IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 May 2024, accepted 22 July 2024, date of publication 29 July 2024, date of current version 6 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434584

RESEARCH ARTICLE

Evaluating Image-Based Interactive 3D Modeling Tools

ARSLAN SIDDIQUE[1,2], **PAOLO CIGNONI**[1], **MASSIMILIANO CORSINI**[1], **AND FRANCESCO BANTERLE**[1]

[1]Visual Computing Laboratory, ISTI-CNR, 56124 Pisa, Italy
[2]Department of Computer Science, University of Pisa, 56127 Pisa, Italy

Corresponding author: Francesco Banterle (francesco.banterle@isti.cnr.it)

This work was supported in part by European Union's Horizon 2020 Programme EVOCATION: Advanced Visual and Geometric Computing for 3D Capture, Display, and Fabrication under Grant 813170; and in part by the Electric Power Research Institute, Washington, DC, USA.

ABSTRACT Structure from Motion (SfM) is a computer vision technique used to reconstruct three-dimensional (3D) structures from a series of two-dimensional (2D) images or video frames. However, SfM tools struggle with transparent objects, reflective surfaces, and low-resolution frames. In such situations, image-based interactive 3D modeling software packages are employed to model 3D objects and measure dimensions. Our contributions to this work are twofold. First, we have introduced new tools to improve 3D modeling software packages; such tools are aimed at easing the workload for users. Second, we have conducted a comprehensive user study to evaluate the efficacy of popular 3d modeling software packages. The task is to measure certain dimensions for which ground truth measurements are already known. A relative error is calculated for every measurement. The evaluation of each software tool is done through survey form, event logs, and measurement relative error. The results of this user study clearly show that our approach to 3D modeling using multiple images has a lower relative error and produces higher quality 3D models than other software packages. In addition, it shows our new tools reduce the required time for completing a task.

INDEX TERMS User-assisted 3D reconstruction, interactive 3D modeling, computer graphics, image based-3D reconstruction, structure-from-motion.

I. INTRODUCTION

The recovery of a 3D model from a collection of images has been a long-standing goal of computer vision and graphics communities. Structure-from-motion (SfM) [1] is a popular technique for acquiring 3D models from the real world using photographs. To have high-quality results, we need a studio setup, which consists of a series of high-resolution calibrated cameras capturing a fixed volume from different views. However, establishing this type of setup is very expensive due to the requirement of many high-resolution cameras, light sources, and synchronization electronics. Due to economic and logistic constraints, we need to capture images using a single uncalibrated camera that moves in the scene to generate 3D models. Various software applications exist for performing SfM tasks, and they have distinct features and functionalities. Some of these software applications are available for free, such as COLMAP [2], MicMac [3], Open-MVS [4], and so on, while others are commercial software applications such as Metashape,[1] RealityCapture,[2] etc. These fully automatic software tools use keypoint detection and matching algorithms to estimate 3D structures from images. Keypoint detection and matching algorithms struggle with transparent objects, reflective surfaces, dark environments, low-resolution content, motion blur, or compression artifacts (Figure 1).

Video-based SfM poses more challenges such as motion blur, degeneracies in camera motion, and very large computational costs due to unnecessary frames. We also have

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Wei.

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

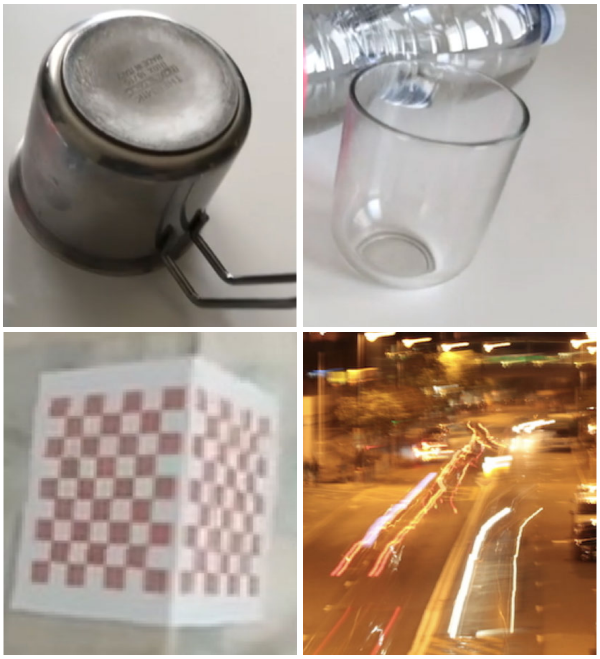


FIGURE 1. Examples of difficult cases for standard SfM pipelines such as reflective, transparent, compressed, and motion-blurred images. Keypoint detection and matching algorithms struggle in such cases, and this results in low-quality 3D models with automatic SfM tools.

extra constraints when trying to recover geometry when monitoring industrial plants. Typically, utility companies have legacy video cameras capturing videos at low-resolution and low lighting conditions (e.g., dim light sources or torch-light mounted on the camera). These videos are generally compressed aggressively, leading to severe compression artifacts such as ringing, blocking, etc. When such low-quality and featureless video frames are employed as inputs for SfM software a very limited number of features and correspondences are detected accurately. This causes a low-quality 3D output. These issues can be reduced by performing user-assisted 3D reconstruction. In this process, the user provides different interventions in SfM and image-based 3D modeling processes. User-assisted 3D reconstruction decreases computational cost and also reduces the number of input images required for 3D reconstruction. We have developed MoReLab [5] to perform user-assisted 3D reconstruction on uncalibrated camera videos or a set of images with different viewpoints.

In this work, we propose the following key contributions that advance our previous work in image-based modeling:

- **Extensions**: we have enriched our tool MoReLab with new features that reduce the workload for users (described in Section III). Additionally, we think that such features may help the design of next-gen image-based modeling tools.
- **User study**: we provide a comprehensive study where participants were asked to do modeling and measurement tasks on MoReLab and other user-assisted 3D reconstruction software tools such as 3-Sweep [6] and Photomodeler[3] (reported in Section IV). This user study is useful to assess the performance and the capability of such software tools in supporting the user in getting accurate measures from images and validating design choices.

II. RELATED WORK

User-assisted 3D reconstruction is a well-studied research area with many existing interfaces and approaches.

A. SFM-BASED INTERFACES

Many user interfaces [7], [8], [9], [10], [11], [12] employ SfM as an automatic pre-processing stage to obtain camera poses and an initial set of sparse 3D points. VideoTrace [7] overlays 3D points on a video frame. Then, the user traces a set of line segments to model a polygonal face. By drawing a few polygons, the shape of the object can be modeled, and a realistic 3D model can be obtained. Sinha et al. [8] computes sparse 3D data in such a way that lines and vanishing points are estimated in the scene as well. Then, the user sketches 2D outlines of the planar sections of the scene. The system computes a 3D planar polygon from 2D outline sketches. Nearby 3D SfM points and vanishing directions are used to compute the 3D plane normal and depth. Few such interactions enable the user to build a piecewise planar 3D model of the scene. The user interface developed by Habbecke and Kobbelt [9] consists of a 2D image viewer and a 3D object viewer. The user paints brush strokes on the 2D image and the reconstruction algorithm computes the corresponding 3D surface mesh. As the user continues modeling, the system continues to build 3D surface patches and guide the surface reconstruction algorithm. Doron et al. [10] utilize brush stroke-based user annotations as smoothness, discontinuity, and depth ordering constraints to guide multi-view stereo algorithms. Their experiments show that a user-guided multi-view stereo algorithm increases the accuracy of the reconstructed depth map. The interface, developed by Baldacci et al. [13], also takes images and sparse 3D points as inputs. The interface allows the user to indicate background and foreground regions using brush strokes. The user can also provide localized hints about the curvature of the surface. These hints serve as constraints for the reconstruction of smooth surfaces. Xu et al. [11] developed a system for interactive 3D modeling and stochastic motion parameter estimation. For the 3D modeling step, stroked-based sweep modeling is used to build a 3D model of the equipment on calibrated 2D images. For the stochastic motion parameter estimation, a video clip recording of the working mechanism of the equipment is used to recover motion parameters. Rasmuson et al. [12] presented a modeling system in which the user marks image points and the system builds a quad connecting marked points on high-quality images. After drawing a few quads, a global optimization algorithm builds the final 3D model; objects

[3] https://www.photomodeler.com/

are modeled as a combination of large number of quads, which can be a tedious task. In all these systems, SfM struggles to obtain accurate camera poses and 3D points on featureless and low-resolution frames and videos. Therefore, such interactive systems are not useful for frames lacking discernible features due to the inaccurate pre-processing stage of SfM. Quan et al. [14] developed an interface for image-based plant modeling. After SfM, the user provides foreground and background hints for the leaves and branches of the plant. The user also selects the plant model. Segmented images, 3D sparse points, and plant priors are used to develop the 3D model for the specific plant.

B. GEOMETRIC CONSTRAINT-BASED INTERFACES

Many interfaces require the user to mark feature correspondences and do not use SfM as a pre-processing stage. In Photomodeler, the creation of 3D models is facilitated through the annotation of structures in one or more images, necessitating manual input of measurements from both the scene and the cameras. Debevec et al. [15] presented an approach combining geometry-based and image-based modeling approaches. Their approach consists of two components. The first component is a photogrammetric modeling, which exploits user-provided constraints of architectural scenes. The second component is a model-based stereo algorithm, which computes depth from image pairs. The system developed by Wilczkowiak et al. [16] requires the corners of primitives to be marked manually by the user. They use parallelepipeds as scene primitives and exploit the duality between the shape of parallelepipeds and the internal parameters of a camera.

