsilon_t = 0.35)$ will be regarded as inconsistent. An image that satisfies pose consistency will be added to the reconstruction. Otherwise, it is excluded from the image cluster corresponding to the current segment. Its features originally associated with the current segment are also excluded. After checking all images in the current image cluster $C_i$, a consistent image subset without ambiguity will be obtained. For the remaining images in $C_i$, we repeat this process and obtain another consistent subset until there is no image in $C_i$. The outputs of this process include one or several consistent segments and their corresponding image clusters.

Once all potential ambiguous segments are corrected, each image cluster is consistent and free of ambiguity. To overcome the drift in incremental SfM, we independently reconstruct each segment based on its corresponding image cluster. After disambiguation, as shown in Fig. 1(e), $N_1$ consistent image clusters are obtained and correctly reconstructed by the traditional incremental SfM pipeline. Note that the original correspondences will be cleared by checking whether they lie in the same segment during registration. When numerous overlapping images between two clusters, we do not need to reconstruct two clusters independently to prevent redundant registration. The image clusters are sorted by the number of images in descending order. We merge two image clusters if the number of common images between them exceeds 20. To avoid unreliable reconstruction on clusters that have too few images (less than 20), clusters with images smaller than this threshold are merged into larger clusters. Specifically, if an image is associated with multiple segments, it will be included in other clusters. If an image is not in any cluster, we add it to other clusters based on its adjacent view in the view-graph.

### D. Local Reconstruction Merging

This section aims to merge the partial reconstructions into a complete 3D model in a unified framework. Each model is reconstructed in its local coordinate system originally. Some studies employ global motion averaging to obtain a globally consistent camera pose for each view. They use more accurate relative motions obtained by local incremental SfM [31] or set global pose prior refined by low-cost sensors as the reference [32]. Without performing additional global SfM, two local models, $model_a$ and $model_b$, can be merged by their relative similarity transformation $T_{ab} \in \text{SIM}(3)$, including rotation transformation, translation transformation, and scale [33]. By iterating through feature matches across two models and collecting corresponding triangulated points, common 3D points can be found. If the two partial reconstructions share common 3D points, these 3D-3D correspondences can be utilized to merge the 3D models by aligning two pieces of point clouds. Some existing methods tried to find overlapping views between two reconstructions [15]. However, overlapping view does not exist in many cases. Other studies solve the transformation by following the image-to-image constraint across two clusters [34]. Nevertheless, the unreliable EGs, which cannot be filtered by geometric validation, limit their performance. In our method, we do not design an additional expansion step to ensure the overlaps between image clusters like prior studies [15], [31]. Therefore, we design a novel cost function that incorporates 3D-3D correspondence and two-view geometry constraints to refine the similarity transformation with bidirectional consistency. The similarity transformation between the two models is initially estimated step by step via three linear equations, followed by optimization through the minimization of reprojection error with bidirectional consistency.

Let $T_{ab}(R_{ab}, t_{ab}, s_{ab})$ is the unknown relative similarity transformation from $model_a$ to $model_b$. $p_{i}^k$ is a 3D point visible in both two local reconstructions, and the local coordinates of $p_{i}^k$ are denoted as $p_{i}^{ab}_k$ and $p_{i}^{ba}_k$, respectively. $C_i^a$ is a camera that can see $p_i^b$ in $model_a$ and $C_i^b$ is a camera that can see $p_i^b$ in $model_b$. The pose of $C_i^a$ in $model_a$ can be denoted as $(R_i^a, t_i^a)$, where $R_i^a$ is the rotation and $t_i^a$ is the translation. Thus, $c_i^a$ denotes the camera position of $C_i^a$ in $model_a$, which is calculated by $-R_i^a t_i^a$. Similarly, $(R_j^b, t_j^b)$ is the pose of $C_i^b$ in $model_b$. The relative transformation from $C_i^a$ to $C_i^b$ can be denoted as $T_{ij}^ab(R_{ij}, t_{ij}, s_{ij}^{ab}, \lambda_{ij}^{ab})$, where $t_{ij}$ is a unit translation vector and $\lambda_{ij}^{ab}$ is the unknown scale.

