





Article

# MoReLab: A Software for User-Assisted 3D Reconstruction

Arslan Siddique <sup>1,2</sup> , Francesco Banterle <sup>2,\*</sup> , Massimiliano Corsini <sup>2</sup> , Paolo Cignoni <sup>2</sup> ,  
Daniel Sommerville <sup>3</sup> and Chris Joffe <sup>3</sup>

<sup>1</sup> Department of Computer Science, Pisa University, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy; arslan.siddique@isti.cnr.it

<sup>2</sup> Visual Computing Laboratory, ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy; massimiliano.corsini@isti.cnr.it (M.C.); paolo.cignoni@isti.cnr.it (P.C.)

<sup>3</sup> EPRI, 3420 Hillview Avenue, Palo Alto, CA 94304, USA; dsommerville@epri.com (D.S.); cjoffe@epri.com (C.J.)

\* Correspondence: francesco.banterle@isti.cnr.it

**Abstract:** We present MoReLab, a tool for user-assisted 3D reconstruction. This reconstruction requires an understanding of the shapes of the desired objects. Our experiments demonstrate that existing Structure from Motion (SfM) software packages fail to estimate accurate 3D models in low-quality videos due to several issues such as low resolution, featureless surfaces, low lighting, etc. In such scenarios, which are common for industrial utility companies, user assistance becomes necessary to create reliable 3D models. In our system, the user first needs to add features and correspondences manually on multiple video frames. Then, classic camera calibration and bundle adjustment are applied. At this point, MoReLab provides several primitive shape tools such as rectangles, cylinders, curved cylinders, etc., to model different parts of the scene and export 3D meshes. These shapes are essential for modeling industrial equipment whose videos are typically captured by utility companies with old video cameras (low resolution, compression artifacts, etc.) and in disadvantageous lighting conditions (low lighting, torchlight attached to the video camera, etc.). We evaluate our tool on real industrial case scenarios and compare it against existing approaches. Visual comparisons and quantitative results show that MoReLab achieves superior results with regard to other user-interactive 3D modeling tools.

**Keywords:** image-based 3D reconstruction; 3D modeling; user-assisted 3D reconstruction; video-based 3D reconstruction; Structure from Motion; HCI



**Citation:** Siddique, A.; Banterle, F.; Corsini, M.; Cignoni, P.; Sommerville, D.; Joffe, C. MoReLab: A Software for User-Assisted 3D Reconstruction.

*Sensors* **2023**, *23*, 6456. <https://doi.org/10.3390/s23146456>

Academic Editor: Robert Sitnik

Received: 22 May 2023

Revised: 3 July 2023

Accepted: 10 July 2023

Published: 17 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

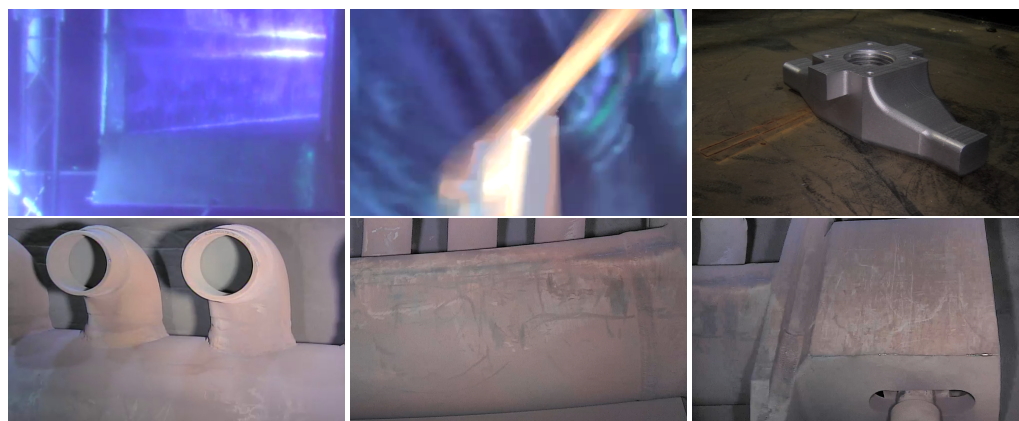
## 1. Introduction

Three-dimensional (3D) reconstruction is the process of creating a three-dimensional representation of a physical object or environment from two-dimensional images or other sources of data. The goal of 3D reconstruction is to create a digital model that accurately represents the shape, size, and texture of the object or environment. It can create accurate models of buildings, terrain, and archaeological sites, as well as virtual environments for video games and other applications. These 3D models can be created by automatic scanning of static objects using LiDAR scanners [1] or structured light scanners [2]. However, structured light scanning is sometimes expensive and is viable under certain conditions. Another solution is to create 3D models directly from high-resolution camera images captured under favorable lighting conditions. One such solution is a multi-camera-based photogrammetric setup capturing a fixed-size volume. Such camera setups are typically calibrated and capture high-resolution static photos simultaneously. These camera setups produce high-quality 3D models and precise measurements. However, such a setup is also very expensive due to the requirement of special equipment such as multiple cameras, special light sources, and studio setups. A low-cost solution to this problem is Structure from Motion (SfM), which aims to create sparse 3D models using multiple images of the same object, captured from different viewpoints using a single camera, and without requiring camera locations and orientations.

SfM has become a popular choice to create 3D models due to its low-cost nature and simplicity. Structure from Motion is a very well-studied research problem. In early research works, Pollefeys et al. [3] developed a complete system to build a sparse 3D model of the scene from uncalibrated image sequences captured using a hand-held camera. At the time of writing, there is a plethora of choices for SfM software packages, each with its unique features and capabilities. Some are open-source software, such as COLMAP [4], MicMac [5], OpenMVS [6], and so on, while some others are commercial software packages, such as Metashape (<https://www.agisoft.com> (accessed on 23 May 2023)), RealityCapture (<https://www.capturingreality.com> (accessed on 23 May 2023)), etc. They rely on automatic keypoint detection and matching algorithms to estimate 3D structures. The input to such an SfM software is only a collection of digital photographs, generally captured by the same camera. However, these fully automatic tools usually require suitable lighting conditions and high-quality photographs, to generate high-quality 3D models. These conditions are very difficult to be fulfilled in industrial environments because there may be low lighting (which exacerbates blurring) and utility companies may have legacy video cameras capturing videos at low resolution. These legacy cameras are meant for plants' visual inspection and enduring chemical, temperature, and radiation stresses.

