User-Guided Correction of Reconstruction Errors in Structure-from-Motion

Sotaro Kanazawa
sotaro-kanazawa317@g.ecc.u-tokyo.ac.jp
The University of Tokyo
Tokyo, Japan
Preferred Networks, Inc.
Tokyo, Japan

Jinyao Zhou
dayaogen@gmail.com
The University of Tokyo
Tokyo, Japan
Preferred Networks, Inc.
Tokyo, Japan

Yuta Kikuchi
kikuchi@preferred.jp
Preferred Networks, Inc.
Tokyo, Japan

Sosuke Kobayashi
sosk@preferred.jp
Preferred Networks, Inc.
Tokyo, Japan

Chunyu Li
chunyuli@preferred.jp
Preferred Networks, Inc.
Tokyo, Japan

Fabrice Matulic
fmatulic@preferred.jp
Preferred Networks, Inc.
Tokyo, Japan

Takeo Igarashi
takeo@acm.org
Preferred Networks, Inc.
Tokyo, Japan

Keita Higuchi
khiguchi@acm.org
Preferred Networks, Inc.
Tokyo, Japan

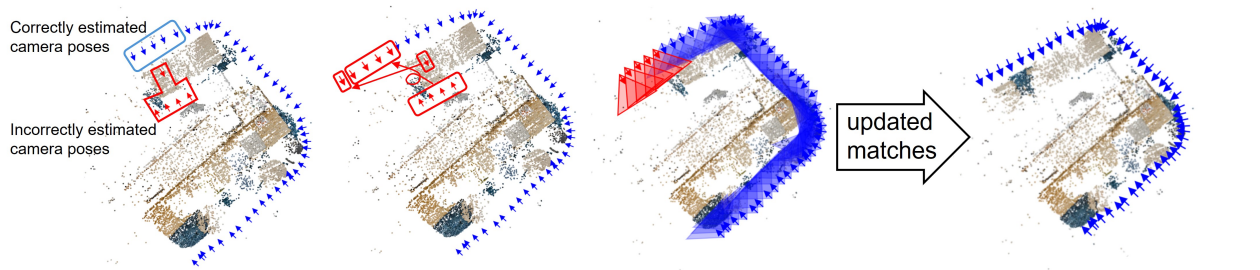

Figure 1: User-guided correction of reconstruction errors in Structure-from-Motion (SfM) workflows. (a) Initial reconstruction results showing both correctly and incorrectly estimated camera poses. (b) a user adjusts camera poses, and (c) defines camera view overlaps. The system identifies false matches based on the overlap of these camera views. (d) Improved reconstruction result with accurate camera poses after user-guided false match removal.

Abstract

We propose a user-guided method to correct reconstruction errors in Structure-from-Motion (SfM) processes. SfM takes a set of camera images as input and then estimates the cameras' poses and three-dimensional point clouds based on keypoint matching. However, scenes with repetitive or similar structures often result in false matches, leading to inaccuracies in camera pose estimation. While automatic methods for removing false matches exist, achieving perfect accuracy with them remains challenging. Conversely, human intervention can ensure high accuracy, but manual identification and elimination of false matches is a tedious and error-prone process. Our proposed method strikes a balance by introducing a more efficient user-guided approach. Users provide approximate camera poses, which the system then uses to detect false matches. Specifically, the system examines overlaps between view frustums of camera pairs post user adjustments, classifying pairs as false matches if no overlap is found. This method leverages the user's recollection of camera movements during scene capture to guide the reconstruction process. Evaluation with test cases and a user study confirm that our technique can efficiently remove false matches and enable accurate reconstruction of camera poses.

IUI '25, March 24–27, 2025, Cagliari, Italy

ACM ISBN 979-8-4007-1306-4/25/03
https://doi.org/10.1145/3708359.3712141

CCS Concepts

• **Human-centered computing → Human computer interaction (HCI)**; • **Computing methodologies → Artificial intelligence**.

Keywords

Structure-from-motion, Human-in-the-loop computer vision

ACM Reference Format:

Sotaro Kanazawa, Jinyao Zhou, Yuta Kikuchi, Sosuke Kobayashi, Chunyu Li, Fabrice Matulic, Takeo Igarashi, and Keita Higuchi. 2025. User-Guided Correction of Reconstruction Errors in Structure-from-Motion. In *30th International Conference on Intelligent User Interfaces (IUI '25), March 24–27, 2025, Cagliari, Italy*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3708359.3712141

1 Introduction

Structure-from-Motion (SfM) is a crucial technique in three-dimensional (3D) reconstruction that estimates camera poses (positions and orientations), as well as the point cloud of a scene, from multiple 2D images. SfM constitutes the initial workflow in image-based 3D reconstruction, and its results serve as the foundation for recent advanced reconstruction methods such as Neural Radiance Fields (NeRF) [22] and 3D Gaussian Splatting [16].

However, SfM faces several challenges. One significant issue arises when similar or identical structures are present within the captured space, leading to false keypoint matches. When false keypoint matches occur, the relative positions between cameras are incorrectly estimated, which can cause the reconstruction to fail. Specifically, cameras may be reconstructed at positions different from their actual positions at the capture time, resulting in decreased accuracy or catastrophic collapse of the reconstructed scene.

Several approaches are used to correct reconstruction errors in SfM. The most direct method is to capture additional images. SfM can fail when there is insufficient matching between images. Therefore, adding images taken from new viewpoints can increase the number of matches and improve the reconstruction accuracy. However, this method is feasible only when additional capture is possible and becomes challenging in environments that change frequently or are difficult to access.

When additional capture is impractical, adjustments to SfM parameter settings, such as the thresholds and methods for keypoint matching, can be attempted to improve reconstruction. SfM software like COLMAP [27] allows for various parameter adjustments, including keypoint extraction methods and the search range for image matching. However, determining which parameter changes will be effective is often not intuitive, and merely adjusting parameters may not suffice to correct reconstruction errors.

Recent studies have highlighted that similar structures present in the reconstruction environment can cause keypoint ambiguity, leading to false matches [11]. For example, when objects with similar appearances at the front and back are photographed from different directions, SfM may misinterpret them as being captured from only one direction [34]. To address such false matches, a machine learning (ML) approach has been proposed [4] to automatically detect and remove false matches via image-pair classifications, followed by reconstruction. However, machine learning approaches depend on training data and are only effective in environments reflected in that data, limiting their generalizability.

In this paper, we propose an interactive method to efficiently correct reconstruction errors by removing false matches using minimum human intervention. Specifically, users remove false matches and perform reconstruction based on the improved matching information. A straightforward method involves users identifying cameras with incorrect poses in the reconstruction results and manually removing false matches associated with those cameras by visually comparing matched camera images. However, it requires significant effort and time, making it impractical for regular use.

Our user-guided approach removes false matches by allowing users to manually specify the approximate camera poses to guide the reconstruction process. This method assumes that users have knowledge of or can easily identify the image sequences and the capturing environment of the reconstruction target. Users identify and adjust images with incorrect camera poses in a top-down view showing the reconstructed scene. Then, matches between two images without overlap of their view frustums are removed, and reconstruction is performed again. It is not necessary for a user to set the cameras' poses precisely because those poses are used just as guidance for false match removal and the final poses are determined by bundle adjustment when performing reconstruction. This iterative refinement process is repeated until the user obtains satisfactory results. Our proposed approach focuses on the correction of SfM errors in small-scale datasets or the subsets of large-scale datasets.