1) Relative Rotation Estimation: We estimate the relative rotation between all partial reconstructions. The rotation between two 3D models can be obtained from EGs since the merged model should still satisfy the constraints between image pairs. Therefore, the relative rotation between the two 3D models can be estimated by using a linear system equation as:

$$R_j^b R_{ab} = R_{ij} R_i^a.$$  \hspace{1cm} (2)

2) Scale Estimation: To calculate the scale between two models, the distance between 3D points transformed into the same frame is minimized. The scale factor is estimated by:

$$s_{ab} R_{ij} (p_i^{ab} + t_i^a) + \lambda_{ij}^{ab} t_{ij} = R_j^b p_i^b + t_j^b.$$  \hspace{1cm} (3)

3) Relative Translation Estimation: With the rotation $R_{ab}$ and scale ($s_{ab}, \lambda_{ij}^{ab}$) fixed, the relative translation $t_{ab}$ between $model_a$ and $model_b$ are estimated based on the transformations of image pairs that cross two models. The linear equation system is defined as:

$$t_{ab} + s_{ab} R_{ab} c_i^a - \lambda_{ij}^{ab} (R_{ij}^b)^T t_{ij} = c_j^b.$$  \hspace{1cm} (4)

4) Bidirectional Consistency Optimization: After the initial similarity transformations between partial reconstructions are obtained, all parameters will be further optimized. Here, a novel cost function is designed to accurately merge the models by enforcing the 3D-3D correspondence constraint and two-view relative rigid transformation constraint. As shown in Fig. 5, the 3D point $p_i^b$ from the local coordinate system of $model_a$ can be transformed into the local coordinate system of $C_i^b$ in $model_b$ in two ways. We define $p_i^{ab,bj}$ as the 3D point transformed by the similarity transformation...
Fig. 5. Illustration of bidirectional consistency cost. The 3D point visible in two reconstructions are transformed from one model to another by two transformations. (a) Two partial reconstructions that share common 3D points. The green triangle denotes the common 3D point. (b) Two types of transformations. The red and black dotted lines represent the 3D points transformed by the similarity transformation and relative pose, respectively. (c) Two transformed points are projected onto the image plane, and $d_{k}^{ab,j}$ is the distance between their projected points.

$T_{ab}(R_{ab}, t_{ab}, s_{ab})$ and define $q_{k}^{ab,j}$ as the 3D point transformed by relative pose $T_{ij}^{ba}(R_{ij}, t_{ij}, s_{ab}, \lambda_{ij}^{ba})$ between $C_{i}$ and $C_{j}$. Then, we have:

$$p_{k}^{ab,j} = R_{b}^{j}(s_{ab}R_{ab}p_{k}^{a} + t_{ab}) + t_{b}^{j},$$

$$q_{k}^{ab,j} = s_{ab}R_{ij}(R_{ij}^{a}p_{k}^{a} + t_{ij}^{a}) + \lambda_{ij}^{ba}t_{ij}^{j}. \quad (6)$$

We want to enforce the constraint that two transformations should be consistent with each other. Therefore, $p_{k}^{ab,j}$ and $q_{k}^{ab,j}$ should be as close as possible. According to Eq. 3, $q_{k}^{ab,j}$ is close to the point of transforming $p_{k}^{a}$ to $C_{j}$. This indirectly enforces the 3D correspondence constraint. Furthermore, we utilize the reprojection error $d_{k}^{ab,j}$ to eliminate the range difference of the local models:

$$d_{k}^{ab,j} = \left\| p_{j}^{b}(p_{k}^{ab,j}) - p_{j}^{b}(q_{k}^{ab,j}) \right\|^{2}, \quad (7)$$

where $p_{j}^{b}(\cdot)$ means projecting the 3D point $\cdot$ onto the image plane of $C_{j}$ in model$b$. In the same way, $p_{k}^{b}$ are also transformed by the inversed transformations. The distance between the projected points is denoted as $d_{k}^{ab,i}$. Thus, the bidirectional consistency cost function is formulated as:

$$E = \sum_{(a,b) \in C_{a}} \sum_{i \in C_{a}} \sum_{j \in C_{b}} \sum_{k \in \mathbb{P}_{ij}^{a,b} \cap \mathbb{P}_{ij}^{b,a}} \omega_{ij}(d_{k}^{ab,j} + d_{k}^{ba,i}). \quad (8)$$

where $\omega_{ij}$ is the edge weight of $C_{i}$ and $C_{j}$ in the view-graph defined in Section III-A. We find the 3D point set $\mathbb{P}_{ij}^{a,b}$ observed by $C_{i}$ in model$a$ and $C_{j}$ in model$b$ for all reconstruction pairs.