The mentioned issues may become more severe in video-based SfM because video frames have motion blur and are aggressively compressed, leading to strong compression artifacts (e.g., ringing, blocking, etc.). Most modern cameras capture videos at 30 fps, so a few minutes of video produces a high number of frames, e.g., 10 min of footage is already 18,000 frames. Such a high number of frames not only increase computational time significantly but also give low-quality 3D output due to insufficient camera motion in consecutive frames. If we pass such featureless images (e.g., see Figure 1) as inputs to an SfM software, the number of accurately detected features and correspondences will be very low, leading to a low-quality 3D output. In this context, we have developed Movie Reconstruction Laboratory (MoReLab) (<https://github.com/cnr-isti-vclab/MoReLab> (accessed on 23 May 2023)), which is a software tool to perform user-assisted reconstruction on uncalibrated camera videos. MoReLab will address the problem of SfM in the case of featureless and poor-quality videos by exploiting the user indications about the structure to be reconstructed. A small amount of manual assistance can produce accurate models also in these difficult settings. User-assisted 3D reconstruction can significantly decrease the computational burden and also reduce the number of input images required for 3D reconstruction.

In contrast to automatic feature detection and matching-based SfM systems, the main contribution of MoReLab is a user-friendly interactive way that allows the user to provide topology prior to reconstruction. This modification allows MoReLab to achieve better results in featureless videos by leveraging the user's knowledge of visibility and understanding of the video across frames. Once the user has added features and correspondences manually on 2D images, a bundle adjustment algorithm [7] is utilized to estimate camera poses and a sparse 3D point cloud corresponding to these features. MoReLab achieves accurate sparse 3D points estimation by adding features on as few as two or three images. The estimated 3D point cloud is overlaid on manually added 2D feature points to give a visual indication of the accuracy of estimated 3D points. Then, MoReLab provides several primitives such as rectangles, cylinders, curved cylinders, etc., to model parts of the scene. Based on a visual understanding of the shape of the desired object, the user selects the appropriate primitive and marks vertices or feature points to define it in a specific location. This approach gives control to the user to extract specific shapes and objects in the scene. By exploiting inputs from the user at several stages, it is possible to obtain 3D reconstruction even from poor-quality videos. Additionally, the overall computational burden with regard to a fully automatic pipeline is significantly reduced.



**Figure 1.** Examples of frames from videos captured in industrial environments. These videos are not suitable for automatic SfM tools due to issues such as low resolution, aggressive compression, strong and moving directional lighting (e.g., a torchlight mounted on the camera), motion blur, featureless surfaces, liquid turbulence, low lighting, etc.

## 2. Related Work

There have been several research works in the field of user-assisted reconstruction from unordered and multi-view photographs. Early research works include VideoTrace [8], which is an interface to generate realistic 3D models from video. Initially, automatic feature detection-based SfM is applied to video frames, and a sparse 3D point cloud is overlaid on the video frame. Then, the user traces out the desired boundary lines, and a closed set of line segments generates an object face. Sinha et al. [9] modeled architectures using a combination of piecewise planar 3D models. Their system also computes sparse 3D data in such a way that lines are extracted, and vanishing points are estimated in the scene as well. After this automatic preprocessing, the user draws outlines on 2D photographs. Piecewise planar 3D models are estimated by combining user-provided 2D outlines and automatically computed sparse 3D points. A few such user interactions can create a realistic 3D model of the scene quickly. Hu et al. [10] developed an interface for creating accurate 3D models of complex mechanical objects and equipment. First, sparse 3D points are estimated from multi-view images and are overlaid on 2D images. Second, stroke-based sweep modeling creates 3D parts, which are also overlaid on the image. Third, the motion structure of the equipment is recovered. For this purpose, a video clip recording of the working mechanism of the equipment is provided, and a stochastic optimization algorithm recovers motion parameters. Rasmuson et al. [11] employ COLMAP [4] as a preprocessing stage to calibrate images. Their interface allows users to mark image points and place quads on top of images. The complete 3D model is obtained by applying global optimization on all quad patches. By exploiting user-provided information about topology and visibility, they are able to model complex objects as a combination of a large number of quads.

Some researchers developed interfaces where users can paint desired foreground regions using brush strokes. Such an interface was developed by Habbecke and Kobbelt [12]. Their interface consists of a 2D image viewer and a 3D object viewer. The user paints the 2D image in a 2D image viewer with the help of a stroke. The system computes an optimal mesh corresponding to the user-painted region of input images. During the modeling session, the system incrementally continues to build 3D surface patches and guide the surface reconstruction algorithm. Similarly, in the interface developed by Baldacci et al. [13], the user indicates foreground and background regions with different brush strokes. Their interface allows the user to provide localized hints about the curvature of a surface. These hints are utilized as constraints for the reconstruction of smooth surfaces from multiple views. Doron et al. [14] require stroke-based user annotations on calibrated images, to guide multi-view stereo algorithms. These annotations are added into a variational optimization framework in the form of smoothness, discontinuity, and depth ordering constraints. They

show that their user-directed multi-view stereo algorithm improves the accuracy of the reconstructed depth map in challenging situations.

Another direction in which user interfaces need to be developed is single-view reconstruction. Single-view reconstruction is complicated without any prior knowledge or manual assistance because epipolar cannot be established. Töppe et al. [15] introduced convex shape optimization to minimize weighted surface area for a fixed user-specified volume in single-view 3D reconstruction. Their method relies on implicit surface representation to generate high-quality 3D models by utilizing a few user-provided strokes on the image. 3-Sweep [16] is an interactive and easy-to-use tool for extracting 3D models from a single photo. When a photo is loaded into the tool, it estimates the boundary contour. Once the boundary contour is defined, the user selects the model shape and creates an outline of the desired object using three painting brush strokes, one in each dimension of the image. By applying the foreground texture segmentation, the interface quickly creates an editable 3D mesh object which can be scaled, rotated, or translated.

Recently, researchers have made significant progress in the area of 3D reconstruction using deep learning approaches. The breakthrough work by Mildenhall et al. [17] introduced NeRF, which synthesizes novel views of a scene using a small set of input views. A NeRF is a fully connected deep neural network whose input is a single 5D coordinate (spatial location  $(x, y, z)$  and viewing direction  $(\theta, \phi)$ ), and output is emitted radiance and volume density. To the best of our knowledge, a NeRF-like method that tackles at the same time all conditions of low-quality videos (blurred frames, low resolution, turbulence caused by liquids, etc.) have not been presented yet [18]. A GAN-based work, Pi-GAN [19], is a promising generative model-based architecture for 3D-aware image synthesis. However, their method has the main focus on faces and cars, so to be applicable in our context, there is the need to build a specific dataset for re-training (e.g., a dataset of industrial equipment, 3D man-made objects, and so on). Tu et al. [20] presented a self-supervised reconstruction model to estimate texture, shape, pose, and camera viewpoint using a single RGB input and a trainable 2D keypoint estimator. Although this method may be seminal for more general 3D reconstructions, the current work is currently focused on human hands.