To evaluate our approach, we constructed a small-scale 3D reconstruction dataset consisting of 48 images designed to replicate situations prone to false matches. We confirmed that applying standard SfM methods to this dataset resulted in false matches and reconstruction failures. To verify the effectiveness of our proposed method, we compared it with other reconstruction correction methods, including parameter adjustments and machine learning-based false match removal techniques. Furthermore, we conducted a user study to assess the effectiveness of false match removal with our method compared to a baseline, in which the user manually identifies and removes false image matches. The results demonstrated that participants were able to more successfully remove false matches with our technique and achieve higher reconstruction quality with less effort compared to the baseline. Based on our findings, we discuss the effectiveness and limitations of our approach.

To summarize, our contributions are threefold:

- We propose a novel user-guided method that removes false matches in SfM by utilizing user-provided camera poses and view frustums.
- With our dataset, we demonstrate the effectiveness of our technique and conduct comparative evaluations with other correction methods.
- We show through a user study that our method enables high-precision reconstruction corrections with less effort compared to manually identifying and removing false matches.

2 Related Work

2.1 Structure-from-Motion and Its Applications

Structure-from-Motion (SfM) is a well-established technique in computer vision and photogrammetry that involves reconstructing 3D structures from 2D image set taken from different viewpoints. This technique simultaneously estimates the camera poses and the sparse 3D structure of the scene.

From early algorithms and mathematical frameworks by Tomasi and Kanade [29] to recover 3D structure and camera motion from image correspondences, the process has been iteratively refined since [1, 9], with robust software tools, such as the popular COLMAP [27], now available to effectively perform various 3D reconstruction tasks. COLMAP operates by extracting keypoints, matching them across multiple views, and then optimizing both camera poses and 3D points based on incremental SfM. In a recent study, Pan et.al proposed GLOMAP as a new general-purpose system that takes a global SfM approach for superior accuracy and robustness [26].

Recent advancements in 3D reconstruction and rendering heavily rely on COLMAP for pre-processing. For instance, approaches like NeRF (Neural Radiance Fields) [22] and its derivatives, including Instant NGP [23], Zip-NeRF [3], Mip-NeRF [2], NeuS [32], and NeRF++ [38], require accurate camera poses, which are often obtained with COLMAP, to initialize their rendering pipelines.

Similarly, 3D Gaussian Splatting [16], including recent works [5, 8, 21, 35–37, 39], also employ COLMAP to estimate camera parameters and reconstruct sparse 3D points. In these applications, improving the accuracy of COLMAP's reconstructions, particularly with respect to camera pose estimation and the completeness of the 3D point cloud, directly benefits the quality of subsequent reconstructions using NeRF or 3D Gaussian Splatting.

Other methods like VSRD [20] utilize 3D reconstruction to achieve instance-level recognition in complex 3D scenes. In such cases, accurate camera positioning and high-quality 3D point clouds, as produced by COLMAP, are essential to improving object detection and recognition performance.

In conclusion, SfM, particularly through tools like COLMAP, has become critical for 3D reconstruction. Enhancing COLMAP's reconstruction accuracy and completeness will benefit not only applications based on NeRF and 3D Gaussian Splatting but also other techniques that rely on accurate 3D scene understanding.

2.2 Correction of SfM Failures

Structure-from-Motion (SfM) techniques, especially those implemented in COLMAP [27], can achieve high-quality results in 3D reconstruction from images, but they often fail when dealing with scenes that contain visually similar or repetitive patterns. In such cases, due to the strong resemblance between parts of the scene, COLMAP may erroneously match images from different locations, resulting in incorrect 3D reconstructions. These false matches can lower the overall accuracy of the reconstruction and, in some cases, prevent COLMAP from generating a valid 3D point cloud or estimating accurate camera poses.

Several methods have been proposed to improve COLMAP's performance in handling scenes with similar or repetitive structures [4, 7, 11, 15, 34]. One notable example is the work by Cui et al. [7], which introduces a global optimization approach using similarity averaging to refine camera poses and structure. Similarly, Heinly et al. [11] present a post-processing method that detects and corrects errors caused by duplicate scene structures, leading to more accurate reconstructions.

Kataria et al. [15] provide a solution by introducing a subset of reliable matches for camera pose estimation, helping to reduce ambiguity in complex scenes. Finally, Doppelgangers [4] proposes a learning-based approach for distinguishing between visually similar but distinct 3D surfaces. This method effectively reduces errors in SfM pipelines by disambiguating matches in challenging cases and improving reconstruction results.

Because these automatic SfM correction methods heavily rely on underlying problem settings and training data, users have limited ways of intervention when these methods fail to correct errors. Control points in the RealityCapture software[1] serve as tools for user intervention in 3D reconstruction. While they are effective for enhancing reconstruction accuracy and merging models, correcting incorrectly estimated camera poses is challenging for users. To address these remaining challenges, we propose a user-guided method aimed at further enhancing COLMAP's reconstruction accuracy in such scenes.

2.3 Interaction Design in Human-in-the-Loop Computer Vision

Several studies have explored the effectiveness of interactive design for human-in-the-loop frameworks for the computer vision domain. For example, such kinds of systems have significantly enhanced the efficiency of annotation and data augmentation in computer vision tasks [12, 12, 17]. LabelAR [17] is an augmented reality (AR) interface for automatically generating 2D image bounding boxes by guiding users to capture images from various angles.

Recent studies have also focused on human-in-the-loop frameworks to achieve effective training and inference of computer vision models [10, 13, 18, 24, 33]. Zensor [18] utilizes real-time feedback from online crowd workers for automatically annotating datasets used to train machine learning models working as intelligent sensors. SwipeGANSpace [24] is a human-in-the-loop approach in which users can generate desired images by exploring the latent space of StyleGAN [14] using simple swipe operations. Weber et al. [33] propose the combination method between the human-in-the-loop process and Deep Image Prior [31].

In addition, there are some user interventions related to the 3D domain of computer vision [17, 19, 25, 25, 30]. iPose [19] demonstrates the importance of user involvement in reconstructing human poses from video data. Their system enables users to adjust 3D poses using 2D input, making it easier to handle complex poses and ambiguous visual information. PhotoCity [30] is a competitive game designed to make users proficient at collecting useful image datasets for 3D modeling.

Overall, designing interactions for human-in-the-loop computer vision is crucial for overcoming the limitations of fully automated methods. In particular, our research concentrates on facilitating effective human intervention in SfM processes. By exploring and developing interactive methods that enable users to correct false

[1] https://www.capturingreality.com

matches, we address the challenges that fully automated reconstruction often cannot overcome.