The similarity transformations are refined by minimizing $E$.

After all the pairwise similarity transformations are refined, the partial reconstructions will be aligned to a global frame. We adopt the merging strategy proposed in GraphSFM [15], which relies on a cluster-graph and its Minimum-cost Spanning Tree $\mathcal{T}$. Each partial reconstruction is regarded as the node, and two nodes are connected if a similarity transformation exists between them. The weight of the edge is defined as the similarity transformation cost of Eq. 8. Firstly, the edges in $\mathcal{T}$ that connect the leaf nodes are selected for merging. We merge the model with fewer images into the other via the refined similarity transformation. $\mathcal{T}$ is updated by iteratively removing the leaf nodes. All the leaf nodes in $\mathcal{T}$ and their neighbors are merged in the same way. At last, only one node is left in $\mathcal{T}$, and all the partial reconstructions are aligned into a unified frame. To make the 3D points and all camera poses more accurate, we perform the final BA on the merged reconstruction to minimize the reprojection error.

IV. EXPERIMENTS AND DISCUSSIONS

In this section, we evaluate the proposed TC-SFM on four types of datasets: ambiguous image datasets, sequential image datasets, unordered Internet image datasets and datasets with ground truth. The specifications of these datasets are listed in Table I. The organization of the experiments is as follows:

- **Given that TC-SFM is targeted at the ambiguity problem, we first evaluate our method on 12 ambiguous datasets, which are associated with highly visual ambiguities and common in the real world. Traditional SFM pipelines usually fail to recover correct scene structures on these datasets. Therefore, we compare our approach with recent representative methods for evaluating disambiguation.**

- **In addition to focusing on ambiguity, we also refine the whole SFM pipeline. To demonstrate the versatility of TC-SFM, we evaluate its overall performance on the general image dataset, especially the sequential image datasets and the unordered Internet image datasets. These datasets are widely used in other SFM pipeline evaluations [35], [36]. On these bases, we compare the performance of our method with state-of-the-art traditional as well as deep learning-based SFM methods. We also demonstrate the scalability of our method on the large-scale scenes.**

- **Finally, quantitative evaluation is performed on KITTI odometry datasets [37] and our human body datasets [38] to show the accuracy and efficiency of our method. Moreover, we use them to demonstrate the validity of the bidirectional consistency constraint in merging.**

| Table I | SPECIFICATIONS OF THE IMAGE DATASETS. THE "GT" COLUMN REPORTS WHETHER THE DATASET HAS GROUND TRUTH | |
|---------|-------------------------------------------------|---|
| Dataset type | # of Images per dataset | GT | Main evaluation dimension |
| Ambiguous | 12 | 21–1559 | No | Disambiguation |
| Sequential | 4 | 152–1108 | No | Scalability & Versatility |
| Unordered | 15 | 247–5433 | No | Quantitative performance |
| KITTI | 11 | 542–9322 | Yes | & Ablation study |
In our implementation, each scene segment is reconstructed by the standard incremental pipeline of COLMAP [9] with the default configuration. Since the feature extraction and matching are common steps for SfM, the time consumption of these two steps is not included in the reported runtime. For a fair comparison, we input the same feature matches for all methods on each dataset. The experiments were conducted on a PC equipped with an Intel Core i9-9900K CPU (3.60GHz), 128GB of RAM and an NVIDIA RTX 2080Ti GPU. Moreover, the configuration of the parameters and the limitations of our method are provided in the supplementary material.

A. Evaluation on Ambiguous Image Datasets

We tested our TC-SfM on 12 datasets with large volume of ambiguous structures, including eight benchmark datasets, namely, Books [17], Desk [17], Street [17], Cup [17], Forbidden City [40], Sports Arena [19], Indoor [40], Temple of Heaven [40], as well as four larger Internet datasets, namely, Arc de Triomphe [22], Church on Spilled Blood [22], Brandenburg Gate [22], and Berliner Dom [22]. We compare TC-SfM with two traditional methods and two learning-based methods for evaluating disambiguation, namely, geodesic-aware SfM proposed by Yan et al. [20], GraphSfM [15], PixSfM [39] and Doppelgangers [18]. The geodesic-aware SfM and Doppelgangers are specifically designed for disambiguation. GraphSfM is a divide-and-conquer SfM method based on view-graph clustering as most of the existing methods. PixSfM is based on the featuremetric refinement, while Doppelgangers focuses on identifying whether a pair of visually similar images depict the same or distinct 3D surfaces. The comparison results are shown in Fig. 6.