Existing research works pose several challenges for low-quality industrial videos, which are typically captured by industrial utility companies. First, most works [8–11,14] in user-assisted reconstruction, still require high-quality images because they are using automatic SfM pipelines as their initial step. Our focus is on low-quality videos in industrial scenarios, where SfM generates an extremely sparse point cloud, making subsequent 3D operations extremely difficult. Second, these research works lack sufficient functionalities to be able to model a variety of industrial equipment. Third, these research works are not available as open-source, limiting their usage for non-technical users. Hence, our research contributions are as follows:

- A graphical user interface for the user to add feature points and correspondences manually to model featureless videos;
- Several primitive shapes to model the most common industrial components.

In MoReLab, there is no feature detection and matching stage. Instead, the user needs to add features manually based on the visual understanding of the scene. We have implemented several user-friendly functionalities to speed up this tedious process for the user. MoReLab is open-source software targeted for modeling industry scenarios and available for non-commercial applications for everyone.

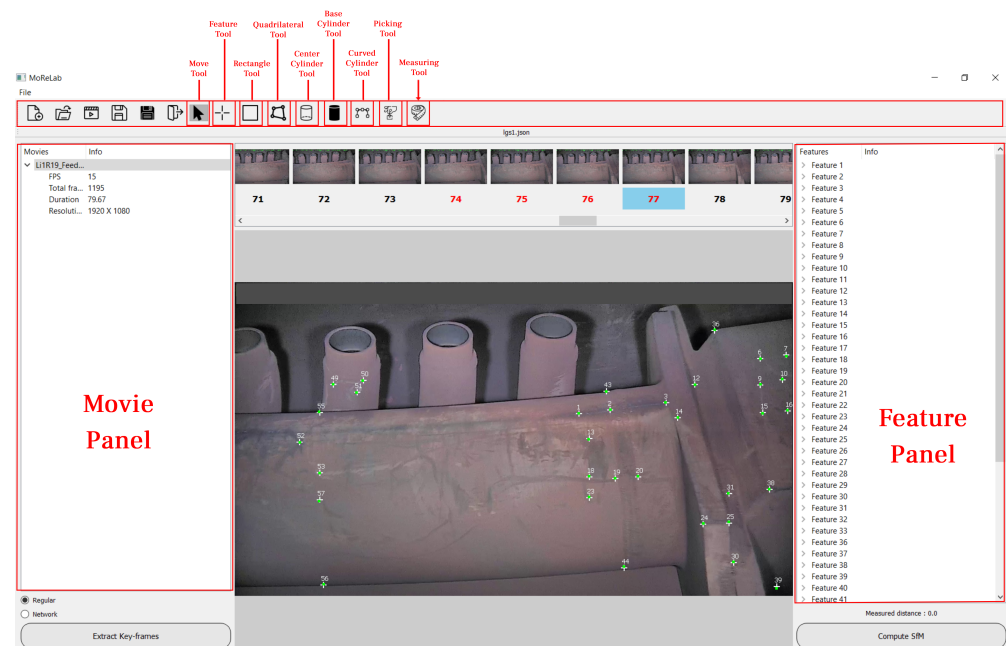
### 3. Method

In this section, we describe the pipeline, the graphical user interface, and the primitive tools of MoReLab. We designed the software to be user-friendly and easy to use for new users. However, understanding the tools and design of this software will enable the user to achieve optimal results with MoReLab.



### 3.1. Graphical User Interface

Figure 2 shows the graphical user interface (GUI) of MoReLab. The user starts the 3D modeling process by importing a video, which is loaded into the movie panel. Then, by clicking on the 'Extract Key-frames' button, the extracted keyframes would appear in the central top scroll bar area. The user can click on the thumbnail, and display the corresponding image in the central area. At this point, it is possible to use the 'Feature Tool' to add features to the image with a mouse double-click at the desired location. A white-colored plus-shaped feature appears on the image, and the information about the feature will appear in the right feature panel. Information includes the associated frame and the feature location. Once the user has marked features, the 'Compute SfM' can be launched. This option will perform bundle adjustment and calculate the 3D structure. 3D points are visualized on the image as green-colored points. Figure 2 shows estimated 3D points that are approximately at the same locations as marked 2D features. Once 3D points have been estimated, the user can make use of the shape tools, i.e., the rectangle tool, quadrilateral tool, center cylinder tool, base cylinder tool, and curved cylinder tool, to model different shapes. The picking tool allows the user to select and delete different primitives. Finally, the measuring tool allows the user to calibrate 3D data points and perform measurements.



**Figure 2.** The graphical user interface of MoReLab. The toolbar at the top allows the user to switch between different tools.

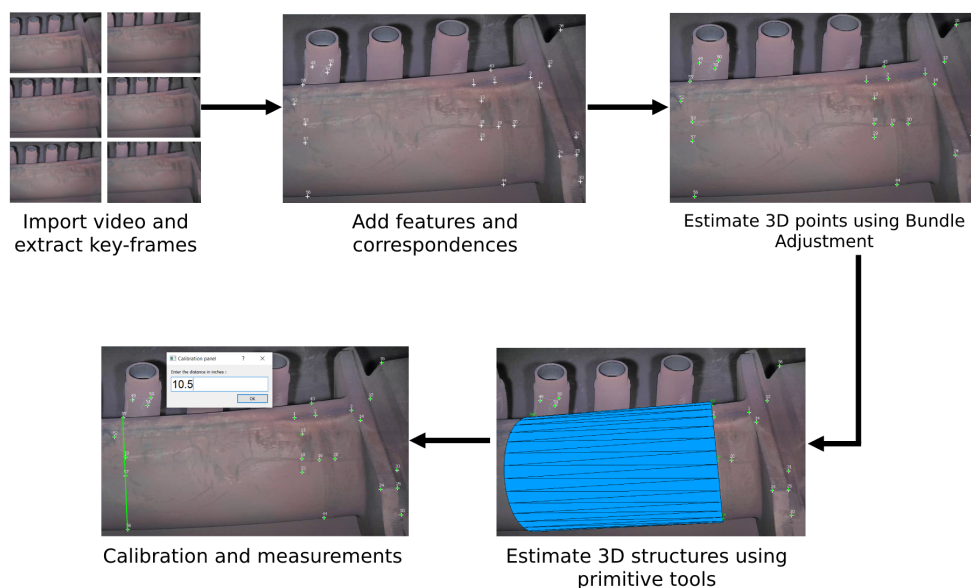
### 3.2. Pipeline

Figure 3 presents the pipeline of our software. This pipeline consists of the following steps:

#### 3.2.1. Manual Feature Extraction

In the second step, the user grabs the feature tool and starts to add features. A feature refers to an identifiable and distinctive pattern, shape, color, texture, or point of interest in an image. The user needs to choose only a few frames based on the recognizability of features. Since we are using the eight-point algorithm [21] to compute the fundamental matrix in the next step, the user needs to add a minimum of eight features in at least two frames. However, increasing the number of features and adding features on more views would increase computational accuracy. To speed up this tedious process, the user can copy the location of all features on an image with a simple keyboard press and paste features at

pixel coordinates on other keyframes. Each feature location can be adjusted by dragging it to the correct location.



**Figure 3.** MoReLab reconstruction pipeline.

### 3.2.2. Extract Keyframes

In the first step, a video is loaded into the software, and frames are extracted. However, all frames are not required because of several reasons. First, processing all frames is computationally very expensive. Second, some video frames have motion blur, making it difficult for the user to add features. Third, a very small baseline between consecutive frames causes inaccurate triangulation and reconstruction. We implemented two methods of keyframe extraction in MoReLab: The first approach is to regularly sample frames at a desired frequency, and the second approach is based on a network [22].

This latter method automatically removes out-of-focus frames, blurred frames, and redundant frames (i.e., due to a static scene). In addition, it selects frames that may lead to a high-quality reconstruction. Note that other frame selection methods can be employed such as Nocerino et al. [23].

We designed a simple calibration panel containing a combo box to switch easily between both approaches. The first approach is faster than the latter.

### 3.2.3. Bundle Adjustment

In the third step, feature locations provided by the user are utilized to compute a sparse 3D point cloud through bundle adjustment. bundle adjustment is the process of refining camera parameters and 3D point locations simultaneously, by minimizing the re-projection error between input 2D locations and projected 2D locations of 3D points on the image. The minimization algorithm being used is the Trust Region Reflective Algorithm [24]. Assume that  $n$  3D points can be observed in  $m$  views. Let  $x_{ij}$  denote the  $i$ -th feature location on the  $j$ -th image,  $X_i$  denote the corresponding  $i$ -th 3D point, and  $C_j$  denote the camera parameters corresponding to the  $j$ -th image, then the objective function for bundle adjustment can be defined as:

$$\arg \min_{X_i, C_j} \sum_{i=1}^n \sum_{j=1}^m b_{ij} d(f(X_i, C_j), x_{ij}), \quad (1)$$

where  $b_{ij}$  denotes a binary variable that equals 1 if the feature  $i$  is visible on the image  $j$  and 0 otherwise.  $f(X_i, C_j)$  is the projection of  $i$ -th 3D point on  $j$ -th image.  $d(f(X_i, C_j), x_{ij})$  indicates the Euclidean distance between the projection point and  $x_{ij}$ . After this mini-

mization, we obtain optimal camera parameters and locations of 3D points in the world coordinate frame.

### 3.2.4. Primitive Tools

We have implemented tools based on geometric primitive shapes, to be able to model a variety of industrial equipment. These tools are described as follows:

- **Rectangle Tool:** This tool allows the user to model planar surfaces. To estimate a rectangle, the user should click on four features in an anti-clockwise manner. The 3D sparse points, corresponding to four selected features, compute new vertices to form a rectangle in which all inner angles are constrained to be 90 degrees.
- **Quadrilateral Tool:** This tool allows creating a quadrilateral using four 2D features. This tool connects 3D sparse points corresponding to selected 2D features. Unlike the rectangle tool, there is no 90-degree angle constraint and opposite sides might not be parallel. Hence, inner angles might not be 90 degrees. If the selected four points are not in a single plane, a quadrilateral is also not planar.
- **Center Cylinder Tool:** This tool models a cylindrical object using a specific point. This is useful when the center of the base of cylindrical equipment is visible. The user needs to click on four points. The point can be either a 2D feature or an area containing a 3D primitive. For 2D features, we get the corresponding 3D sparse point computed from bundle adjustment. The initial three points form the base of the cylinder, and the fourth point determines the height of the cylinder. The first point corresponds to the center of the base, the second point forms an axis point, and the third point corresponds to the radius of the cylinder.

A cylinder is estimated by computing new axes. Let us denote input 3D point data points as  $\mathbf{P1}$ ,  $\mathbf{P2}$ ,  $\mathbf{P3}$ , and  $\mathbf{P4}$ . We define a reference system:

$$\mathbf{T} = \frac{\mathbf{P2} - \mathbf{P1}}{\|\mathbf{P2} - \mathbf{P1}\|} \quad \mathbf{b} = \mathbf{P3} - \mathbf{P1} \quad \mathbf{N} = \frac{\mathbf{T} \times \mathbf{b}}{\|\mathbf{T} \times \mathbf{b}\|} \quad \mathbf{B} = \mathbf{T} \times \mathbf{N}, \quad (2)$$

where  $\mathbf{N}$  is the normal,  $\mathbf{B}$  is the bi-normal, and  $\mathbf{T}$  is the tangent.  $\|\mathbf{b}\|$  is the radius of the cylinder, and the base of the cylinder lies in the plane formed by  $\mathbf{T}$  and  $\mathbf{B}$  axes. The height of the cylinder is calculated by projecting the vector  $\mathbf{P4} - \mathbf{P1}$  on  $\mathbf{N}$ .