3 Proposed Method: User Guided Removal of False Matches in SfM

3.1 Problem Definition and Target Setting

The problem addressed in this study is the situation where, when users utilize Structure-from-Motion (SfM) for 3D reconstruction from images, false matches between images lead to erroneous estimation of camera poses, as well as reconstructed point clouds. In SfM software such as COLMAP [27], keypoint extraction and image matching are performed in the initial stage of reconstruction. During the keypoint extraction, an SfM algorithm identifies distinctive visual features in the images, such as corners or edges, which can be reliably recognized across different views. In the image matching stage, these keypoints are compared across pairs of images to find correspondences, ensuring that overlapping features between images are accurately identified, and the results are stored in a matching database[2]. Subsequently, based on this database, the spatial relationships between cameras and the 3D positions of corresponding keypoints are reconstructed. However, if a false match exists during keypoint matching, the reconstruction error increases. This issue is particularly prevalent when multiple structures exhibiting similar visual patterns exist within the captured environment, leading to frequent false matches. We address this type of problem and attempt to correct reconstruction errors by removing false matches through an effective user-guided process.

We specifically consider scenarios where re-capturing images is difficult. Capturing additional images can improve the accuracy and quality of the reconstruction, but it is not always feasible. In particular, changing environments (e.g., event venues, construction sites) and where access is inherently challenging (high-security sites, hospitals, etc.) cannot be sampled again in the same conditions. Furthermore, we make the following assumptions about the users of our system: 1) They have basic practical knowledge of 3D reconstruction and how image data should be captured for SfM. 2) They have some understanding of the environment to be reconstructed, and they know the paths that the camera followed when the scene was captured. In essence, we consider users who either have captured the scene themselves or can identify the scene and its capturing conditions using the source images or additional information they have.

3.2 Method Overview

Our approach is based on an iterative refinement process, where users utilize their knowledge of the captured scene to adjust camera poses and their view frustums when they notice errors. The system then updates the 3D reconstruction to reflect those corrections (Figure 2).

Specifically, users identify incorrectly positioned cameras and adjust their poses to approximately match their real positions. Users also adjust the view frustums of these cameras. If the view frustums of two cameras do not overlap, the system judges that the match between the two camera images is false and removes the images from the matching database. Subsequent SfM reconstruction based on the updated matching database may return improved camera poses and enhance the overall reconstruction quality. This process can be repeated until the user is satisfied with the results.

[2]More precisely, the system finds matches among keypoints in the images. Then, if sufficient matches between keypoints in two images are detected, the image pair is registered as a match in the matching database. When the pairing of two images is removed, then all the matches between keypoints in the image pair are removed.

3.3 User-Guided Removal of False Matches

We describe our user-guided method for removing false matches.

3.3.1 Identifying Incorrectly Reconstructed Cameras. The user examines the sequence of camera images and the reconstruction results (camera poses and point cloud) to identify cameras that have been incorrectly positioned in the 3D scene. Objects in the reconstructed point cloud that are placed in incorrect positions serve as clues to track the causes of reconstruction errors. The (dis)continuity of the camera trajectory may also provide another clue. To facilitate the identification of such errors, our system simultaneously presents the user with a top-down view of the reconstructed point cloud and the detected camera poses, as well as the sequence of camera images.

3.3.2 Manual Adjustment of Camera Poses. The user manually adjusts the inaccurately reconstructed camera poses to approximate their actual positions during capture. Leveraging their knowledge of the cameras' capture paths, users can position the cameras close to their true locations. However, due to the limitations of manual placement and operating a 2D user interface, precisely specifying 3D camera poses can be challenging. Consequently, these user-specified poses serve primarily as guidance for the system to refine the reconstruction. Once false matches are effectively removed, the system optimizes the camera poses in subsequent reconstruction stages.

3.3.3 Adjusting Matchability View Frustums to Guide False Match Removal. After adjusting the camera poses, the user can tune the extent of false match removal by refining "matchability" view frustum, imitating the real view frustum of each camera. While the original view frustum is three-dimensional, for simplicity, we represent these frustums as two-dimensional isosceles triangles with the camera position at the apex (Figure 3). The apex angle of these triangles corresponds to the camera's field of view (FoV), which determines the width of the area captured by the camera. The height of the triangle represents the shooting range, indicating the maximum distance the camera can capture in that direction. By manipulating the base length (FoV) and the height (shooting range), the user can precisely control the overlap between each camera's shooting ranges. Our method operates under the assumption that matching does not occur between cameras whose view frustums do not overlap since common keypoints must be visible in both camera views. The system allows the user to iteratively adjust these parameters until the reconstruction results are satisfactory.

Defining view frustums in two dimensions simplifies the user's task of specifying camera views. Humans have limited spatial awareness when interpreting complex three-dimensional spaces, especially when dealing with numerous cameras. Accurately assessing and defining overlaps among many cameras in 3D space can be

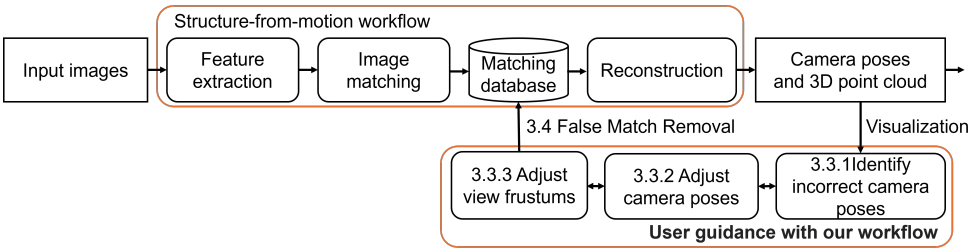

Figure 2: Overview of the proposed user-guided Structure-from-Motion (SfM) workflow. The process starts with input images and metadata, followed by keypoint extraction and matching, which are stored in a matching database. After the initial reconstruction, camera positions and 3D points are estimated. If the user finds incorrect camera positions in the reconstruction, they manually adjust the camera positions. Based on those new positions, the system removes false matches and updates the reconstruction. The process is repeated until all cameras are correctly positioned and the 3D scene is properly reconstructed. The numbers indicate corresponding sections.

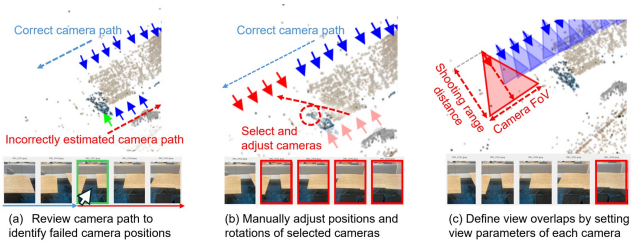

Figure 3: User-guided removal of false matches. (a) The user compares correctly and incorrectly estimated camera poses, (b) selects and manually adjusts incorrect camera poses, and (c) modifies the camera's view frustum. That information allows efficient identification of false matches and enables iterative refinement of reconstructed camera poses based on view frustum overlaps.

cognitively demanding and error-prone. By representing view frustum as two-dimensional regions (i.e., isosceles triangles on a plane), users can more intuitively and efficiently specify the approximate areas covered by each camera. This simplification reduces cognitive load and facilitates the identification of overlaps between camera views, which is crucial for the false match correction process. While this approach ignores the 3D nature of true view frustum, it strikes a balance between usability and effectiveness, allowing users to contribute meaningfully to the correction without being overwhelmed by spatial complexity. This simplification works well for scenes where camera moves are mainly horizontal, but it faces challenges when camera motion is more three-dimensional.