Benefiting from deep features and featuremetric optimization, PixSfM produces reasonable results on some ambiguous datasets like Berliner Dom. However, for scenes with strong visual resemblance, such as Books, Church on Spilled Blood and Brandenburg Gate, PixSfM incorrectly registers cameras and points of duplicated structures into the same place.

For GraphSfM, images are divided into clusters by view-graph clustering. The reconstruction performance greatly depends on graph cutting. However, the mismatched image pairs usually have large edge weights, causing them to be grouped into the same image cluster. As shown in Fig. 6, GraphSfM fails to obtain reasonable reconstructions on some ambiguous image datasets, such as Berliner Dom and Arc de Triomphe.

The geodesic-aware SfM [20] performs well in sequential image datasets that have a uniform view distribution and sufficient overlaps between views, such as Temple of Heaven. However, the false EG removal and completeness of reconstruction are difficult to balance in unordered Internet image datasets with various FoVs and illuminations. The EGs that do not satisfy particular conditions are directly rejected, which greatly affects completeness. For Arc de Triomphe and Berliner Dom, the obtained several parts cannot be merged. In Books and Brandenburg Gate, a part of the scene is missing as shown in Fig. 6. If the thresholds of filtering are relaxed, then ambiguities cannot be detected.

By learning a classifier from data, Doppelgangers effectively disambiguates and reconstructs more scenes than other baselines that rely on hand-designed heuristics. However, it fails on datasets that have large differences from its training scenes, such as Books. In addition, Doppelgangers also compromises the completeness of the reconstructed scenes as it directly removes the edges from the view-graph with a probability below a certain threshold. This can be observed in Forbidden City and Sports Arena datasets, Doppelgangers produces incomplete results for image sequences.

In contrast, EGs are not directly handled in our TC-SfM. The scene is divided into several segments based on track-community. Furthermore, by performing track diversity analysis in the track-graph and pose consistency checking with distinct parts, the segments which contain ambiguities are detected and corrected. The correspondences derived from different segments will be discarded during registration. Therefore, erroneous correspondences belonging to ambiguous segments are filtered to recover correct camera poses and scene structures. For example, the highly ambiguous dataset Temple of Heaven has large erroneous EGs. The detailed processing results are shown in Fig. 7. The building is rotationally symmetrical and has a very similar appearance in all views. First of all, the scene is initially divided into 5 segments, as shown in Fig.7(c). Different colors represent different segments and the numbers below indicate the view of the corresponding position in Fig.7(b). Benefiting from other features on the ground, the lower part of the building is divided into 4 segments, while the upper part (yellow segment) of the building is divided into one segment. Then, a lot of potential erroneous tracks are detected in the yellow segment, indicating that it contains ambiguous structures. After checking the pose consistency with other distinct parts. The upper part of the building is further divided into two additional segments, as illustrated in Fig.7(d). After disambiguation, we perform partial reconstruction based on the segments and merge them into a unified framework. Fig.7(e) shows that each partial model is reconstructed correctly. Meanwhile, the feature correspondences from different segments are discarded, thereby ensuring that the merging step is not misguided by false pairwise matches. To demonstrate the necessity of ambiguous structure detection and correction. We also reconstruct the scene based on the initial segment, instead of performing ambiguity detection and correspondence filtering. The result is shown in Fig. 7(f). Different parts are merged incorrectly, while the correct model is obtained after disambiguation, as shown in 7(b). For datasets like Arc de Triomphe, Brandenburg Gate and Church on Spilled Blood, two sides of the building are divided into different segments. Thus, the scene is successfully recovered.