Single-view reconstruction also does not utilize SfM because it cannot be computed from a single image. There have been some interfaces developed to tackle single-view reconstruction as well. For example, Toppe et al. [17] builds an object silhouette based on user scribbles in an image. Implicit surface representation and a transparent optimality criterion are used to minimize weighted surface area for a fixed volume. This leads to smooth surfaces and a high-quality 3D model. 3-Sweep [6] is a user-friendly and interactive tool designed for extracting 3D models from a single photograph. Upon loading a photo into the tool, 3-Sweep calculates the boundary contour. After defining the boundary contour, the user chooses the model shape and outlines the desired object with three brush strokes–one for each dimension of the image. The interface employs foreground texture segmentation to swiftly generate an editable 3D mesh object, which can be translated, rotated, or scaled.

C. DEEP LEARNING-BASED 3D RECONSTRUCTION

There has been a sharp rise in deep learning-based approaches for 3D reconstruction. RealPoint3D [18] is an efficient generation network to predict a 3D point cloud from an image containing a single object. The input to this network is a single object image and the nearest shape retrieval from ShapeNet [19]. The two encoders are integrated adaptively according to their information integrity, followed by the decoder to obtain fine-grained point clouds. 3D-ReConstnet [20] is an end-to-end neural network that predicts a point cloud from a single 2D image. 3D-ReConstnet uses a residual network to extract features of a 2D image and exploits Gaussian probability distribution to deal with self-occluded parts of the object. It is a memory-efficient multi-view reconstruction network with a pyramid encoder-decoder structure, searching for depth correspondences incrementally. This network encodes the image features to a much smaller resolution to substantially reduce memory requirements.

Recent works focus on neural implicit representations to reconstruct 3D models from unstructured Internet photo collections. Zhang et al. [21] utilizes a neural shape representation that deforms a unit sphere to capture the geometry of the object and a neural vector field to represent surface texture. This method relies on neural networks to learn bidirectional surface reflectance functions (BRDFs), which factorize view-dependent appearance into components such as environmental illumination, diffuse color, and specular highlights. Sun et al. [22] combine hybrid voxel- and surface-guided sampling techniques to improve ray sampling efficiency around surfaces, resulting in higher reconstruction quality. NeuS [23] involves the use of a neural network to represent a signed distance function (SDF), which defines the surfaces to be reconstructed. This neural implicit surface representation is trained using a novel volume rendering approach. Specifically, the SDF is encoded by a fully connected neural network, which is optimized through the rendering process to accurately reconstruct the 3D surfaces from multi-view 2D images.

Deep-learning based reconstruction networks perform very well on high-resolution images. Our focus is on low-quality industrial videos and frames. The performance of deep learning-based reconstruction networks on such videos is not satisfactory. Hence, we have to rely on interactive 3D modeling for our low-quality dataset videos. Geometric constraint-based interfaces are generally used to model low-quality frames and videos. The reason is that the pre-processing stage of SfM is prone to large errors in low-quality frames and videos.

In this paper, we will describe the improvements done in MoReLab. A user study has been conducted to evaluate the effectiveness of MoReLab, 3-Sweep, and Photomodeler. The reason to assess the performance of these three software programs relates to the wide availability of these software programs.

III. MORELAB

MoReLab [5] has been developed to model objects in low-resolution and low-quality videos, which are common in industrial settings. Instead of automatic feature detection and matching, the user marks feature correspondences across frames on the interface of MoReLab. Bundle adjustment [24] uses these feature locations to simultaneously obtain camera poses and 3D sparse points.

We achieve this by minimizing the reprojection error between input 2D locations and projected 2D locations of 3D points on the image. Let us suppose that n 3D points can be observed in m views. Let $\mathbf{x}_{ij}$ be the $i-$th feature location on the $j-$th image, $\mathbf{X}_i$ is the corresponding $i-$th 3D point, and $\mathbf{C}_j$ is camera parameters corresponding to the $j-$th image, then the objective function for bundle adjustment can be defined as:

$$\arg\min_{\mathbf{X}_i,\mathbf{C}_j} \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij}\, d\left(\Pi\left(\mathbf{X}_i, \mathbf{C}_j\right), \mathbf{x}_{ij}\right), \tag{1}$$

where, $\Pi(\mathbf{X}_i, \mathbf{C}_j)$ is the projection of $i-$th 3D point on $j-$th image. $d\big(\Pi(\mathbf{X}_i, \mathbf{C}_j), \mathbf{x}_{ij}\big)$ is the Euclidean distance between the projected point and $\mathbf{x}_{ij}$. $\mathbf{b}_{ij}$ is a binary variable that equals 1 if the i-th feature is visible on the j-th image and 0 otherwise.

Geometric shape priors such as rectangles, cylinders, curved cylinders, etc. are used to model various parts of the frame. For a detailed discussion of these tools, we remind the reader of the original paper [5].

Here, we will describe the new tools and capabilities of MoReLab since this work has the aim to evaluate the performance of this software, and these new capabilities have been designed to speed up the image-based interactive 3D modeling process. The new functionalities include automatic feature detection and matching using SuperGlue [25], another feature tool for quickly adding features, more robust placement of features from one frame to another, an anchor tool, and a guiding lines tool. The interface of MoReLab has been shown in Figure 2. This overview will also serve to provide ideas to make next-gen image-based modeling tools more performing and comfortable for the user.

A. AUTOMATIC FEATURE MATCHING

MoReLab has been developed to focus on 3D reconstruction of low-quality images and videos. Hence, features needed to be always manually marked in the previous version of MoReLab. However, there may be situations in which a high-quality video is loaded in MoReLab or the user may wish to complete the modeling process very quickly. In such situations, automatic feature matching becomes a very useful function. While there are many existing techniques for feature detection and matching, SuperGlue stands out as a state-of-the-art and open-source method with very high accuracy.

SuperGlue is a graph neural network that jointly finds correspondences and rejects unmatchable points. The architecture of SuperGlue consists of two parts. The first component is an attentional graph neural network. First, this attentional graph neural network aggregates the position vector and visual descriptor of a keypoint into a single layer. Then, it uses self- and cross-attention layers to develop a powerful feature descriptor of the keypoint. The second component is an optimal matching layer, which utilizes the Sinkhorn algorithm [26] to create a matching score matrix. SuperGlue achieved superior results compared to handcrafted feature matches and learned inlier classifiers in tasks of homography estimation, indoor pose estimation, and outdoor pose estimation. We have used their model weights for the indoor pose estimation task, trained on ScanNet dataset [27]. SuperGlue performs real-time on a modern GPU and also allows us to increase or decrease the number of matching features by changing the matching threshold.

While existing 3D modeling interfaces either do automatic feature matching or manual feature addition, the new version of MoReLab provides both options to the user. We also added a Selection tool that allows the user to draw a rectangular selection around the area of interest in the frame. Hence, MoreLab allows the user to perform SuperGlue feature matching on any part of the frame.

B. STEP FEATURE TOOL

The step feature tool is useful for situations where the camera motion is so large that features of one frame are no longer visible in another frame. This tool takes the maximum label from all features of other frames and adds the next feature. As an example, if there are 15 features on one image and none of these 15 features is present on another frame, then this tool will directly add feature number 16 on the new frame, which is the desired functionality. With the regular feature tool, feature number 1 will be added to the new frame if there is no previous feature on the new frame. Since feature 1 is not visible on the new frame, it will be deleted. Similarly, 14 more features will be added and deleted with the regular feature tool. Finally, after adding and deleting a total of 15 features, feature number 16 will be added. The reason is that the regular feature tool always starts from feature 1 on a new frame. On the other hand, the step feature tool directly allows the participant to add the new feature.

C. ANCHOR TOOL

The anchor tool moves any primitive to a 3D point. This tool is useful for the proper alignment of a 3D primitive model. When using the anchor tool, a user needs to click on a 3D point P_1. Then, the user clicks on a point P_2 on a 3D primitive. The vector is computed as $P_{12} = P_1$ - P_2. The entire 3D primitive shape is translated along this vector.