3.4 Algorithm for False Match Removal

Following the user's manual adjustments of the cameras, the system updates the database, removing false matches among images. We adopt exhaustive matching for the initial image matching to consider potential matches between all camera pairs. In the false match removal step, we remove matches whose view frustums do not overlap. Algorithm 1 shows the false match removal process. After removing false matches, reconstruction is performed based on the updated database, and the corresponding results are presented to the user in the scene view.

Algorithm 1 False Match Removal

Require: C_{moved}: Set of moved camera IDs
Require: C_{all}: Set of all camera IDs
Require: current_matches: Set of current matches
Ensure: Updated current_matches after removing false matches

```
Initialize false_matches ← ∅
for each c in C_moved do
    for each c′ such that (c, c′) is in current_matches do
        Compute View(c) and View(c′)
        if View(c) ∩ View(c′) == ∅ then
            false_matches ← false_matches ∪ {(c, c′)}
        end if
    end for
end for
current_matches ← current_matches \ false_matches
```

4 Prototype System

We implemented a prototype system to demonstrate and test our proposed method.

4.1 Overview

Figure 4 shows a screenshot of the prototype system. It consists of the following components: Component (a) displays the reconstructed camera poses and point clouds. Component (b) is a panel that allows the user to browse the sequence of images used in the reconstruction. When searching for incorrectly estimated camera poses, the user refers to components (a) and (b). When the user hovers the mouse over an image in (b), the corresponding camera is highlighted in green in (a). The user checks whether the estimated camera poses of the images are correct, and if an incorrect viewpoint is found, the user clicks on the corresponding image to select the camera, then moves the camera (shown in red) in (a) using the keyboard (arrow keys for translation and "q" and "r" keys for rotation). The user can select and display multiple images simultaneously, and the poses of the selected cameras can be adjusted

together. Additionally, by pressing the "Show/Hide Matching" button, the user can view other cameras that match the selected camera in the matching database.

Component (c) is the UI's status bar, which displays the system process during which the user cannot edit, e.g., generating the top-down view from COLMAP results, deleting matches from the database, and executing the reconstruction.

Component (d) is a panel that displays the group of matched images when the "Show Matching" button is pressed for a selected image in (b). The matched cameras are connected by lines in (a). Furthermore, this component supports the manual removal of matches. When the user discovers incorrect matches in (d), they can select them and press the "Delete Selected Matching" button to remove the specified matches. This function can be used in combination with our technique, and we use it as a baseline interaction method in our user evaluation (see below).

In component (e), the user can show and hide view frustums in the scene panel (a). When the view frustums are visible, the user can adjust the view frustums of the selected cameras. By using the "Delete Distant Matching" button in (f), the user can request the system to execute false match removal based on the modified camera poses and view parameters. By pressing the "Run Reconstruction" button, the reconstruction process can be executed.

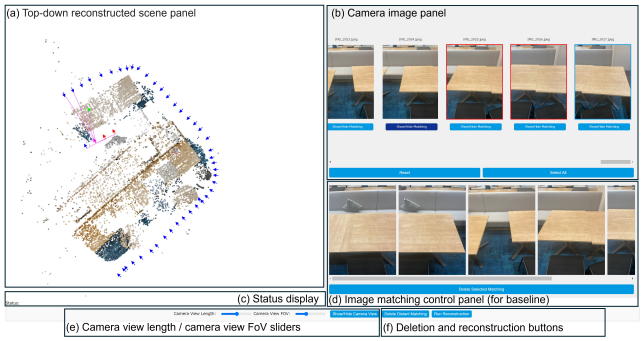

Figure 4: Overview of the prototype system. It contains several components: (a) Scene Panel, showing the 3D reconstructed space with camera positions; (b) Camera Image Sequence Panel, allowing users to confirm images, showing matching visibility for individual images, and selecting all images for further adjustments; (c) Status Display; (d) Image Matching Control Panel; where users can directly delete false matches (we use this function for annotation and baseline approach); (e) adjust view frustums of selected cameras; and (f) delete matches between non-overlapping cameras and run reconstruction buttons.

4.2 Implementation Details

We implemented the prototype system as a web-based application that receives SfM reconstruction results and input images from COLMAP. Initially, ground detection is performed on the point cloud reconstructed by COLMAP, which, combined with camera pose information, generates a top-down view of the reconstruction. Users then perform adjustments on incorrectly positioned cameras, upon which false matches are identified and marked for removal, with COLMAP updating its matching database accordingly. The processes of removing false matches and re-executing the reconstruction are carried out using the Pycolmap[3] library. Additionally, intermediate results are incrementally backed up, enabling users to revert to any previous version through the command-line interface.

5 Comparison of Error Correction Methods

5.1 Data Collection

To evaluate the effectiveness of our proposed method, we constructed a small dataset consisting of 48 images captured with an iPhone 12 Pro while moving through an office environment. Multiple desks with identical textures were present in this scene, which caused false matches in a direct reconstruction. Specifically, when COLMAP was executed with a simple camera model (Simple Pinhole Model) and exhaustive matching settings on this dataset, the estimated positions of 21 out of 48 cameras were significantly different from their actual positions, as shown in 6 (a).

Figure 5: The dataset images were captured using an iPhone 12 Pro. A total of 48 images were taken while moving through an office environment with multiple desks with identical textures.

5.2 Annotation of Ground-Truth False Matches

We manually labeled images corresponding to incorrectly positioned cameras to create the ground truth data for the false match removal task. Specifically, the human labeler visually inspected all images and removed instances with incorrect camera positions, i.e., cameras with different positions and orientations and with shooting areas that do not sufficiently overlap. This task utilized the manual removal function of the prototype system, which serves as the baseline method. The labels were created by the person who captured the dataset and required more than 30 minutes to complete. After removing false matches with this carefully curated dataset and updating the reconstruction, we obtained high-quality results which we use as ground truth.

Figure 6 (b) shows the reconstruction results after false match removal. By thoroughly removing all false matches, camera positions are correctly estimated, and the quality of the 3D point cloud is improved. However, manual removal of false matches through visual inspection is a time-consuming task. The result shown in Figure 6 (c) is from a user experiment discussed in Chapter 7 where a participant (P3) performed manual removal for 5 minutes.