B. Evaluation on Sequential Image Datasets

The proposed TC-SfM method is evaluated on sequential image datasets from Tanks and Temples Datasets [42]. A uniformly distributed image set is provided for every scene from each high-resolution video sequence. Here, four representative image datasets, including outdoor and indoor environments, are selected for evaluation. Five
Fig. 6. Comparison of the reconstruction results on 12 ambiguous image datasets by using the geodesic-aware SfM [20], Doppelgangers [18], GraphSfM [15], PixSfM [39], and our TC-SfM, respectively. The first panel shows the sampled images of the dataset. The separate models are split with dashed lines.

On the outdoor dataset Family, Theia fails to recover the structures, while the other four methods successfully reconstruct the scene. The other three datasets contain slight visual ambiguities. The indoor datasets Auditorium and Meetingroom exhibit repetitive furnishings. The outdoor dataset Courthouse contains the same facades on the building. Although incremental SfM is more robust to correspondence outliers, COLMAP registers ambiguous structures in the wrong location. GraphSfM is also disturbed by mismatches in the merging step. PixSfM does not work well in the presence of large mismatches, such as in Courthouse. Doppelgangers successfully recovers the structures of Courthouse dataset, but some cameras are missing in Auditorium dataset. In contrast, TC-SfM achieves better results due to the filtering of matches from two distinct segments. For example, in Auditorium, the seats in the three regions are divided into three segments by TC-SfM, thereby alleviating interference from the mismatches.

C. Evaluation on Unordered Internet Image Datasets

We evaluated TC-SfM on unordered Internet image datasets from 1DSfM [43]. 1DSfM datasets consist of normal or less...
ambiguous scenes with different scales, and can be used to demonstrate the scalability and versatility of TC-SfM. Table II shows the comparison of the reconstruction results. Theia and PixSfM register more cameras than other methods. However, these methods have large average reprojection errors and less accurate results. Theia generated more reconstructed points.
TABLE II
COMPARISON OF THE RECONSTRUCTION PERFORMANCE ON THE UNORDERED IMAGE DATASET. \#N AND \#Reg Denote the Number of Views and Registered Views, Respectively. t Denotes the Runtime in Seconds. e Denotes the Average Reprojection Error. \#Points Is the Number of Reconstructed 3D Points. The Average Track Length of Points Is Also Reported. The Datasets Are Listed in the First Column. “Th”, “CM”, “GS”, “PS”, “DG” and “TS” Are the Abbreviation for Theia [41], COLMAP [9], GraphSfM [15], PixSfM [39], Doppelgangers [18] and Our TC-SfM, Respectively. The Best Results Are Highlighted in Bold Font.