D. GUIDING LINES TOOL

The guiding lines tool helps a user in adding features by creating an epipolar line where a matching feature in another image may lie in the current image. This tool is based on the Fundamental matrix, $\mathbf{F}$. The computation of the Fundamental matrix is achieved using feature locations of the current image and the last image using the 8-point algorithm [28]. Suppose we have n 2D correspondences between one image, I_1, and another one, I_2. Let's define the i-th correspondence as the couple $< \mathbf{p}_1^l; \mathbf{p}_2^l >$, where $\mathbf{p}_1^l$ is a 2D point in I_1 and $\mathbf{p}_2^l$ is a 2D

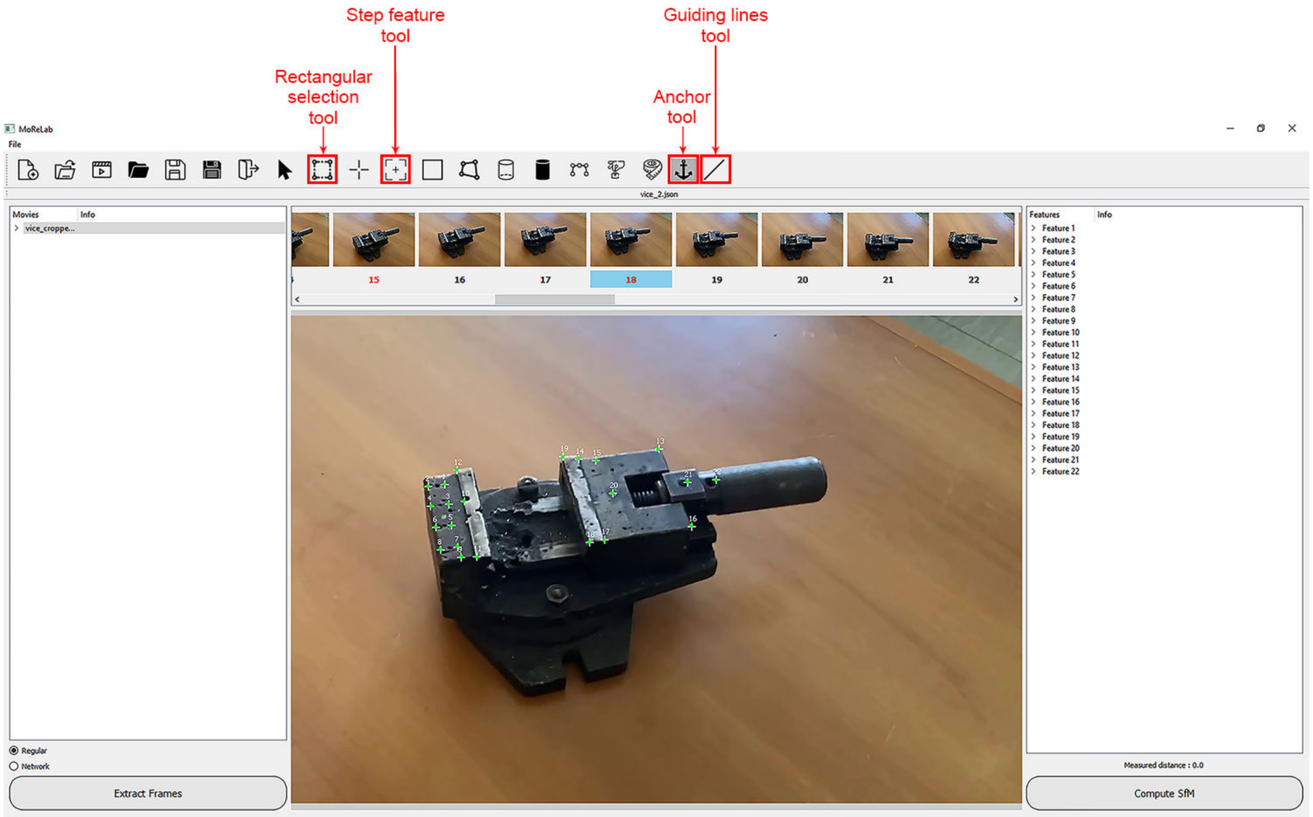


FIGURE 2. The Graphical User Interface of MoReLab. The tools highlighted are the new one w.r.t the original tool, and they are the rectangular selection tool for automatic feature detection, the step feature tool, the anchor tool, and the guiding lines tool.

point in I_2. Given the Lounget-Higgings equation, we know that:

$$\mathbf{p}_2^{l,\top}\mathbf{F}\cdot\mathbf{p}_1^{l} = 0. \tag{2}$$

Given Equation 2, we can define a linear system of the form $\mathbf{A}\mathbf{x} = 0$ using the n correspondences. This system can be solved with the singular value decomposition.

Since we can only know $\mathbf{F}$ up to scale, we need a minimum of eight constraints to determine $\mathbf{F}$. A higher value of n would lead to a better estimate of the solution because it reduces the effects of noisy measurements.

Once $\mathbf{F}$ is computed, the user can now click on any feature on the last image and the corresponding epipolar line will be displayed on the current image. This epipolar line affectionately called *guiding line*, assists the user in marking a feature on the current image. Figure 3 shows this process on the project done by a participant. After selecting the guiding line tool, the user clicks on frame 1, selects the feature tool, and adds a new feature 27 highlighted in blue color. The corresponding guiding line is displayed in frame 2. Once the user will add feature 27 on frame 2, the guiding line will disappear.

IV. USER STUDY

We conducted a user study to evaluate different image-based interactive 3D modeling tools. The tools being evaluated are MoReLab, 3-Sweep (which uses a single image as input), and Photomodeler. MoReLab and 3-Sweep are free-to-use tools, with simple tutorials; while Photomodeler is a commercial software tool.

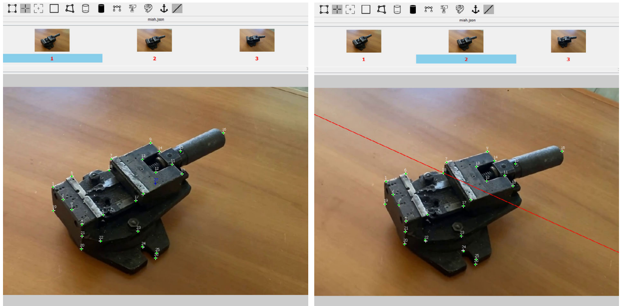


FIGURE 3. An example of guiding lines. The left figure shows a newly added feature (blue colored) on the last frame (frame 1) added after the selection of the anchor tool, and the right figure shows the corresponding guiding line (red colored) or epipolar line displayed on the new frame (frame 2).

A. PROCEDURE

The user study was conducted in two sessions, spaced 3–5 days apart (depending on the availability of a participant) to reduce the time spent in one single session and minimize the influence of the first session on the second one. Participants worked on MoReLab in one session which lasted one hour and twenty minutes; while participants worked on 3-Sweep and Photomodeler in another session which lasted two hours in total. 21 volunteers participated in our user study. we randomly assigned half of the participants to group A, who conducted MoReLab session first and worked on 3-Sweep and Photomodeler in the second session. The remaining half of the participants were assigned group B, who did the reverse.

For each software, there was a learning phase and an experiment phase. In the learning phase, participants watched a tutorial and could ask questions if something was not clear. The learning phase lasted for twenty minutes for each software. In the experiment phase, participants were asked to perform the modeling tasks, and they were also allowed to re-watch the tutorial during this phase.

B. EVALUATION

The evaluation of the software programs was done in three different ways. The first evaluation method is self-reporting, which is an easy and fast approach, commonly done by filling out surveys. We designed four questionnaires (Q1, Q2, Q3, Q4). Questionnaire Q1 gathers data about the participant; while, Questionnaires Q2, Q3, and Q4 allow the participant to evaluate MoReLab, 3-Sweep, and Photomodeler respectively. The second evaluation method is the collection of event logs during the session. We modified the publicly available code of MoReLab[4] to collect a log file for each session. The log file collects time stamps of different user interactions with MoReLab. The log file can be analyzed later to understand usage patterns. Since 3-Sweep and Photomodeler do not provide open-source code, we could not modify their code to generate log files for their usage. The third method for evaluation is through the tasks of getting physical measurements.

C. TASKS

The task was to measure a dimension when a measurement for another length had been provided. Since these software programs possess different tools and capabilities for 3D modeling, it is not possible to evaluate all functionalities in approximately 1 hour. Evaluating all capabilities would require many hours of user study for a single participant. The main motivation behind the development of MoReLab is to measure the dimensions of equipment in different scenarios. Hence, we focus our user study on the task of calibration and measurements. Given a ground truth length M_g and measurement M_e obtained from the estimated 3D model, the relative error E is calculated as:

$$E = 100 \times \left(\frac{|M_e - M_g|}{M_g} \right). \tag{3}$$

Each participant worked on frames of two videos. Video 1 is the video of the vice tool captured in an indoor scenario (our laboratory). Figure 4 shows a frame of this video labeled with known and unknown measurements. The task 1 was to obtain the unknown measurement shown in Figure 4 for the video 1. This video was captured to demonstrate an example of planar modeling and planar measurement.

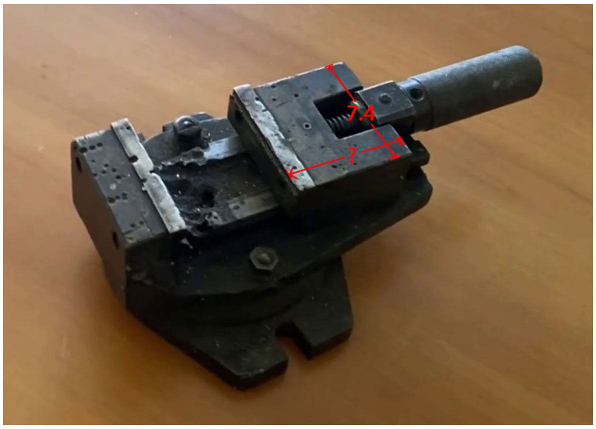


FIGURE 4. An example showing known and unknown measurements for video 1. The known measurement is 7.4 centimeters and task 1 is to find this unknown measurement.

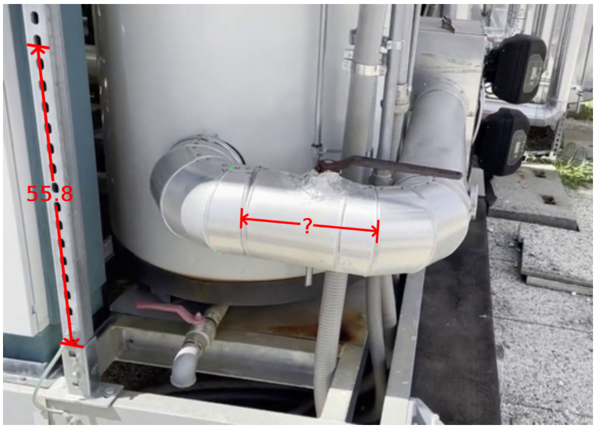


FIGURE 5. An example of known and unknown measurements for the video 2. The known measurement is 55.8 centimeters and task 2 is to find this unknown measurement on the pipe.