- **Base Cylinder Tool:** This tool allows users to create a cylinder in which the initial three selected points lie on the base of the cylinder. The fourth point determines the height of the cylinder. This is useful for most industrial scenarios because, in most cases, we can only see the surface of the cylindrical equipment, and the base center is not visible. As in other tools, the user needs to select the points by clicking on them. The point can be either a 2D feature or an area containing a 3D primitive. For 2D features, we get the corresponding 3D sparse point computed from bundle adjustment. Similar to the center cylinder tool, first, we need to calculate a new local axes system, i.e.,  $\mathbf{T}$ ,  $\mathbf{B}$ , and  $\mathbf{N}$  similar to how these axes were calculated in the center cylinder tool. In the new local system, the first point is considered to be at the origin; while the second and third 3D points are projected on  $\mathbf{B}$  and  $\mathbf{T}$  to obtain their 2D locations in the plane formed by  $\mathbf{B}$  and  $\mathbf{T}$ . Given these three 2D points, we find the circle passing through these three points. If three points are in a straight line, the circle would not be estimated because it would have an infinite radius. Once we know the center and radius of this circle, we calculate the base and top points, similar to the center cylinder tool.
- **Curved Cylinder Tool:** This tool models curved pipes and curved cylindrical equipment. The user clicks on four points at any part of the image. Then, the user clicks on a sparse 3D point obtained from bundle adjustment, this last point assigns an approximate depth to the curve just defined. To do this, first, we estimate the plane containing this 3D point, denoted as  $\mathbf{P}$ . Typically, a plane is defined as:

$$ax + by + cz + d = 0, \quad (3)$$

where coefficients  $a$ ,  $b$ , and  $c$  can be obtained from the  $z$ -vector of a camera projection matrix,  $M$ .  $d$  is obtained by the dot product of the  $z$ -vector and  $\mathbf{P}$ . Assume that  $\mathbf{s}$  represents the 2D point clicked by the user at  $(x, y)$  coordinates on the image and  $\mathbf{X}$  represents the unknown 3D point corresponding to  $\mathbf{s}$ .

$$M = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} \quad x = \frac{M_1 \mathbf{X}}{M_3 \mathbf{X}} \quad y = \frac{M_2 \mathbf{X}}{M_3 \mathbf{X}} \quad [a \quad b \quad c] \cdot \mathbf{X} + d = 0. \quad (4)$$

Equation (4) can be re-arranged into the form of linear equation  $\mathbf{A}\mathbf{X} = \mathbf{b}$  and a linear solver finds  $\mathbf{X}$ . Through this procedure, four 3D points are obtained corresponding to the clicked points on the frame. These four 3D points act as control points to estimate a Bézier curve [25] on the frame. Similarly, the user can define the same curve from a different viewpoint. These curves defined at different viewpoints are optimized to obtain the final curve in 3D space. This optimization is about minimizing the sum of the Euclidean distance between control points across frames and the Euclidean distance between the location of the projected point and the location of the 2D feature in each frame containing the curve.

Assume that  $m$  frames contain curves. Let  $\mathbf{x}_{ij}$  denote the  $i$ -th feature location on the  $j$ -th image,  $\mathbf{CP}_{ij}$  denotes  $i$ -th control point on the  $j$ -th frame.  $\mathbf{X}_i$  denotes corresponding  $i$ -th 3D point, and  $\mathbf{C}_j$  denotes camera parameters corresponding to  $j$ -th image, then the objective function for optimization of curves is defined as:

$$\arg \min_{\mathbf{CP}_{ij}} \sum_{j=1}^{m-1} \mathbf{CP}_j - \mathbf{CP}_{j+1} + \sum_{j=1}^m \sum_{i=1}^4 d(f(\mathbf{CP}_{ij}, \mathbf{C}_j), \mathbf{x}_{ij}), \quad (5)$$

where  $f(\mathbf{CP}_{ij}, \mathbf{C}_j)$  is the projection of the  $i$ -th control point on the  $j$ -th image. The Euclidean distance between the projected point and  $\mathbf{x}_{ij}$ , is represented by  $d(f(\mathbf{CP}_{ij}, \mathbf{C}_j), \mathbf{x}_{ij})$ . The optimal control points, obtained from optimization, estimate the final Bézier curve and the cylinder needs to be built around this curve. In order to define the radius of this curved cylinder, the user clicks on a 3D point, and a series of cylinders are computed around the final curve.

### 3.2.5. Calibration and Measurements

Taking real-world measures on the reconstructed object is important in industrial scenarios. For example, the 3D reconstruction can be used to evaluate if a pipe or other objects have been deformed and then make the necessary maintenance/actions. The measurement tools allow the user to measure the distance between two 3D points. These points can be in any primitive, i.e., quad, cylinder, or simple 3D point.

The sparse point cloud obtained from bundle adjustment cannot be used directly to get real-world measurements because the camera is calibrated up to a scale factor. Hence, first, the user needs to assign the proper scale between two 3D points. In this step, the user draws a line between two 3D points, and a simple panel opens up and asks the user to input the corresponding known distance. This ground-truth distance is employed to calculate a distance scaling factor. The second step is the actual measurement, in which the user can draw a line between any 3D points, and MoReLab calculates the corresponding properly scaled distance using the scaling factor.

## 4. Experiments and Results

We analyzed the performance of MoReLab and other approaches on some videos for modeling different industrial equipment. We started our comparison using an image-based reconstruction software package, showing that the results are of poor quality in these cases. Then, we will show what we obtain with user-assisted tools for the same videos. We performed our experiments on two datasets. The first dataset consists of videos provided



by a utility company in the energy sector. Ground-truth measurements have also been provided for two videos of this dataset for quantitative testing purposes. The second dataset was captured in our research institute to provide some additional results.

Agisoft Metashape is a popular high-quality commercial SfM software, which we applied to our datasets. Such software extracts features automatically, matches them, calibrates cameras, densely reconstructs the final scene, and generates a final mesh. The output mesh can be visualized in a 3D mesh processing software such as MeshLab [26].

Results obtained with SfM software allow us to model these videos with user-assisted tools, e.g., see Figure 7b. 3-Sweep is an example of software for user-assisted 3D reconstruction from a single image. It requires the user to have an understanding of the shapes of the components. Initially, the border detection stage uses edge detectors to estimate the outline of different components. The user selects a particular primitive shape, and three strokes generate a 3D component that snaps to the object outline. Such a user-interactive interface combines the cognitive abilities of humans with fast image processing algorithms. We will perform a visual comparison of modeling different objects with an SfM software package, 3-Sweep, and our software. Table 1 presents a qualitative comparison of the functionalities of software packages being used in our experiments. The measuring tool in MeshLab performs measurements on models exported from Metashape and 3-Sweep.