5.3 Error Correction with Our Proposed Method

We detail the step-by-step process by which users correct errors in the test dataset using our proposed method. First, the user reviews

[3]https://github.com/colmap/colmap/tree/main/pycolmap

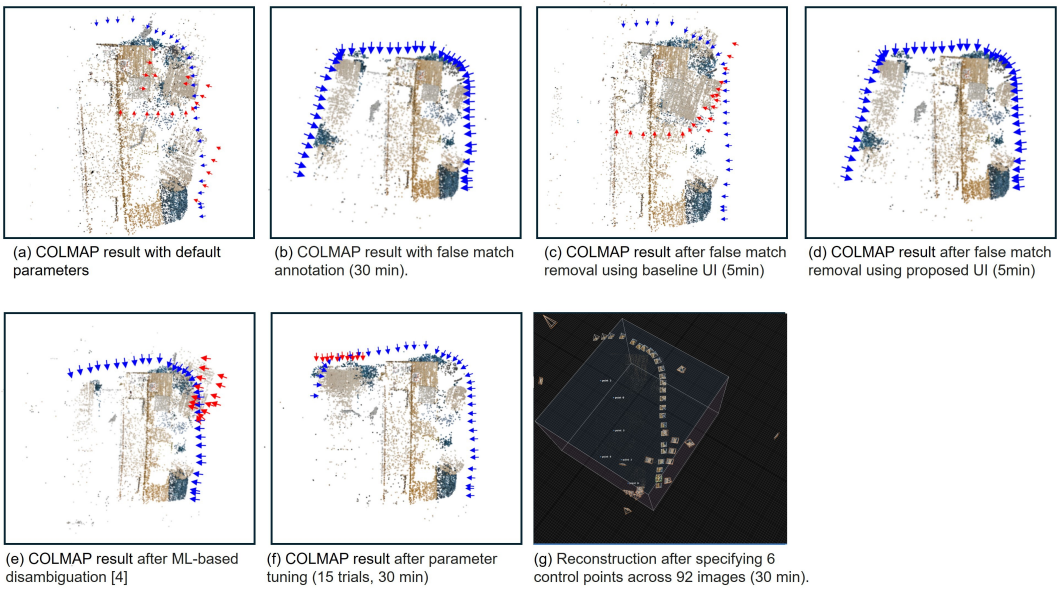

Figure 6: Comparison of SfM results under various conditions. (a) COLMAP result using default parameters, where false matches lead to significant errors in camera path estimation. (b) COLMAP result after manual false match annotation, showing improvement in alignment. (c) Baseline UI with 5 minutes of user intervention, showing partial improvement. (d) Proposed UI with 5 minutes of user intervention, demonstrating an accurate reconstruction. (e) Result of an ML-based disambiguation method [4], which provides a certain level of correction but is still prone to false matches. (f) Result of parameter tuning after 15 trials, indicating that fine-tuning can improve camera path estimation, but is time-intensive. (g) Result with 6 control points and 30 minutes of user intervention, showing limited improvement in reconstruction accuracy.

the image sequence and reconstruction results to identify cameras with incorrectly estimated poses. Upon examining the reconstruction results of this dataset, we found that 21 out of 48 images were incorrectly positioned within the reconstructed 3D scene. These 21 images were further categorized into six subsequences. The user selects images from each subsequence and adjusts their corresponding camera poses. Following this, the user adjusts the camera view frustums and performs reconstruction after removing matches between cameras whose view frustums do not overlap. Figure 6 (d) illustrates the outcome of a 5-minute intervention carried out by a user. This result is from one of the participants of the user study described in Chapter 6.

5.4 Error Correction in Other Methods

We compare our proposed method with three other techniques. The first technique is an automatic approach, while the latter two require user intervention.

5.4.1 Machine Learning-Based Disambiguation. The first method we examine is Doppelgangers, an automatic false match removal approach using a pre-trained classifier [4]. This method uses a binary classifier to determine whether an image pair is a correct match or not. Specifically, after a reconstruction failure with COLMAP, the classifier is applied to all matched image pairs. All pairs with likelihoods below a certain threshold are removed, and reconstruction is attempted again. The advantage of this method is that it does not require user intervention. However, if it fails, other correction methods must be used, or the classifier must be retrained.

We ran the publicly available code and pretrained model provided by the authors on COLMAP's matching results for our dataset, as shown in Figure 6 (e), which resulted in 7 out of 21 incorrect camera positions to be corrected.

5.4.2 Parameter Tuning. We consider parameter tuning to be the most basic form of user intervention. COLMAP offers numerous parameters for various aspects, including camera models, keypoint extraction, image pair matching, and reconstruction. Properly setting these parameters can potentially improve the success rate of reconstruction. However, experimenting with many parameters is time-consuming, and given that users must evaluate the quality of reconstruction results, automating parameter optimization is a complex challenge.

In our experiment, we conducted 15 trials by combining 5 different camera models with 3 matching methods. Using the combination of the OpenCV camera model and Sequential Matcher and parameter adjustment over a span of 30 minutes, the best configuration corrected 11 out of 21 incorrect camera positions (Figure 6 (f)).

5.4.3 Control Points in RealityCapture. Lastly, we evaluate a method implemented in the commercial software RealityCapture, which aims to enhance reconstruction accuracy by providing hints about the positional relationships between images through the definition

of multiple control points captured in three or more camera images. While this approach can effectively improve generally successful reconstructions or assist in merging multiple reconstruction results, it proves to be less efficient when the reconstruction process has failed, resulting in an incorrect point cloud. In our testing, one of the authors spent 30 minutes defining six control points across 92 images from our dataset. Despite this effort, no improvement in the reconstruction was observed (Figure 6 (g)).

6 Representative Examples

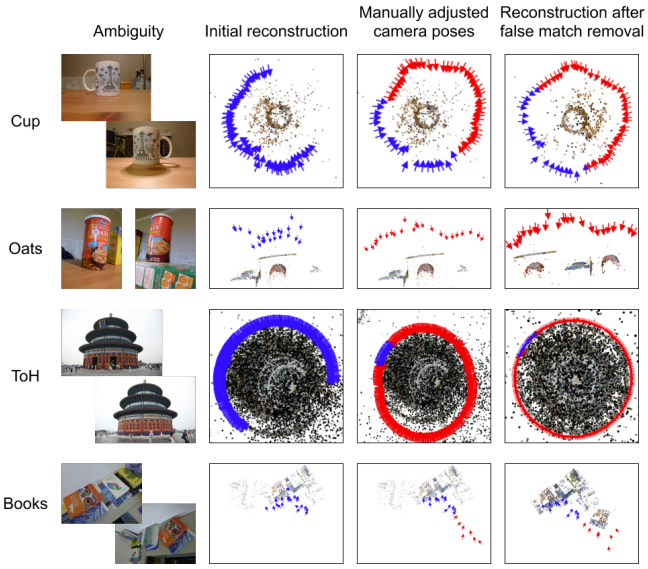

Figure 7: Representative examples of user interventions across three datasets for disambiguation of SfM [34]: cup, oats, ToH, and books from the top. The blue arrows indicate estimated camera poses (left). The red arrows show user-guided estimated poses (center) and fixed camera poses (right), respectively. User intervention successfully corrects false matches, improving the quality of camera pose estimation and point cloud reconstruction across the datasets.

We applied our method to four datasets (cup, oats, ToH, and books) from Yan et al. [34]. Figure 7 shows the outcomes of user-guided reconstruction for each dataset. The cup dataset contains 64 images exhibiting ambiguity because of symmetric patterns on the cup's outer surface. The oats dataset includes 23 images, introducing ambiguity because it features two identical oatmeal boxes placed side by side. The ToH dataset comprises 338 images capturing the Temple of Heaven, a historical building characterized by identical patterns around its surface, which also brings about ambiguity. The books dataset consists of 20 images portraying a scene in which stacked books are arranged in a T-shape on desks. The presence of the same books at both ends of the T-shaped desks introduces ambiguity as well. These types of ambiguities present challenges for COLMAP, but they fall within the capabilities of our proposed method.