Video 2 is an outdoor scenario, a video captured on the roof of our research institute. Figure 5 shows a frame of this video labeled with known and unknown measurements, demonstrating the task 2. This video was captured to demonstrate an example of planar as well as curved surface modeling and measurement.

V. RESULTS

In this section, we present the results obtained by conducting the user study. The data from questionnaires has been analyzed to identify systematic and random errors. Data from questionnaires Q2, Q3, and Q4, is aggregated by making a box plot for each question. Appendix A presents all questions asked in questionnaires. In total, 21 volunteers participated in the user study. Measurements obtained by participants have been utilized to evaluate software programs. Log analysis of event logs collected during the user study of MoReLab, helps to identify usage patterns of different participants.

[4] https://github.com/cnr-isti-vclab/MoReLab

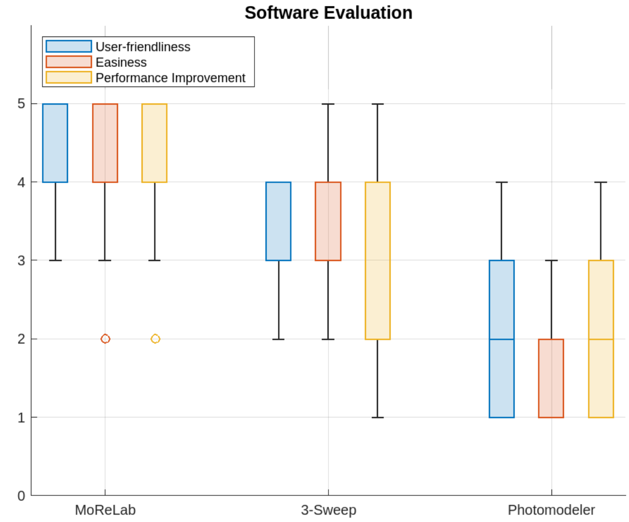


FIGURE 6. Box plots of evaluation scores of tested software tools. In the box plots, the box represents the interquartile range (IQR), which is the middle 50% of the data. The lower and upper edges of the box correspond to the first (Q1) and third (Q3) quartiles, respectively. The median value is indicated by a red line in the box. The whiskers extend from the box to indicate the range of the data. They extend to 1.5 times the IQR. Data points beyond the whiskers are considered outliers and have been indicated with a red plus sign.

A. PARTICIPANT INFORMATION

Questionnaire Q1 provides information about the participant. Out of 21 participants, 18 identify as males and 3 identify as females. Ages are distributed with 5 Gen Z (19-26), 13 Millennials (27-42), and 3 Gen X (43-58) at the time of the user study. In terms of the highest educational degree, 10 participants have a Master's degree, 7 have a doctoral degree, and 4 have completed a Bachelor's. In response to the question about familiarity with structure from motion, 11 participants claim to be familiar, and the remaining 10 participants claim not to be familiar. Hence, we can consider 11 participants as *experienced* who are familiar with this field of research. The remaining 10 participants can be considered *novices*, who do not have prior experience with this field of research. However, none of the participants had worked on any of the three software tools. Software tools were new for all participants, and all participants had to watch a tutorial to learn the usage of each software program.

B. SOFTWARE EVALUATION

Questionnaires Q2, Q3, and Q4 give the evaluation scores given by participants for software tools. Three questions were asked to each participant to evaluate user-friendliness, easiness, and perceived performance improvement from one video to another for a software program. The score can range from 1 to 5. Figure 6 illustrates box plots of evaluation scores for each tested software. For MoReLab, the interquartile range is (4; 5) for all properties. The median values are 4, 4, and 5 for user-friendliness, easiness, and performance improvement respectively. This indicates a very high level of user satisfaction with MoReLab since most scores are high. For 3-Sweep, interquartile ranges are (3; 4) with a median value of 3 for user-friendliness and easiness and (2; 4) with a median value of 4. Since 3-Sweep is simple and easy to learn, the ability to improve is very low through experience and hence, the IQR for 3-Sweep is (2; 4). For Photomodeler, the median value is 2 for all properties. However, IQR ranges are (1; 3), (1; 2), and (1; 3) for user-friendliness, easiness, and performance improvement. This indicates an extremely low level of satisfaction with Photomodeler.

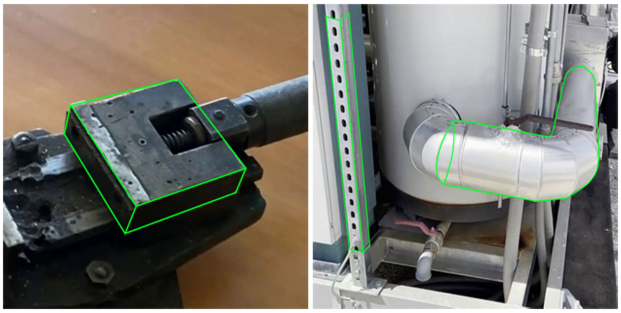


FIGURE 7. Figure on the left and right illustrate modeling tasks for video 1 and video 2, respectively.

MoReLab achieves a high score as compared to other software tools for all aspects; i.e., user-friendliness, easiness, and performance improvement. Extremely low values of Photomodeler are because participants could not figure out how to get the desired result in Photomodeler in the given time slot. So, in the end, participants performed the task only on MoReLab and 3-Sweep.

C. QUALITATIVE RESULTS

Users were asked to model some objects in 3-Sweep and MoReLab. Modeling tasks are illustrated in Figure 7. Task 1 requires modeling a cuboid in video 1 and task 2 requires modeling a cuboid and a curved pipe in video 2.

Figure 8 shows the results of modeling task 1 done by an experienced participant. The modeling process in 3-Sweep starts with a boundary detection stage. This boundary detection stage needs color contrasts between foreground and background for a smooth boundary. Despite changing thresholds, boundaries are not smooth and the shape of the extracted model is very irregular, as shown in the bottom left of Figure 8. Even by spending more time, a user struggles to obtain the perfect boundary to extract a 3D model. On the other hand, MoReLab achieves a relatively better cuboid shape using the quadrilateral tool (see details of the quadrilateral tool in Siddique et al. [5]), as compared to the 3-Sweep result.

Figure 9 shows the results of modeling task 2. We can observe that the pipe is broken into two pieces in the bottom left of Figure 9. 3-Sweep models are broken and surfaces are not smooth. The lack of color contrast around the pipe causes an irregular boundary, leading to a broken pipe with a variable radius. 3-Sweep struggles with curved cylinders and breaks curved cylinders into pieces. The curved cylinder tool of MoReLab (see details of curved cylinder tool in Siddique et al. [5]) enables users to model a curved cylinder.

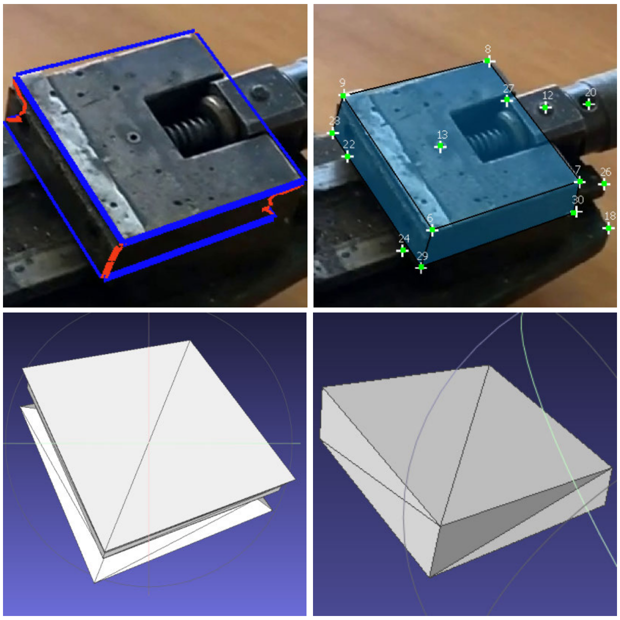


FIGURE 8. Figures on the left show 3-Sweep results, and figures on the right show MoReLab results for modeling task 1.

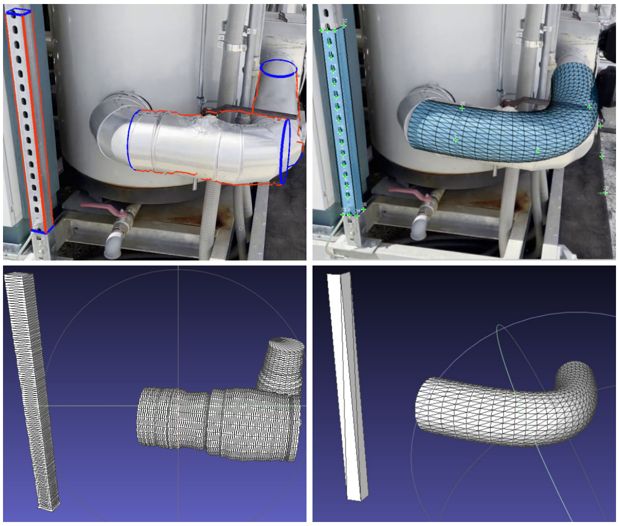


FIGURE 9. Figures on the left showing 3-Sweep results, and figures on the right show MoReLab results for modeling task 2.