| \#N | \#Reg | t [minute] | e [pixel] | \#Points (Avg. track length) |
|-----|-------|-----------|-----------|-----------------------------|
| Th  | CM    | GS        | PS        | DG | TS |
| AI  | 627   | 562       | 546       | 556 | 568 | 535 | 549 | 11 | 44 | 12 | 106 | 553 | 50 | 1.39 | 0.48 | 0.49 | 1.11 | 0.48 | 0.48 |
| IE  | 247   | 231       | 228       | 229 | 218 | 215 | 230 | 1 | 6 | 4 | 16 | 176 | 10 | 1.30 | 0.74 | 0.75 | 1.19 | 0.71 | 0.74 |
| GM  | 742   | 706       | 673       | 550 | 704 | 617 | 682 | 2 | 52 | 13 | 119 | 121 | 44 | 1.20 | 0.68 | 0.67 | 1.13 | 0.69 | 0.69 |
| MN  | 394   | 348       | 309       | 330 | 337 | 283 | 316 | 1 | 16 | 21 | 41 | 114 | 17 | 0.96 | 0.50 | 0.50 | 1.14 | 0.50 | 0.50 |
| MN  | 474   | 458       | 446       | 448 | 396 | 466 | 535 | 5 | 35 | 20 | 68 | 276 | 49 | 1.25 | 0.65 | 0.66 | 1.22 | 0.64 | 0.68 |
| ND  | 535   | 543       | 536       | 518 | 500 | 535 | 7 | 142 | 40 | 107 | 651 | 132 | 1.32 | 0.64 | 0.69 | 1.15 | 0.65 | 0.65 |
| NL  | 376   | 339       | 320       | 320 | 344 | 305 | 320 | 1 | 13 | 12 | 30 | 118 | 19 | 1.40 | 0.62 | 0.63 | 1.10 | 0.63 | 0.62 |
| PP  | 354   | 339       | 320       | 320 | 338 | 309 | 320 | 1 | 11 | 6 | 24 | 136 | 34 | 1.42 | 0.60 | 0.64 | 1.13 | 0.60 | 0.60 |
| PC  | 2187  | 2255      | 2133      | 2136 | 2180 | 1957 | 2144 | 19 | 290 | 124 | 1120 | 1796 | 258 | 147 | 0.65 | 0.65 | 1.23 | 0.64 | 0.66 |
| RF  | 1134  | 1079      | 1038      | 1029 | 1074 | 1010 | 1050 | 1 | 5 | 103 | 32 | 165 | 457 | 105 | 1.46 | 0.59 | 0.67 | 1.21 | 0.59 | 0.59 |
| TL  | 508   | 485       | 433       | 442 | 449 | 423 | 433 | 2 | 15 | 12 | 23 | 174 | 24 | 1.21 | 0.50 | 0.52 | 1.01 | 0.50 | 0.50 |
| Tr  | 5433  | 4946      | 4774      | 4706 | 4856 | 4400 | 4756 | 3 | 2362 | 2464 | 2410 | 1474 | 749 | 1.29 | 0.61 | 0.64 | 1.19 | 0.61 | 0.62 |
| US  | 930   | 807       | 696       | 831 | 606 | 730 | 1 | 5 | 58 | 17 | 77 | 137 | 47 | 1.52 | 0.62 | 0.68 | 1.12 | 0.66 | 0.62 |
| VC  | 918   | 845       | 768       | 780 | 774 | 711 | 785 | 5 | 158 | 59 | 127 | 579 | 157 | 1.39 | 0.56 | 0.58 | 1.16 | 0.57 | 0.58 |
| YM  | 458   | 428       | 411       | 415 | 433 | 385 | 408 | 7 | 35 | 20 | 51 | 189 | 39 | 1.32 | 0.61 | 0.65 | 1.08 | 0.61 | 0.61 |

but with shorter track lengths, also indicating less accurate poses as it fails to merge tracks associated with the same physical point in the scene. As shown in Fig. 8, for the dataset GM with ambiguous structures, Theia and GraphSfM produce incorrect models, indicating sensitivity to wrong matches, while other methods obtain correct reconstructions. Due to the weak and cluttered connections between images in unordered Internet datasets, Doppelgangers suffers from a significant problem of incomplete reconstruction. As shown in Table II, Doppelgangers leads to a considerable number of missing cameras due to the removal of excessive edges from the view-graph. In contrast, benefiting from a finer filtering strategy between different segments, TC-SfM achieves comparable robustness and accuracy to COLMAP in most scenes. In terms of time efficiency, Doppelgangers trains a classifier to remove the illusory pairs, and then uses the cleaned view-graph as input to COLMAP. However, the disambiguation step is quite time-consuming, which limits its usage. In comparison, the ambiguity detection and correction of TC-SfM are very efficient, its overall runtime is comparable to COLMAP on most datasets, and even faster on large-scale datasets.

D. Application in Human Body Reconstruction

We also apply our method to eight datasets with ground truth to quantitatively evaluate the accuracy compared with state-of-the-art SfM methods. Our human body acquisition system is equipped with 90 cameras [38]. We utilize a cuboid calibration object (Fig. 9(a)) with ChArUco patterns comparable to the human body size to calibrate all 90 cameras. The calibration results can serve as the ground truth to quantitatively evaluate the accuracy compared with other methods. The human body datasets contain large low-texture skin and clothes regions, and these features might not be included in the training data of PixSfM. Therefore, PixSfM tends to reconstruct the structures with prominent texture (e.g., the silhouette of clothes), resulting in low completeness and accuracy. COLMAP and GraphSfM perform well on D1-D6, which have fewer mismatches in all human body datasets. This deep learning-based method only recovers a few cameras and points in our human data, because such an approach is data-driven and might perform poorly in unfamiliar datasets. The human body datasets share a highly similar appearance. (e) Structures of front and back disambiguated by our method.
resemblance. The cameras located on the back are folded on the front side, as shown in Fig. 9(c). Doppelgangers achieves more accurate and robust reconstruction but also outlier feature correspondence filtering, TC-SfM not only alleviates drift and improve efficiency on large-scale datasets. Due to the lack of the scaling factor, COLMAP also exists. Both GraphSfM and our TC-SfM adopt the 7DoF similarity transformation to estimate ground truth trajectories to evaluate the accuracy and efficiency on large-scale datasets. Due to the lack of the scaling factor, a 7DoF similarity transformation is estimated to scale and align the predicted camera trajectories to the ground truth. The quantitative results are listed in Table IV.