Following the procedure outlined in Section 5.2, we first identified sequences of cameras with incorrect poses using the camera image sequence depicted in Figure 4. Subsequently, we manually adjusted the erroneous camera sequences to approximate the correct poses and then removed false matches based on the non-overlapping areas of the user-specified camera view frustums. Finally, we performed reconstruction and obtained corrected camera poses and higher-quality point clouds compared to the initial results from COLMAP for all datasets.

The human intervention process took approximately 2-3 minutes for the cup dataset, where we adjusted 43 out of 64 cameras, and similarly 2-3 minutes for the oats dataset, where all 23 cameras were adjusted, and likewise for the books dataset, with 7 of the 20 cameras being corrected in about the same amount of time. For the ToH dataset, which is considerably larger, adjustments were made to 314 out of 338 cameras, requiring about 30 minutes for the manipulations.

In summary, our method demonstrates that a small amount of user intervention can significantly enhance SfM results in various ambiguous scenarios, including relatively large datasets.

7 User Study

We conducted a user study to evaluate the effectiveness of our proposed method. The task is to improve a 3D reconstruction result that had failed due to false matches. We used our dataset as described in 5.

We recruited eight participants (2 females) with an average age of 28.6 (SD: 6.3) from our institute. All participants had basic knowledge of 3D reconstruction and three had experience in developing 3D reconstruction technologies. This study received approval from our institution's ethics review board.

In the study, we compared our method, where participants adjusted camera poses to guide the removal of false matches (Section 3), with a baseline method where participants manually deleted false matches (Section 5.2). Our hypothesis was that our proposed method would enable participants to remove false matches more efficiently and improve SfM results. To ensure participants' familiarity with the environment captured in the task dataset, the study was conducted within the actual office setting used in the dataset.

7.1 Task

Each participant was tasked with removing false matches from our dataset using both our proposed method and the baseline technique, each for 5 minutes. This relatively short task duration was selected based on pilot tests, which indicated that longer use of the baseline method led to a significant increase in mental workload. Following this 5-minute manipulation period, participants proceeded with the reconstruction, and we evaluated the resulting outputs. Regarding the proposed method, participants could perform the whole operation once, although the proposed framework is the human-in-the-loop method. This is because reconstruction of our dataset took 2-3 minutes, and it was hard to make the second attempt within 5 minutes. The order in which participants used the proposed and baseline methods was counterbalanced to eliminate order effects.

To ensure that the comparison focused solely on the interaction methods for false match removal, we did not include the task of

identifying incorrectly reconstructed cameras. Participants were provided with information on which camera poses were incorrect from the outset. Additionally, we used the same dataset for both conditions, as we estimated that any potential learning effect related to the dataset would likely have minimal impact on the results because participants were given information on cameras with incorrect poses and shooting conditions for the dataset.

7.2 Procedure

The experiments were conducted as follows:

First, we obtained informed consent from each participant. We explained the experiment's purpose, procedure, estimated duration (approximately 30 to 50 minutes), data handling protocols, and the participants' rights.

Next, we provided an overview of the task background, introducing the basic concepts of SfM and explaining how false matches can negatively impact reconstruction quality. Participants were informed that their task involved removing these false matches to enhance the reconstruction results.

We then detailed how the dataset was captured, including the environment of the capture, the camera locations at the time of capture, and the characteristics of the images, to help participants better understand the task at hand.

Following this, we gave instructions on how to use the systems for both the proposed method and the baseline method. We detailed the functions of each system, the specific steps for removing false matches, and important operational considerations.

After the instructions, participants engaged in a practice session for the first method using a practice dataset. This hands-on experience allowed them to familiarize themselves with the system's operations.

Subsequently, participants proceeded to the main session for the first method. They had 5 minutes to remove false matches, after which the reconstruction process was executed. Participants took a short break during the reconstruction.

Following the break, participants began to practice with the second method in the same manner. After ensuring they were comfortable with the system, the main session for the second method began. Participants spent another 5 minutes removing false matches and then executed the reconstruction process.

Upon completing both sessions, participants filled out a questionnaire to provide subjective evaluations and feedback.

7.3 Evaluation Metrics

We used several metrics to assess the accuracy of camera pose estimation and the effectiveness of user guidance on the reconstruction results. Specifically, we calculated the Mean Squared Error (MSE) for translation and the Mean Absolute Error (MAE) for rotation to evaluate the accuracy of reconstructed camera poses. To measure the accuracy of false match removal, we computed recall and precision. Additionally, we evaluated participants' subjective experiences using the User Experience Questionnaire Short Version (UEQ-S).

7.3.1 Error for Translation and Rotation. We calculated the error between the camera poses obtained through user intervention and the ground truth data described in Section 5.2.

Since SfM results do not contain scale and orientation information of the real world, we aligned the ground truth and the users' reconstruction results to a common coordinate system. Specifically, we assumed that the 27 cameras that were correctly reconstructed (out of 48) remained unchanged by user interventions. Using these 27 correctly reconstructed cameras, we estimated the scale and rotation to transform the users' reconstruction results into the coordinate space of the ground truth data. After alignment, we calculated the translation MSE and the rotation MAE for the 21 cameras that had incorrect poses in the initial reconstruction.

The MSE for translation was calculated from the cameras' positional differences between the ground truth and the reconstruction results. The MAE for rotation was determined by computing the differences in angles (degree) using a rotation matrix for each camera pair.

7.3.2 Recall and Precision on False Match Removal . We used the following definitions to calculate recall and precision for false match removal:

- Recall: The ratio of false matches correctly removed by the participants against all the false matches in the ground truth. It evaluates the completeness of the false match removal process.
- Precision: The ratio of false matches correctly removed by the participants against all the matches removed by the participants. It assesses the accuracy of the false match removal.

The mathematical formulations are:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Where:

TP (True Positives): false matches correctly removed both by the participants and in the ground truth.

FN (False Negatives): false matches removed in the ground truth but not removed by the participants.

FP (False Positives): matches removed by the participants but not removed in the ground truth.

We also calculated the F1 score from the recall and precision results.

7.3.3 User Experience Questionnaire (UEQ-S). To assess participants' experience, we used the User Experience Questionnaire Short Version (UEQ-S). Participants responded to eight questions for each method, from which we calculated the Pragmatic Quality, Hedonic Quality, and Overall scores. The evaluation was conducted based on the benchmarks published by UEQ-S [28].

7.4 Results

Table 1 presents the results of our objective metrics. With the exception of P2's precision result, all metrics indicate that for every participant our method was substantially more accurate and produced higher quality results compared to the baseline. There was, however, a failure case with our method for P5, which we discuss in Section 8.2.

Comparing the average camera pose error across all participants, our method achieved a mean translation MSE of 1.4227 with a 95 % confidence interval ranging from 0.5067 to 2.3388. In contrast, the baseline method had a mean translation MSE of 6.4116 with a 95 % confidence interval from 5.7085 to 7.1148. The lack of overlap between these confidence intervals suggests a significant improvement in translation accuracy using the proposed method. Similarly, for rotation MAE, the proposed method's mean was 6.6481 (95 % CI [2.7830, 10.5132]) compared to the baseline's 115.98 (95 % CI [86.70, 145.26]), further indicating a significant enhancement in performance.