The curved cylinder obtained from MoReLab is not broken, has a constant radius throughout the cylinder depth, and has a smooth surface. The surfaces of the sidebar are modeled with a quadrilateral tool. These surfaces are smoother as compared to 3-Sweep results.

Results of all participants for both modeling tasks have been provided in Appendix B.

D. MEASUREMENT RESULTS

Figure 4 and Figure 5 show the given and required measurement for video 1 and video 2, respectively. The data of measurements is only available for MoRelab and 3-Sweep. All the measurements are in centimeters. Photomodeler turned out to be extremely complicated software, and not even a single participant was able to understand how to perform calibration and measurement with Photomodeler, given uncalibrated image sequences in the time available. Hence, no measurement has been obtained from Photomodeler. In MoReLab, users add feature correspondences across multiple views, compute SfM, and then utilize the measurement tool for calibration and measurement. 3-Sweep does not provide calibration and measurement capabilities. Hence, the 3D model is exported from 3-Sweep as a mesh and loaded in MeshLab [29] for calibration and measurements.

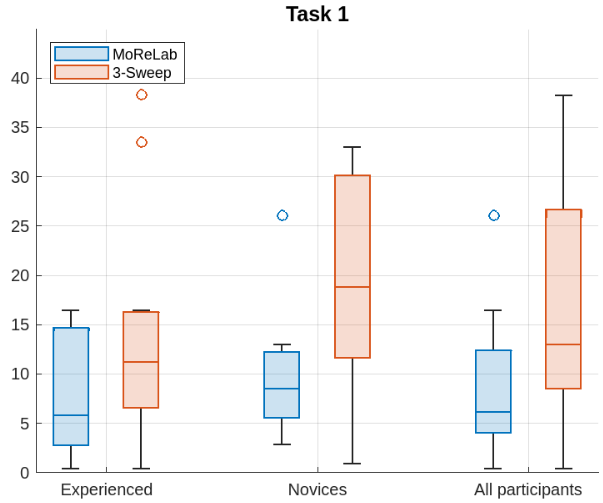


FIGURE 10. Box plots of relative errors for MoReLab and 3-Sweep for task 1 for different kinds of participants. Experienced participants obtain a median error of 5.841 with MoReLab, as compared to a median error of 11.28 with 3-Sweep. Novices obtain median errors 8.573 and 18.872 with MoReLab and 3-Sweep. All participants achieve a median error of 6.159 with MoReLab, whereas they record a median error of 13.044 with 3-Sweep.

Figure 10 shows a comparison of results for all kinds of participants between MoReLab and 3-Sweep for task 1. It is clear that MoReLab consistently shows a lower median error in comparison to 3-Sweep for experienced, novices, and all participants. It can also be observed that novices obtain significantly larger errors as compared to experienced participants with both MoReLab and 3-Sweep.

We can also observe similar results for task 2 (see Figure 11). Novices exhibit higher errors compared to experienced participants across both software programs. However, the gap in errors between MoReLab and 3-Sweep is relatively smaller in task 2 as compared to task 1. This can be attributed to the relatively complex scenario of video 2 in which we are capturing an entire scene as opposed to capturing a single object in video 1. So, feature marking becomes relatively more difficult, especially for novices. However, MoReLab still demonstrates lower median errors as compared to 3-Sweep across all participant categories.

Table 1 presents relative errors in the form of mean ± standard deviation (SD) for all categories of participants for the task 1. MoReLab achieves a lesser mean value of relative errors as compared to 3-Sweep for all categories of participants, for both tasks. Higher standard deviation

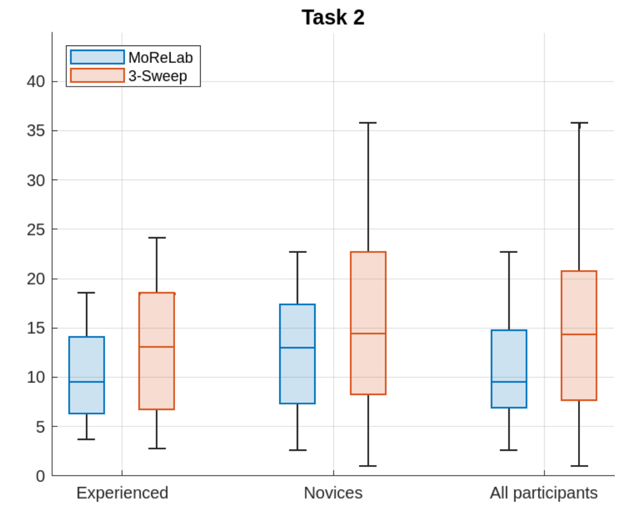


FIGURE 11. Box plots of relative errors for MoReLab and 3-Sweep for task 2 for different participant categories. Experienced participants obtain a median error of 9.542 with MoReLab, as compared to a median error of 13.120 with 3-Sweep. Novices obtain median errors of 13.035 and 14.490 with MoReLab and 3-Sweep. MoReLab yields a median error of 9.608 for all participants, whereas 3-Sweep results in a median error of 14.401.

TABLE 1. Mean ± Standard Deviation of the relative errors for all categories of participants for task 1 and task 2.

	Task 1		Task 2	
	MoReLab	3-Sweep	MoReLab	3-Sweep
Experienced	7.864 ± 6.035	14.131 ± 11.905	9.999 ± 4.672	13.314 ± 7.047
Novices	9.775 ± 6.677	19.387 ± 11.288	12.234 ± 6.488	15.713 ± 11.178
All	8.774 ± 5.861	16.633 ± 10.934	11.064 ± 5.313	14.456 ± 8.504

values for 3-Sweep as compared to MoReLab, can be attributed to large changes in results due to the different boundary detection thresholds for different participants. Lower standard deviation values for MoReLab as compared to 3-Sweep, indicate relatively more stable results for the same task done by different participants. From Table 1, we can see that mean values are always greater than standard deviation values. This indicates a positive skew in the data distribution, and the data points are clustered more towards larger positive values than smaller negative or positive values.

E. TWO SAMPLE T-TEST

The two-sample t-test is a statistical test used to determine if the means of two independent samples are significantly different from each other. This is commonly used when you have two groups, and you want to compare their means to see if they are truly different or if any observed difference could have occurred by chance. Most t-test calculating functions take the two samples as inputs and calculate p-value as an output. The computed p-value is compared with the significance level.

We conducted a two-sample t-test on measurement errors between MoReLab and 3-Sweep for different participant categories. We utilized MATLAB's t-test2[5] function to compute the hypothesis result and p-value. At the significance level 0.05, Table 2 reports p-values for all categories of participants for both tasks. A very high p-values, $\geq$ 0.9, suggests that we have achieved statistically significant results with current participants, and there is a high probability of observing the observed difference between the groups. The test results do not support the idea that the mean of MoReLab samples is statistically higher than the mean of 3-Sweep samples. This is consistent with Table 1 where the error obtained by using MoReLab is typically lower than the error obtained by using 3-Sweep.

TABLE 2. P-values of right-tailed t-tests performed between MoReLab and 3-Sweep samples under the assumption of unequal variances.

	Experienced	Novices	All
Task 1	0.929	0.982	0.995
Task 2	0.895	0.796	0.923

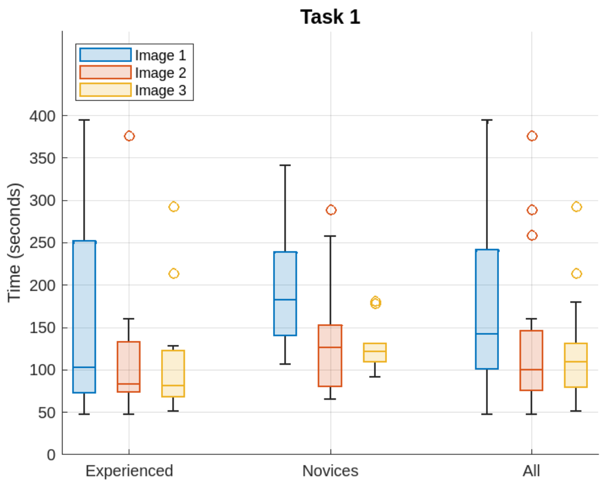


FIGURE 12. Box plot of time consumed for task 1. Time consumed on the first image is mostly higher as compared to other images for a participant for task 1.

F. LOG ANALYSIS

Session logs are records of interactions that occur during a session, often in the context of computing or online interactions. They can provide a detailed record of events leading up to an issue, aiding in troubleshooting and debugging processes. By analyzing session logs, we can get insights about user behavior with the software. We can collect session logs only for MoReLab because the source code for MoReLab is open-source, and it can be modified to collect event logs during the sessions. Figure 12 presents a box plot of time consumed in seconds, by experienced, novices, and all participants on each image for task 1, and Table 3 lists the corresponding median values. These three images are the selected ones to obtain the SfM reconstruction (see also the Discussion Section for more details about this aspect),

[5] https://www.mathworks.com/help/stats/ttest2.html

TABLE 3. The median time consumed by different categories of participants to add features on different frames in Task 1.