While COLMAP achieves superior rotational errors on some datasets, it suffers from severe translational drift, as shown in Fig. 10. Doppelgangers is more accurate than COLMAP but less time-efficient due to the additional disambiguation step. As its reconstruction is based on COLMAP, the drift problem still exists. Both GraphSfM and our TC-SfM adopt the partitioning scheme, resulting in greatly reduced translational error and improved efficiency. Note that benefiting from outlier feature correspondence filtering, TC-SfM not only achieves more accurate and robust reconstruction but also higher efficiency on large-scale datasets than GraphSfM. The results in Table IV also demonstrate that the reconstruction accuracy can be effectively improved by using a bidirectional consistency cost function. Therefore, besides the ability to solve the ambiguity, another benefit of our method is to alleviate drift and improve efficiency on large-scale data.

E. Evaluation on KITTI Visual Odometry Datasets

We compared TC-SfM with three state-of-the-art and closely related SFM methods (i.e., COLMAP [9], GraphSfM [15] and Doppelgangers [18]) on KITTI visual odometry dataset [37]. We use 11 sequences (00-10) with ground truth trajectories to evaluate the accuracy and efficiency on large-scale datasets. Due to the lack of the scaling factor, a 7DoF similarity transformation is estimated to scale and align the predicted camera trajectories to the ground truth. The quantitative results are listed in Table IV.

The time cost of each component is reported in Table V. Note that the ambiguity correction is not time-consuming.
because we only perform PnP [44] and triangulation to obtain camera pose for checking consistency and don’t perform BA. Additionally, we plot the overall runtime on datasets of varying image numbers from 1DSfM and KITTI in Fig. 11. A comparison is made between three state-of-the-art SfM methods: the incremental method COLMAP [9], utilized for reconstructing each image cluster in our method, GraphSfM [15], which uses a partition registration scheme based on the view-graph, and the learning-based method Doppelgangers [18], which removes EGs by a classifier and then reconstructs the scene by COLMAP.

For a small number of images (less than 1k), the runtime of our method is similar to that of COLMAP. As the number of images increases, the runtime of COLMAP increases rapidly. Compared with GraphSfM, our method is not very efficient, especially when the number of images is less than 6k. This is mainly due to uneven image grouping as we cluster the images according to the scene segment without additional view-graph clustering. Nevertheless, benefiting from more reliable feature correspondence, our method achieves comparable efficiency to GraphSfM, or even faster, when the number of images exceeds 6k. The wrong EG detection of Doppelgangers is quite time-consuming, resulting in low time efficiency on all datasets. Therefore, our method presents notable efficiency advantages when dealing with large-scale datasets. Furthermore, there is potential for future work to improve the efficiency and accuracy of our method on larger-scale datasets by integrating the track-graph and a skeletal set of view-graph or combining a carefully designed image clustering algorithm.

V. CONCLUSION

In this work, a track-community-based SfM method is proposed to address the ambiguity problem caused by visually indistinguishable structures. The proposed track-community structure is used to partition the scene into several segments and enables disambiguation at the scene structure level. To detect potential ambiguous segments, we analyze the diversity of the neighborhood for each track. To distinguish the similar parts in the scene, we perform consistency validation between the poses estimated by the distinct and ambiguous segments. After obtaining ambiguity-free segments, each of them is individually reconstructed to mitigate the drift.

Correspondences from different segments are regarded as incorrect and are ignored during the registration. The proposed bidirectional consistency cost can refine the similarity transformations between the local reconstructions, thereby further improving the merging accuracy. The experiments show that TC-SfM can effectively alleviate reconstruction failure resulting from ambiguity and achieve a more robust and accurate reconstruction without sacrificing performance on normal datasets. Our future work includes further exploring the information between segments to improve the performance of reconstruction under extremely challenging cases.