Regarding false match removal, the proposed method also demonstrated superior performance. The mean recall for the proposed method was 0.88 with a 95% confidence interval ranging from 0.78 to 0.98, whereas the baseline method had a mean recall of 0.50 (95% CI [0.42, 0.59]). The mean F1 score was 0.93 (95% CI [0.86, 0.97]) for the proposed method, compared to 0.66 (95% CI [0.57, 0.73]) for the baseline. For precision, the proposed method achieved a mean of 0.98 (95% CI [0.96, 1.00]) versus the baseline's 0.93 (95% CI [0.90, 0.97]). Although there is a slight overlap in the 95% confidence intervals for precision, the overall results indicate that the proposed method outperforms the baseline in mismatch deletion accuracy.

Similar to the objective measurements, the subjective ratings of all participants on the UEQ-S were more favorable for our method compared to the baseline (Figure 8). The UEQ-S results of our method reveal high scores in both Pragmatic Quality (e.g., efficient and easy to use) and Hedonic Quality (e.g., interesting and enjoyable), suggesting an overall positive user experience. Pragmatic Quality reflects the system's ability to support users in achieving their tasks efficiently, while Hedonic Quality indicates the system's capability to evoke emotional satisfaction and engagement. Compared to the benchmark provided in the UEQ-S guidelines, the scores of our method fall within the 'Above Average[4]' range, demonstrating that the system performs above average in both dimensions. In contrast, the scores of the baseline were rated as "Bad." These results confirm that our technique offers a superior user experience compared to the baseline. These findings align with our objective to create a system that is not only functional but also enjoyable to use, addressing both task-oriented and emotional needs of users.

8 Discussion and Limitations

8.1 Advantages of Our Method

The experimental results revealed the clear superiority of our user-guided method. Participants were able to significantly improve reconstruction accuracy using our tools and user interface. Additionally, the results confirmed that false matches could be deleted with high precision, demonstrating that users can efficiently enhance reconstruction results within a limited task time. The UEQ-S scores further validated that our method offers a superior interaction experience for detecting false matches.

Moreover, our method proved to be applicable across various scenes. Notably, even in complex scenarios with a large number of images (more than 300), users could easily select and relocate cameras with erroneous subsequences. Furthermore, even when users were unaware of the exact shooting positions, they still managed to improve reconstruction by accurately setting the cameras' relative positions.

[4]'Above Average' indicates that the system's user experience is better than most comparable systems, while 'Bad' reflects significant usability or engagement issues [28].

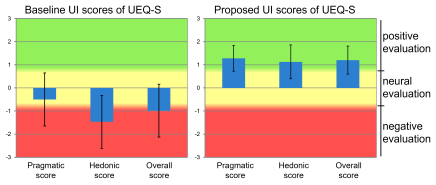

Figure 8: User Experience Questionnaire (UEQ) short version results. The chart shows user feedback on the usability and experience of our technique, measured across two dimensions: Pragmatic quality (left) and Hedonic quality (right). Error bars show the standard deviation, with results suggesting an overall positive experience with our system.

8.2 Limitations of Our Method

In the experiment, an instance occurred with participant P5 where the model became fragmented after manipulation, as illustrated in Figure 9 (a). This likely happened because the deletion of false matches inadvertently removed some correct matches as well. While specifying the matching pairs to be deleted by defining camera poses and their view frustums is possible, users cannot see the results until reconstruction is complete, necessitating a trial-and-error approach. However, given that the proposed method integrates a human-in-the-loop design, users have the flexibility to revert and retry if their initial attempt is unsuccessful.

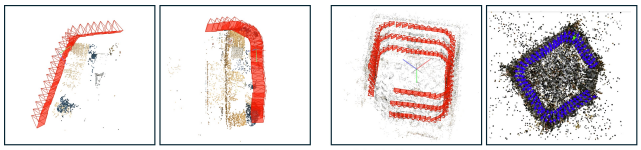

Figure 9: (a) Failure case: The intervention result from P5, where the reconstruction split into two separate models. (b) Challenging case: An indoor dataset from [34], where the vertical hierarchy in the environment makes it difficult to perform effective interventions using the top-down view of our UI.

Additionally, since our interface uses a 2D top-down view, users may find it challenging to adequately visualize complex 3D structures. For instance, as depicted in Figure 9 (b), when multiple cameras are arranged vertically, users must rely solely on the images to determine which layer they are editing. To better address these complex 3D scenes, our UI could be enhanced by incorporating a 3D view, allowing for more precise and intuitive manipulation of camera positions.

Table 1: Translation MSE, Rotation MAE, Recall, Precision, and F1 Score measurements for each participant of the study. Except for P2's Precision score, all metrics show superior results for our method.

Participant	Translation MSE ↓		Rotation MAE (deg) ↓		Recall ↑		Precision ↑		F1 Score ↑	
ID	Baseline	Proposed	Baseline	Proposed	Baseline	Proposed	Baseline	Proposed	Baseline	Proposed
P1	6.41608	**1.56911**	133.267	**10.496**	0.31	**0.62**	0.88	**1.00**	0.46	**0.77**
P2	7.02564	**0.00061**	97.559	**0.668**	0.41	**1.00**	**0.96**	0.94	0.57	**0.97**
P3	5.36414	**2.09507**	76.639	**8.523**	0.57	**0.92**	0.90	**0.99**	0.70	**0.95**
P4	6.04193	**2.09787**	61.630	**8.762**	0.56	**0.83**	0.99	**1.00**	0.72	**0.91**
P5	5.33908	—	168.106	—	0.51	**0.92**	0.95	**0.98**	0.66	**0.95**
P6	6.87383	**2.10698**	121.426	**8.862**	0.61	**0.92**	0.97	**0.98**	0.75	**0.95**
P7	7.81761	**2.08914**	136.051	**8.678**	0.56	**0.82**	0.95	**1.00**	0.71	**0.90**
P8	6.41488	**0.00048**	133.221	**0.547**	0.50	**1.00**	0.88	**0.94**	0.64	**0.97**
Average	6.41165	**1.42275**	115.987	**6.648**	0.50	**0.88**	0.93	**0.98**	0.66	**0.93**

Although the proposed method relies on user interventions based on domain knowledge and observations of captured scenes, it remains unclear the minimal information that is necessary for effective interventions. Consequently, we need further investigation to determine under what condition the SfM correction of captured scenes is feasible. Furthermore, overlaying a shooting route on a 2D map along with reconstructed camera positions and their captured order could help complement users' lack of knowledge, which we consider one of the future works to improve our user interface.

8.3 Combining with Other 3D Reconstruction Correction Techniques

We do not claim that our method is the optimal choice in all scenarios. In our experiment, annotation on around 50 images could be completed in about five minutes, whereas annotating a dataset of roughly 300 images took over 30 minutes. These observations suggest that our user interface is limited to datasets consisting of a few hundred images. Nevertheless, even when dealing with thousands to tens of thousands of images, our interface can still be valuable for correcting specific (sub)sequences within the large-scale data that are incorrect. Once these small-scale corrections are made, the results can be integrated into the larger dataset using techniques such as automatic merging [6] or control points in RealityCapture.