**Table 1.** Qualitative comparison of the functionalities of different software packages.

	Automatic Feature Matching	Bundle Adjustment	Rectangle/Cylinder	Curved Cylinder	Measurements
Metashape	✓	✓	✗	✗	✓
3-Sweep	✗	✗	✓	✗	✗
MoReLab	✗	✓	✓	✓	✓

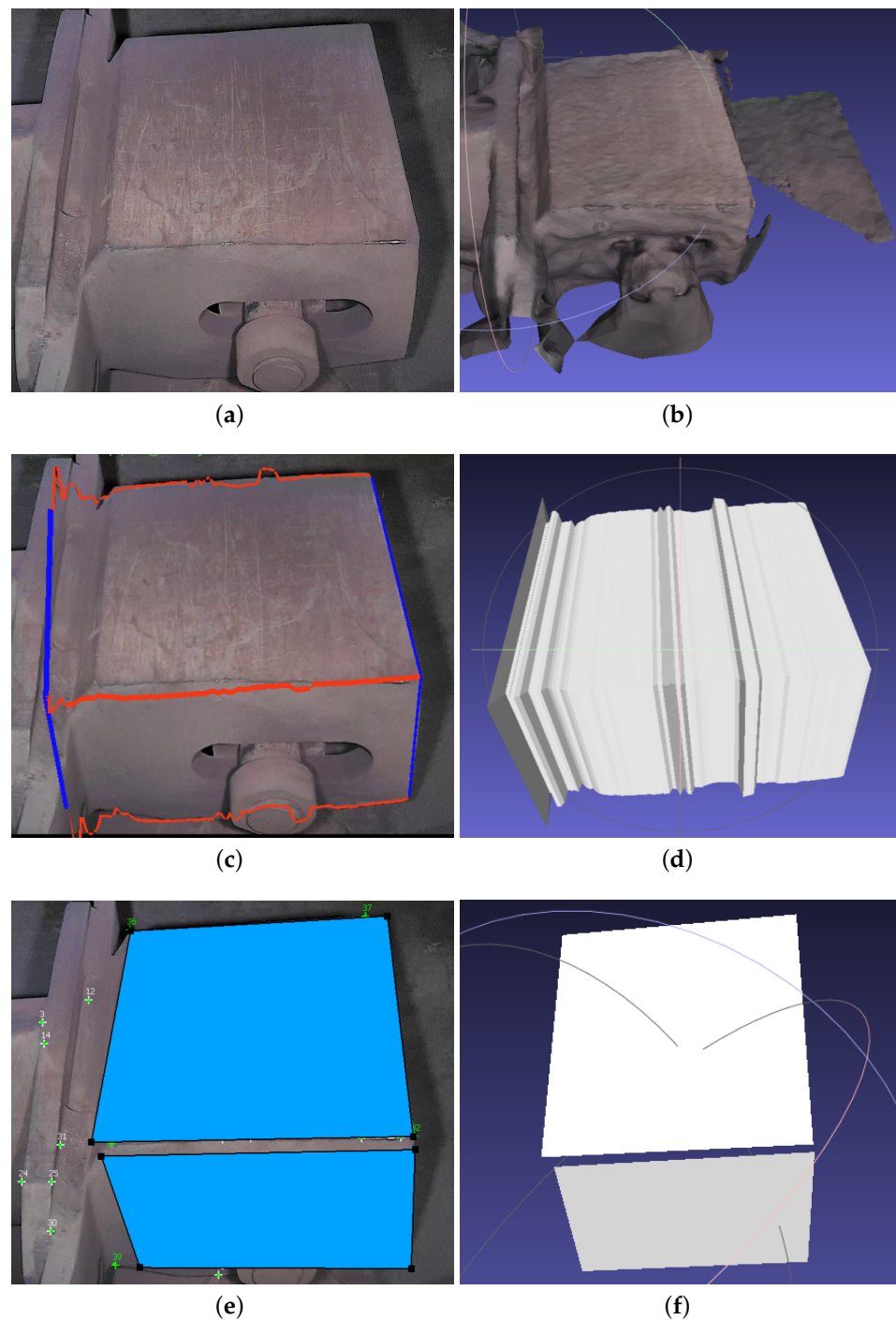
#### 4.1. Cuboid Modeling

3-Sweep allows us to model cuboids. In MoReLab, flat 2D surfaces can be modeled with the rectangle tool and quadrilateral tool. To estimate a cuboid, more rectangles and quadrilaterals need to be estimated in other views as well to form a cuboid. Figure 4 shows the results of modeling an image segment containing a cuboid with Metashape, 3-Sweep, and MoReLab. Figure 4b shows the result of the approximation of the cuboid with Metashape. There is a very high degree of approximation and the surface is not smooth.

Figure 4c,d show the result of extracting a cuboid using 3-Sweep. The modeling in 3-Sweep starts by detecting the boundaries of objects at the start. Despite changing thresholds, this detection stage is prone to errors and shows very little robustness. Hence, the boundary of the extracted model is not smooth, and the shape of the model is irregular. Figure 4e,f illustrate modeling in MoReLab using the rectangle tool and quadrilateral tool. Every rectangle in Figure 4e is planar with 90-degree inner angles. However, each quadrilateral in Figure 4f may or may not be planar, depending on the 3D locations of the points being connected. In other words, orthogonality is not enforced in Figure 4f.