	Image 1	Image 2	Image 3
Experienced	104	84	82
Novices	183	127	122
All	104	101	110

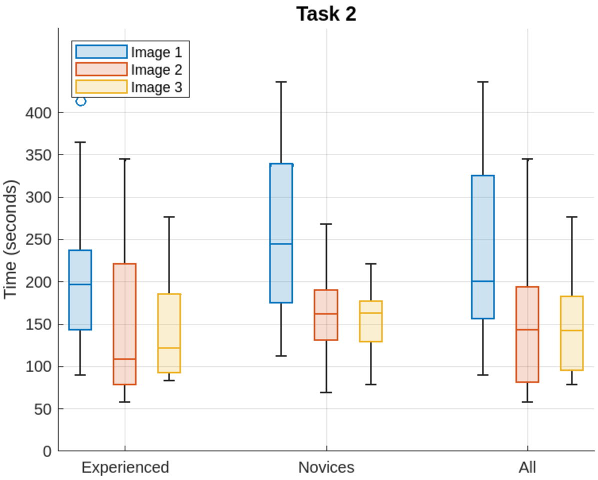


FIGURE 13. Box plot of time consumed for task 2. Time consumed on task 2 is relatively higher than the time consumed on task 1 because of the more complex scenario.

TABLE 4. Median time consumption for Task 2.

	Image 1	Image 2	Image 3
Experienced	197	109	122
Novices	246	163	164
All	143	101	101

Time consumed on the first image is higher as compared to other images for most participants. This is because a participant mostly decides and thinks about the location of a feature on the first image. Subsequently, for the remaining images, they simply replicate the process by adding the feature at the corresponding location. Another noticeable trend is that novices tend to spend more time adding features as compared to experienced participants. This trend is especially pronounced in the case of the first image because novices spend more time deciding the location of the feature as compared to experienced participants.

Figure 13 displays a box plot illustrating the time taken in seconds by all categories of participants for each image in task 2 and Table 4 presents corresponding median values. Similar to task 1, participants tend to spend more time on the first image and novices spend more time than experienced participants for the same task. Time consumption is higher for this task because video 2 presents a more complex scenario with featureless surfaces of pipe and other equipment. Hence, it is more difficult for each participant to identify unique feature points.

MoReLab has been developed in a way that allows the user to improve results by spending more time. The measurement error can be reduced slightly in most cases by adding more features in more viewpoints. Hence, the error is inversely proportional to both the number of features on each frame and the addition of the same features in more viewpoints. Taking into account the time restriction of twenty minutes per task for MoReLab in the user study, we advised participants to add approximately 25 features. Figure 14 shows box plots of a number of features added by different categories of participants for both tasks. Through this user study, we observe that by just adding almost 25 features to each of the three viewpoints, participants obtained sufficiently accurate results, better than 3-Sweep.

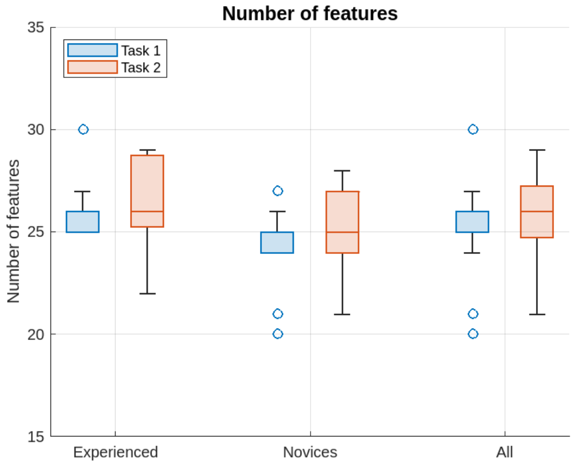


FIGURE 14. Box plots of number of features added by participants for both tasks. Looking at median values, it seems sufficient to add 25 features.

Looking at Figure 12 and Figure 13, we have median values of time consumption for three images. Combining the median values of each category of participant for each task, the average median values for image 1, image 2, and image 3 are 163, 114, and 117 seconds respectively. In total, a participant spent almost 7 minutes adding 25 feature correspondences for a task in the user study.

VI. DISCUSSION

The results obtained from this user study establish that MoReLab yields lesser relative error than 3-Sweep for experienced participants, as well as novices. The manual feature marking process allows the user to improve modeling results by spending more time on a particular task. Feature marking ability also improves, as the experience of using MoReLab for different videos, increases. However, it is very difficult to improve results by spending more time on a task or to improve results by increasing experience in 3-Sweep.

A. FRAMES SELECTION

MoReLab can load a set of images as well as a video. Hence, we conducted three experiments for MoReLab. In task 1 and task 2, we loaded a selected set of frames from video 1 and video 2, respectively. Frames were handpicked by us taking into account the suitability of viewpoint, motion blur, and camera motion. In the third experiment, video is loaded by the

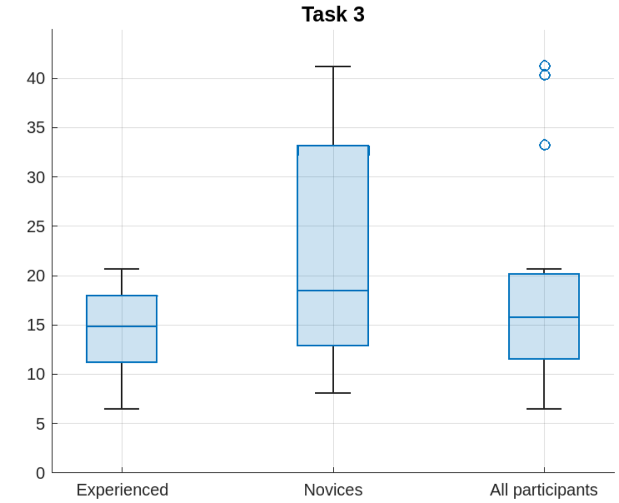


FIGURE 15. Box plots of relative errors for task 3 for all kinds of participants. Median error values are 14.91, 18.56, and 15.81 for experienced, novices, and all participants.

user, and frames are extracted by the user. Then, the user has to choose the frames where to add the features. In other words, in the first two experiments, selected frames were provided to the users, while in the third experiment, frames were extracted and selected by users. In this experiment, users worked on video 2. Similarly, the given and required measurements are the same as task 2 shown in Figure 5. Hence, task 3 is the same as task 2, except that users loaded a video instead of a set of frames.

Figure 15 shows results for task 3 for all kinds of participants. Comparing these results with Figure 11, participants get a higher error when working on a video instead of a set of selected frames. This is because participants can choose frames in which camera motion may be higher or lower than a suitable amount. Hence, the choice of frames or viewpoints is important to obtain good results. This suggest that a frame selection algorithm [30] can help users obtain better results in MoReLab, but also in other image-based modeling tool, by removing unnecessary and low-quality images from a video.

B. COPY AND PASTE FEATURES

To speed up this process of adding features, we added a functionality of copy and paste in MoReLab. By pressing *ctrl* + *c*, MoReLab makes a copy of the location of features and SIFT [31] feature descriptors of an image patch around the location on the frame. Then, the user can click on some other frame and press *ctrl* + *v*. A localized sliding window search is used to look for the corresponding location of each feature. Once features have been added to the new frame, the user can manually adjust the locations of features on the new frame using a mouse and keyboard. This workflow can speed up the process of adding features in some cases because the user can add features on one frame, paste those features on another frame and do minor adjustments manually.

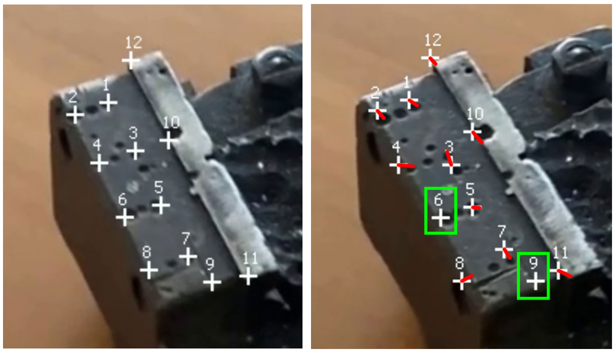


FIGURE 16. The left figure shows a zoom-in view of a small part of the frame on which feature locations are copied. The right figure shows a zoom-in view of a small part of another frame, in which features 6 and 9 have been computed to be placed at the correct locations. Each red line on the right figure illustrates the distance between the desired position and computed position for other features.

TABLE 5. Time consumed by participants in the pilot study to evaluate copy and paste functionality. The time saved by the user on average is about 34.53 % per image.

Participant	Image	Time (s)		Time saved (%)
		Experiment 1	Experiment 2	
#1	#2	170	112	34.12
	#3	277	71	74.37
#2	#2	239	114	52.30
	3	201	200	0.50
#3	#2	166	115	30.72
	#3	222	127	42.72
#4	#2	205	150	26.83
	#3	178	152	14.61

Figure 16 shows the result of copying feature locations from a frame and pasting feature locations on another frame. Features 6 and 9 are at the correct locations. However, other features need to be slightly adjusted by the amount shown by red lines, to bring them to correct locations. This functionality is useful in situations of slight change in viewpoint. The size of the window in which the image patch is searched is static, not dynamic. Hence, if there is a large change in viewpoint, the desired feature point may not even be present in the window to be searched. This operation of copy and paste also struggles with the situation of having two visually similar and close features. In this situation, a similar point is also present in the window to be searched, and the search operation may place features at a similar feature point instead of the original feature point. Despite these issues, this functionality is still very helpful to speed up feature marking in the modeling process of MoReLab.