Additionally, our method can complement other false match correction techniques. For example, false matches that are not identified by automatic algorithms can be effectively removed using our approach. Furthermore, global structure-from-motion methods can be applied to the database once it has been corrected using our method. We believe that user intervention via our technique is particularly effective when existing automatic correction methods fall short.

9 Conclusion

In this paper, we proposed a user-guided method for correcting false matches in Structure-from-Motion (SfM) reconstructions by leveraging users' knowledge of the captured environment. This approach enables users to efficiently identify and relocate incorrectly reconstructed cameras, adjust their view frustums, and update the matching database to improve reconstruction results. The user interface (UI) we developed facilitates this process, allowing users to effectively perform necessary interventions. Experimental results demonstrated that our method outperforms manual removal of false camera matches both in terms of reconstruction accuracy and user experience. However, there were a few challenges, such as instances where correct matches were inadvertently deleted, causing a split in the model. We plan to address these issues in future work. Enhancing the UI to include both 2D and 3D views could help users better visualize and manipulate complex 3D camera arrangements, especially in scenes with significant vertical variations. Additionally, integrating our method with automatic false match correction algorithms may improve scalability and efficiency when dealing with large-scale datasets. By combining the strengths of interactive and automatic approaches, we aim to further enhance the robustness and applicability of SfM reconstruction workflows.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. 2011. Building rome in a day. *Commun. ACM* 54, 10 (2011), 105–112.

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. *ICCV* (2021).

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. *ICCV* (2023).

[4] Ruojin Cai, Joseph Tung, Qianqian Wang, Hadar Averbuch-Elor, Bharath Hariharan, and Noah Snavely. 2023. Doppelgangers: Learning to Disambiguate Images of Similar Structures. *ICCV*.

[5] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. 2024. HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression. In *European Conference on Computer Vision*.

[6] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. 2015. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5556–5565.

[7] Zhaopeng Cui and Ping Tan. 2015. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE International Conference on Computer Vision*. 864–872.

[8] Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 2024. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[9] Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

[10] Yi He, Xi Yang, Chia-Ming Chang, Haoran Xie, and Takeo Igarashi. 2023. Efficient Human-in-the-loop System for Guiding DNNs Attention. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (Sydney, NSW, Australia) *(IUI '23)*. Association for Computing Machinery, New York, NY, USA, 294–306. https://doi.org/10.1145/3581641.3584074

[11] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. 2014. Correcting for duplicate scene structure in sparse 3D reconstruction. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings,*

Part IV 13. Springer, 780–795.

[12] Keita Higuchi, Taiyo Mizuhashi, Fabrice Matulic, and Takeo Igarashi. 2023. Interactive Generation of Image Variations for Copy-Paste Data Augmentation. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 181, 7 pages. https://doi.org/10.1145/3544549.3585856

[13] Geoff Holmes, Eibe Frank, Dale Fletcher, and Corey Sterling. 2022. Efficiently correcting machine learning: considering the role of example ordering in human-in-the-loop training of image classification models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 584–593. https://doi.org/10.1145/3490099.3511110

[14] Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Rajbir Kataria, Joseph DeGol, and Derek Hoiem. 2020. Improving Structure from Motion with Reliable Resectioning. In *2020 International Conference on 3D Vision (3DV)*. 41–50. https://doi.org/10.1109/3DV50981.2020.00014

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[17] Michael Laielli, James Smith, Giscard Biamby, Trevor Darrell, and Bjoern Hartmann. 2019. LabelAR: A Spatial Guidance Interface for Fast Computer Vision Image Collection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 987–998. https://doi.org/10.1145/3332165.3347927

[18] Gierad Laput, Walter S. Lasecki, Jason Wiese, Robert Xiao, Jeffrey P. Bigham, and Chris Harrison. 2015. Zensors: Adaptive, Rapidly Deployable, Human-Intelligent Sensor Feeds. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1935–1944. https://doi.org/10.1145/2702123.2702416

[19] Jingyuan Liu, Li-Yi Wei, Ariel Shamir, and Takeo Igarashi. 2024. iPose: Interactive Human Pose Reconstruction from Video. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 945, 14 pages. https://doi.org/10.1145/3613904.3641944

[20] Zihua Liu, Hiroki Sakuma, and Masatoshi Okutomi. 2024. VSRD: Instance-Aware Volumetric Silhouette Rendering for Weakly Supervised 3D Object Detection. arXiv:2404.00149 [cs.CV]

[21] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20654–20664.

[22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

[24] Yuto Nakashima, Mingzhe Yang, and Yukino Baba. 2024. SwipeGANSpace: Swipe-to-Compare Image Generation via Efficient Latent Space Exploration. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (Greenville, SC, USA) *(IUI '24)*. Association for Computing Machinery, New York, NY, USA, 675–685. https://doi.org/10.1145/3640543.3645141

[25] Kotaro Oomori, Wataru Kawabe, Fabrice Matulic, Takeo Igarashi, and Keita Higuchi. 2023. Interactive 3D Annotation of Objects in Moving Videos from Sparse Multi-view Frames. *Proc. ACM Hum.-Comput. Interact.* 7, ISS, Article 440 (Nov. 2023), 18 pages. https://doi.org/10.1145/3626476

[26] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. 2024. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*.

[27] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4104–4113.

[28] Martin Schrepp, Andreas Hinderks, et al. 2017. Design and evaluation of a short version of the user experience questionnaire (UEQ-S). (2017).

[29] Carlo Tomasi and Takeo Kanade. 1992. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision* 9 (1992), 137–154.

[30] Kathleen Tuite, Noah Snavely, Dun-yu Hsiao, Nadine Tabing, and Zoran Popovic. 2011. PhotoCity: training experts at large-scale image acquisition through a competitive game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 1383–1392. https://doi.org/10.1145/1978942.1979146

[31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2018. Deep Image Prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[32] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS* (2021).

[33] Thomas Weber, Heinrich Hußmann, Zhiwei Han, Stefan Matthes, and Yuanting Liu. 2020. Draw with me: human-in-the-loop for image restoration. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 243–253. https://doi.org/10.1145/3377325.3377509

[34] Qingan Yan, Long Yang, Ling Zhang, and Chunxia Xiao. 2017. Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3836–3844.

[35] Ziyi Yang, Xinyu Gao, Yangtian Sun, Yihua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. 2024. Spec-Gaussian: Anisotropic View-Dependent Appearance for 3D Gaussian Splatting. *arXiv preprint arXiv:2402.15870* (2024).

[36] Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. 2024. Cogs: Controllable gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21624–21633.

[37] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. 2024. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19447–19456.

[38] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv:2010.07492* (2020).

[39] Mingfang Zhang, Jinglu Wang, Xiao Li, Yifei Huang, Yoichi Sato, and Yan Lu. 2023. Structural Multiplane Image: Bridging Neural View Synthesis and 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16707–16716.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009