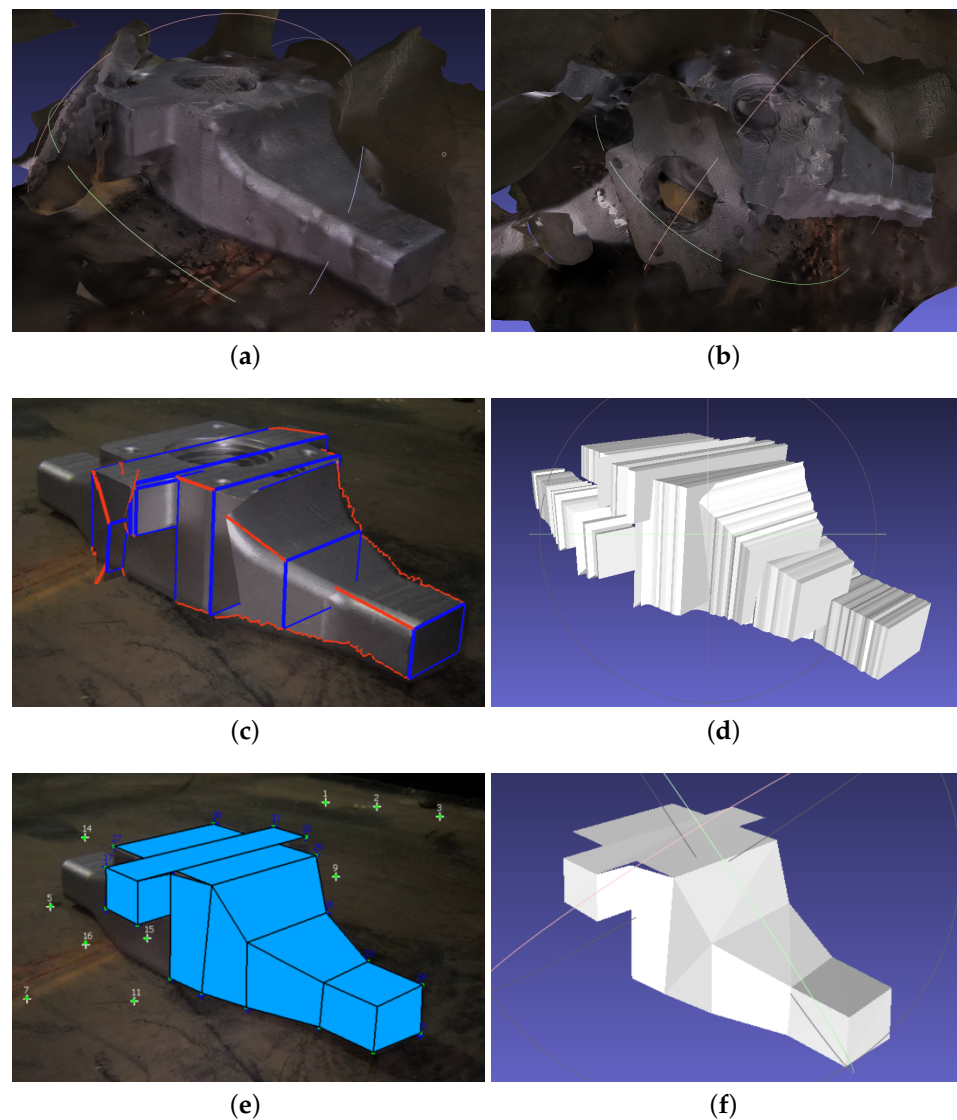
#### 4.2. Jet Pump Beam Modeling

The jet pump beam is monitored in underwater and industrial scenarios, to observe deformations or any other issues. The jet pump beam is also modeled with different software programs in Figure 5. Metashape reconstructs a low-quality 3D model of the jet pump beam. Another view of Figure 8a shows that Metashape has estimated two jet pump beams instead of a single jet pump beam. The beam model is passing through the floor in this reconstruction. The jet pump beam model is missing surfaces at different viewpoints, and the model is merged with the floor at different places. This low-quality result can be attributed to dark environments, the featureless surface of the pump, and the low distance of the object from the camera. The mesh, obtained by modeling the jet pump beam with 3-Sweep, has a low-quality boundary and does not represent the original shape of the jet pump beam (see Figure 5d).



**Figure 4.** Modeling a cuboid with Metashape, 3-Sweep, and MoReLab: (a) A frame of input video; (b) Cuboid modeling with Metashape; (c) Paint strokes snapped to cuboid outline; (d) Cuboid modeling with 3-Sweep; (e) Modeling with rectangle tool; (f) MeshLab visualization of estimated surfaces of the cuboid.

The jet pump beam has also been modeled with MoReLab in Figure 5e. The quadrilateral tool has been used to estimate the surface of the jet pump beam. The output mesh is formed by joining piecewise quadrilaterals on the surface of the jet pump beam. Quadrilaterals on the upper part of the jet pump are aligned very well together; but, some misalignment can be observed on surfaces at the side of the jet pump beam. The resulting mesh has a smooth surface and reflects the original shape of the jet pump beam. Hence, this result is better than the mesh in Figure 5b and mesh in Figure 5d.