A small pilot study has been conducted by 4 participants (2 experienced and 2 novices) who perform task 2 in two different experiments. In experiment 1, participants added features manually; but in experiment 2, participants used copy and paste functionality. In the first frame, each participant has to add features manually. Experiment 1 involves a participant adding features manually in image 2 and 3;

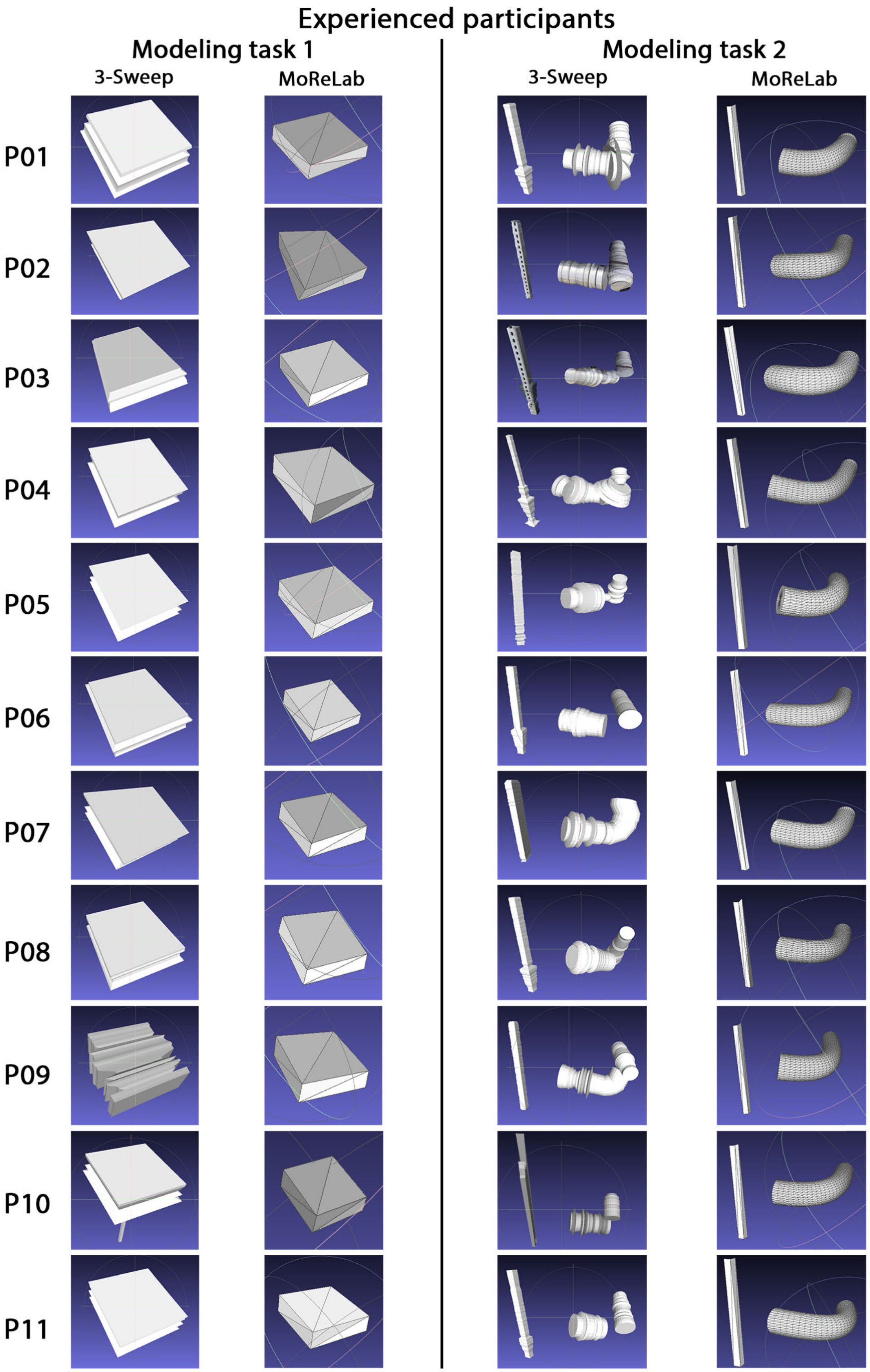


FIGURE 17. Modeling task 1 and modeling task 2 results by experienced participants. P01 denotes participant number 1 and so on.

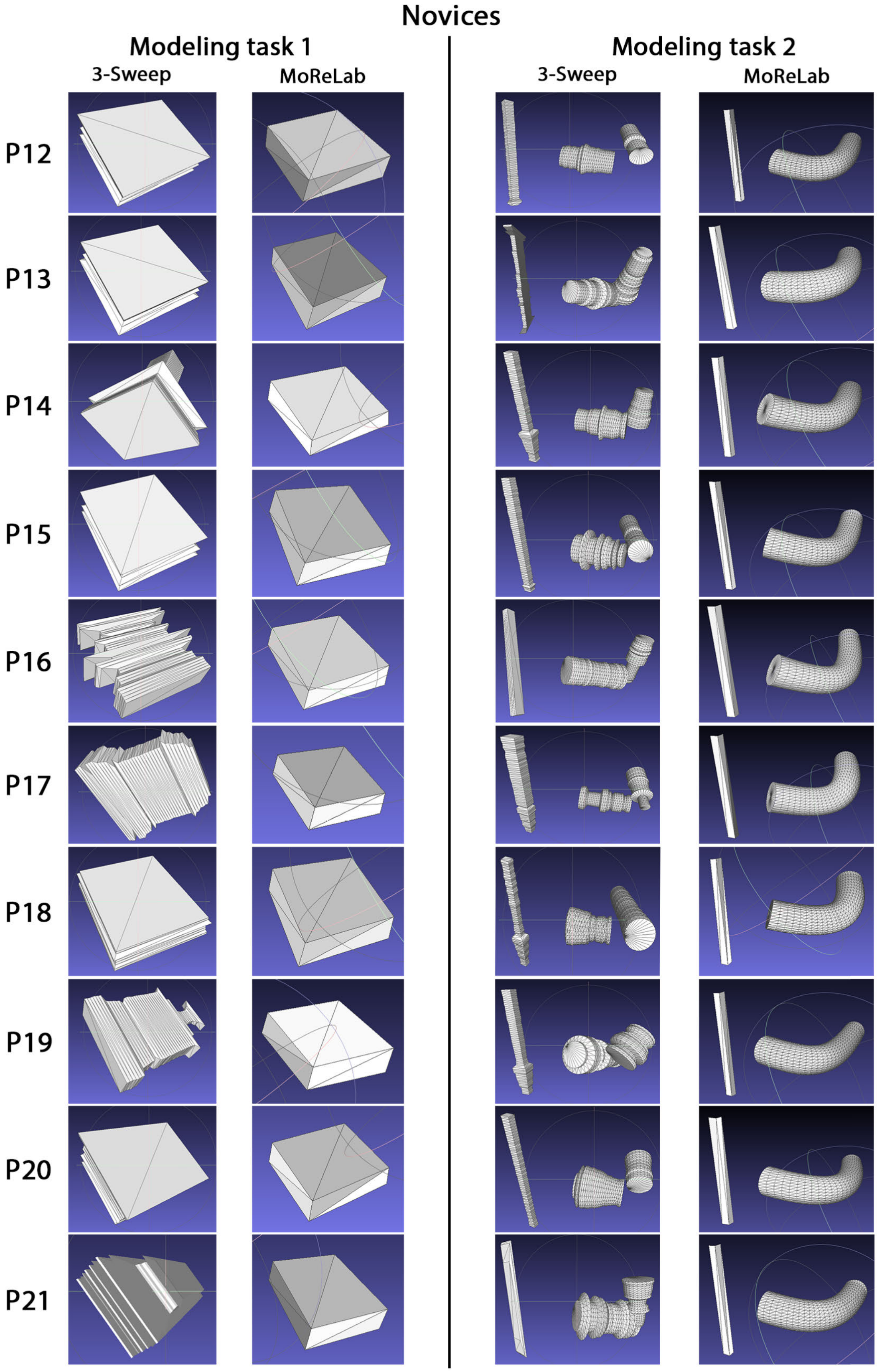


FIGURE 18. Modeling task 1 and modeling task 2 results by 10 novices, participant number 12 (P12) to participant number 21 (P21).

while experiment 2 involves a participant using the copy and paste feature. Table 5 reports the time spent by participants in both experiments. These results indicate an average of 34.529% of time saved per image in the feature marking process.

VII. CONCLUSION

We have proposed an improved version of our image-based modeling software MoReLab, which expands its original capabilities to make it an effective tool for the measurement inside industrial plants using low-quality and resolution videos captured by utility companies. We have shown that these new tools can save time for users when adding features to challenging videos.

More importantly, we have evaluated the performance of the software running a user study that showed three important facts: i) MoReLab generates 3D models with better reconstruction quality than other state-of-the-art tools; ii) MoReLab can create more accurate and precise measurements than state-of-the-art tools; iii) MoReLab is preferred by expert and novice users in terms of user-friendliness and easiness. This shows that MoReLab is a valid and efficient software for challenging situations (i.e., low-quality and low-resolution videos) to reach its design goals.

Based on the user feedback, potential future enhancements as discussed in Section VI are in the direction of an improved key-frame selection and a more robust copy and paste functionality. Other technological advancements that can be integrated into MoReLab include research works on automatic feature matching and camera parameters recovery from 2D points. Applications of MoReLab can be in the Cultural Heritage (CH) field, where precise and accurate measurements are crucial for documenting the current state of a site. In future works, we would like to explore the possibility of adapting MoReLab to accommodate the needs of the CH field.