REFERENCES

[1] S. Agarwal et al., “Building Rome in a day,” Commun. ACM, vol. 54, no. 10, pp. 105–112, Oct. 2011.
[2] N. Snavely, S. M. Seitz, and R. Szeliski, “Modeling the world from internet photo collections,” Int. J. Comput. Vis., vol. 80, no. 2, pp. 189–210, 2008.
WANG et al.: TC-SfM: ROBUST TRACK-COMMUNITY-BASED SfM

[17] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly, “Structure from motion,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1212–1221.

[18] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” J. Stat. Mechanics: Theory Exp., vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[19] L. Jost, “Entropy and diversity,” Oikos, vol. 113, no. 2, pp. 363–375, May 2006.

[20] S. Zhu et al., “Parallel structure from motion and global averaging,” 2017, arXiv:1702.08601.

[21] Y. Chen, Z. Yu, S. Song, T. Yu, J. Li, and G. H. Lee, “AdaSfM: From coarse global to fine incremental adaptive structure from motion,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2023, pp. 2054–2061.

[22] B. Bhowmick, S. Patra, A. Chatterjee, V. M. Govindu, and S. Banerjee, “Divide and conquer: Efficient large-scale structure from motion using graph partitioning,” in Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 273–287.

[23] M. Fang, T. Pollok, and C. Qu, “Merge-SfM: Merging partial reconstructions,” in Proc. 30th Brit. Mach. Vis. Conf., 2019, p. 29.

[24] O. Özyesi and A. Singer, “Robust camera location estimation by convex programming,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 2674–2683.

[25] B. Zhuang, L.-F. Cheong, and G. H. Lee, “Baseline desensitizing in translation averaging,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4539–4547.

[26] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3354–3361.

[27] Q. Zhang et al., “A robust multi-view system for high-fidelity human body shape reconstruction,” Comput. Graph. Forum, vol. 40, no. 5, pp. 19–31, Aug. 2021.

[28] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, “Pixel-perfect structure-from-motion with featuremetric refinement,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Oct. 2021, pp. 5967–5977.

[29] N. Jiang, P. Tan, and L.-F. Cheong, “Seeing double without confusion: Structure-from-motion in highlyambiguous scenes,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1458–1465.

[30] C. Sweeney, T. Hollerer, and M. Turk, “Theia: A fast and scalable structure-from-motion library,” in Proc. 23rd ACM Int. Conf. Multimedia, Oct. 2015, pp. 693–696.

[31] A. Knäpisch, J. Park, Q.-Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” ACM Trans. Graph., vol. 36, no. 4, pp. 1–13, Jul. 2017.

[32] K. Wilson and N. Snavely, “Robust global translations with 1DSfM,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., May 2023, pp. 2054–2061.

[33] L. Xue, D. Scaramuzza, and R. Siegwart, “A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation,” in Proc. CVPR, Jun. 2011, pp. 2969–2976.

Lei Wang received the B.S. degree in computer science from the Hefei University of Technology, Hefei, China, in 2018. She is currently pursuing the Ph.D. degree with the State Key Laboratory of CAD&CG, Zhejiang University, China. Her research interests include multi-view stereo and geometric estimation in computer vision.
Linlin Ge is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, China. He devotes his time in point cloud processing and deeply understands the challenges in this field, including outlier removal, registration, and segmentation. His research interests also extend to machine learning, where he is exploring new and innovative ways to improve this field.

Shan Luo received the Ph.D. degree from the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China, in 2022. She is currently a Lecturer with the College of Media Engineering, Communication University of Zhejiang, China. Her research interests include human body shape analysis and synthesis.

Zihan Yan received the B.S. degree from the Department of Computer Science, Zhejiang University, China, in 2022. She is currently pursuing the Ph.D. degree with the University of Illinois at Urbana–Champaign, Champaign, IL, USA. Her research interests include human–computer interaction (HCI), multimodal analysis, and deep learning.

Zhaopeng Cui received the Ph.D. degree from Simon Fraser University in 2017. He was a Senior Researcher at ETH Zürich. He is currently a Research Professor with the College of Computer Science, Zhejiang University. His research interests include 3D mapping and localization, 3D scene understanding, 3D content generation, and image and video editing.

Jieqing Feng received the Ph.D. degree in computer graphics from Zhejiang University in 1997. He is currently a Professor with the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include geometric modeling, real-time rendering, stereo vision, and modeling and simulation in solar thermal power.