APPENDIX A
QUESTIONNAIRES

In this appendix, we describe the questions that were asked in the questionnaires.

A. QUESTIONNAIRE Q1

The following information was collected in Questionnaire Q1:

1) Gender
2) Age range
3) Highest degree qualification
4) Are you familiar with the topic of Structure from Motion/image-based 3D reconstruction concepts?

B. QUESTIONNAIRE Q2

Questionnaires Q2, Q3, and Q4 were designed to evaluate MoReLab, 3-Sweep, and Photomodeler. Hence, the questions were the same in all these questionnaires. The following questions were asked in Questionnaire Q2:

1) Give a score for user-friendliness (1=very unfriendly, 5=very friendly) of the Graphical User Interface (GUI) of MoReLab.
2) Give a score for easiness (1=extremely difficult, 5=very easy) to learn MoReLab for a new user.
3) Give a score for improvement (1=none, 5=a lot) of your performance on the second video with respect to the first video.

By replacing MoReLab with 3-Sweep or Photomodeler in the above questions, we obtain Questionnaire Q3 or Q4, respectively.

APPENDIX B
MODELING RESULTS

In this appendix, we will put the results of all participants for both modeling tasks. Results of modeling task 1 and modeling task 2 are shown in Figure 17 for experienced participants and Figure 18 for novices. In modeling task 1, participants obtain strange shapes for the cuboid using 3-Sweep. In particular, the results of P09, P14, P16, P17, P19, and P21 are very different from the original shape. These strange results are due to a lack of color contrast around the cuboid. On the other hand, MoReLab shows more consistent results for different participants and obtains relatively smoother surfaces of the cuboid.

In modeling task 2, the pipe is broken for all participants for 3-Sweep. 3-Sweep can only handle straight cuboids or cylinders efficiently. Curved pipes represent a very challenging scenario for 3-Sweep. Even for two parts, the surface is not smooth and hence, the radius of the cylinder does not remain constant through the depth of the cylinder. On the other hand, MoReLab achieves a proper and continuous cylinder with a constant radius throughout the depth and with a smooth surface. The surfaces of the sidebar modeled by MoReLab are also smoother as compared to sidebars modeled with 3-Sweep. These results confirm the effectiveness of MoReLab for image-based interactive 3D modeling tasks.

REFERENCES

[1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[2] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.
[3] E. Rupnik, M. Daakir, and M. Pierrot Deseilligny, "MicMac—A free, open-source solution for photogrammetry," *Open Geospatial Data, Softw. Standards*, vol. 2, no. 1, pp. 1–9, Dec. 2017.
[4] D. Cernea, "OpenMVS: Multi-view stereo reconstruction library," CDC SeaCave SRL, Github.com, 2020.
[5] A. Siddique, F. Banterle, M. Corsini, P. Cignoni, D. Sommerville, and C. Joffe, "MoReLab: A software for user-assisted 3D reconstruction," *Sensors*, vol. 23, no. 14, p. 6456, Jul. 2023.
[6] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-Sweep: Extracting editable objects from a single photo," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–10, Nov. 2013.
[7] A. Van Den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. S. Torr, "Videotrace: Rapid interactive scene modelling from video," *ACM Trans. Graph. (ToG)*, vol. 26, no. 3, p. 86, 2007.
[8] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3D architectural modeling from unordered photo collections," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–10, Dec. 2008.

[9] M. Habbecke and L. Kobbelt, "An intuitive interface for interactive high quality image-based modeling," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1765–1772, Oct. 2009.

[10] Y. Doron, N. D. F. Campbell, J. Starck, and J. Kautz, "User directed multi-view-stereo," in *Proc. Comput. Vision—ACCV*. Cham, Switzerland: Springer, 2014, pp. 299–313.

[11] M. Xu, M. Li, W. Xu, Z. Deng, Y. Yang, and K. Zhou, "Interactive mechanism modeling from multi-view images," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–13, Nov. 2016.

[12] S. Rasmuson, E. Sintorn, and U. Assarsson, "User-guided 3D reconstruction using multi-view stereo," in *Proc. Symp. Interact. 3D Graph. Games*, May 2020, pp. 1–9.

[13] A. Baldacci, D. Bernabei, M. Corsini, F. Ganovelli, and R. Scopigno, "3D reconstruction for featureless scenes with curvature hints," *Vis. Comput.*, vol. 32, no. 12, pp. 1605–1620, Dec. 2016.

[14] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang, "Image-based plant modeling," in *Proc. ACM Siggraph*, vol. 25, Jul. 2006, pp. 599–604.

[15] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Seminal Graphics Papers: Pushing the Boundaries*. Cham, Switzerland: Springer, 2023, pp. 465–474.

[16] M. Wilczkowiak, P. Sturm, and E. Boyer, "Using geometric constraints through parallelepipeds for calibration and 3D modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 194–207, Feb. 2005.

[17] E. Töppe, M. R. Oswald, D. Cremers, and C. Rother, "Image-based 3D modeling via cheeger sets," in *Computer Vision—ACCV*. Cham, Switzerland: Springer, 2011, pp. 53–64.

[18] Y. Zhang, Z. Liu, T. Liu, B. Peng, and X. Li, "RealPoint3D: An efficient generation network for 3D object reconstruction from a single image," *IEEE Access*, vol. 7, pp. 57539–57549, 2019.

[19] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[20] B. Li, Y. Zhang, B. Zhao, and H. Shao, "3D-ReConstnet: A single-view 3D-object point cloud reconstruction network," *IEEE Access*, vol. 8, pp. 83782–83790, 2020.

[21] J. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, "Ners: Neural reflectance surfaces for sparse-view 3D reconstruction in the wild," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 29835–29847.

[22] J. Sun, X. Chen, Q. Wang, Z. Li, H. Averbuch-Elor, X. Zhou, and N. Snavely, "Neural 3D reconstruction in the wild," in *Special Interest Group Comput. Graph. Interact. Techn. Conf. Proc.*, Aug. 2022, pp. 1–23.

[23] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2021, pp. 27171–27183.

[24] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms*, 1999, pp. 298–372.

[25] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4937–4946.

[26] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math.*, vol. 21, no. 2, pp. 343–348, May 1967.

[27] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. Comput. Vis. Pattern Recognit. (CVPR), IEEE*, 2017, pp. 1–13.

[28] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, Sep. 1981.

[29] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: An open-source mesh processing tool," in *Proc. Eurograph. Italian Chapter Conf.*, Jul. 2008, pp. 129–136.

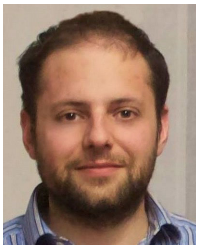
[30] F. Banterle, R. Gong, M. Corsini, F. Ganovelli, L. V. Gool, and P. Cignoni, "A deep learning method for frame selection in videos for structure from motion pipelines," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3667–3671.

[31] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

ARSLAN SIDDIQUE received the M.S. degree in computer science from Innopolis University, Russia, in 2020. He is currently pursuing the Ph.D. degree in computer science with the University of Pisa, Italy. Currently, he is a Graduate Fellow with the Visual Computing Laboratory, ISTI-CNR, Italy. He has published six peer-reviewed research articles. His research interests include 3D reconstruction, computer vision, and computer graphics. He was a recipient of the prestigious Marie Curie Ph.D. Fellowship for his doctoral studies.

PAOLO CIGNONI is currently a Research Director with CNR-ISTI, where he leads the Visual Computing Laboratory. Since 2022, he has been a part of the ACM SIGGRAPH Academy. His laboratory has provided the community with many successful and widely distributed advanced software tools that have helped millions of people's research and professional activities worldwide. He has published more than 180 papers in international refereed journals/conferences and has served on the program committee for all the most important conferences on computer graphics. His research interests include computer graphics fields, including geometry processing and machine learning technologies for 3D, applied to 3D scanning data processing, digital fabrication, scientific visualization, and digital heritage. He was awarded by the Eurographics Association "Best Young Researcher," in 2004, and he got the "Outstanding Technical Contributions Award," in 2021.

MASSIMILIANO CORSINI received the Ph.D. degree in information and telecommunication engineering from the University of Florence. He is currently a Senior Researcher with the Visual Computing Laboratory, ISTI-CNR, Italy. He worked on the appearance and shape acquisition of real objects for computer graphics applications, visual media production, and advanced visualization for cultural heritage applications. His current research interests include the applications of machine/deep learning, computer graphics, computer vision to digital humanities, underwater monitoring and other applications, and visual analytics. In these fields, he developed new algorithms and software tools documented in more than 70 publications in peer-reviewed international conferences and journals.



FRANCESCO BANTERLE received the Ph.D. degree in engineering from Warwick University, in 2009, where he developed inverse tone mapping that bridges the gap between SDR and HDR imaging. He is currently a full-time Researcher with the Visual Computing Laboratory, ISTI-CNR, Italy. He is the first author of the book *Advanced High Dynamic Range Imaging*, a reference book for HDR imaging research, and the co-author of the book *Image Content Retargeting*. His main research interests include HDR imaging, computer graphics (image-based lighting), computer vision, and deep learning.

Open Access funding provided by 'Consiglio Nazionale delle Ricerche-CARI-CARE-ITALY' within the CRUI CARE Agreement