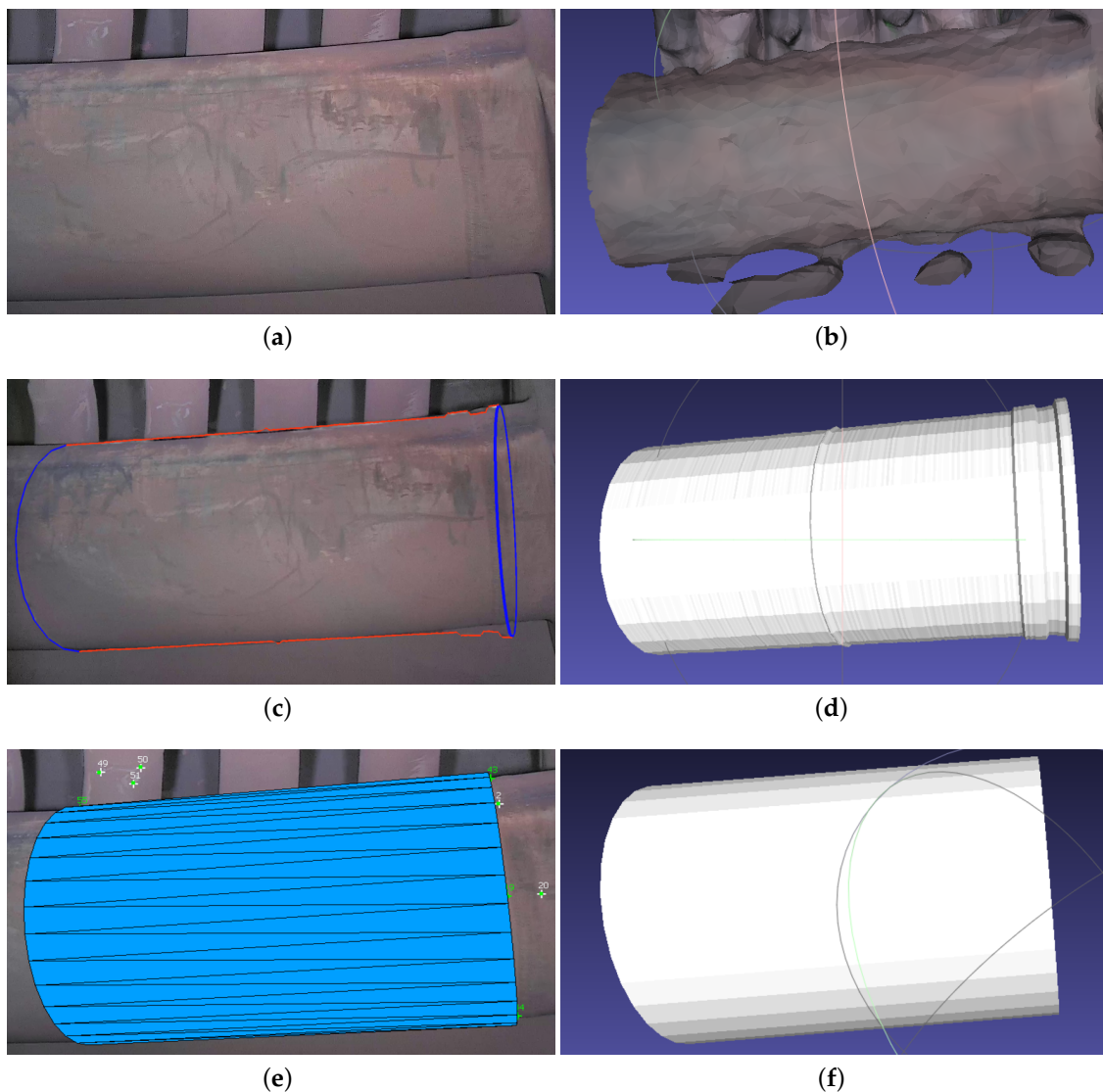


**Figure 5.** Jet pump beam is modeled with tested software programs under consideration: (a) Metashape reconstruction output; (b) Another view of (a); (c) Paint strokes snapped to jet pump beam outline; (d) Output obtained by modeling jet pump beam with 3-Sweep; (e) Estimation of jet pump beam surface using quadrilateral tool in MoReLab; (f) Output obtained by modeling jet pump beam with MoReLab.

#### 4.3. Cylinder Modeling

Equipment of cylindrical shape is common in different industrial plants. We have also modeled a cylinder with our tested approaches, and the results have been presented in Figure 6. In the Metashape reconstruction of the cylinder in Figure 6b, some geometric artifacts are observed, and the surface is not smooth. Figure 6c,d show the result of using 3-Sweep. While the boundary detection is better than that in Figure 4c, the cylinder still does not have a smooth surface. On the other hand, the cylinder mesh obtained by modeling with MoReLab has a smoother surface and is more consistent than that obtained with 3-Sweep.

Figure 6e,f show the result of modeling a cylinder, using the base cylinder tool in MoReLab. The reason to use this specific tool is that the center of the cylinder base is not visible, and features are visible only on the surface of the cylinder. As just stated, the cylinder obtained is more consistent and smooth than the one obtained with Metashape and 3-Sweep.

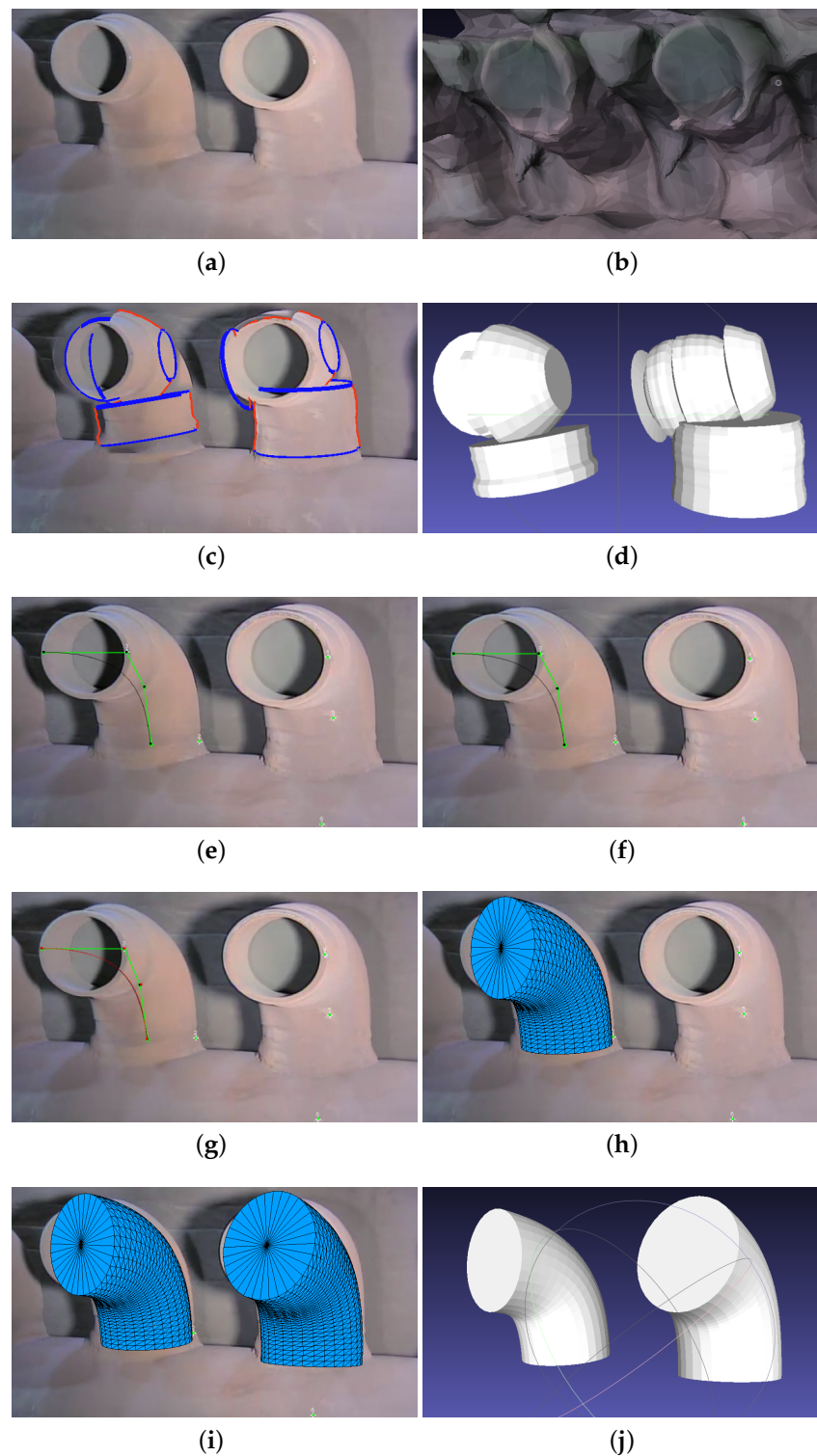


**Figure 6.** An example of modeling a cylinder with Metashape, 3-Sweep, and MoReLab: (a) A frame of input video; (b) MeshLab visualization of a cylinder created using Metashape; (c) Paint strokes snapped to cylinder outline in 3-Sweep; (d) MeshLab visualization of a cylinder modeled using 3-Sweep; (e) Modeling a cylinder using base cylinder tool in MoReLab; (f) MeshLab visualization of a cylinder mesh obtained from MoReLab.

#### 4.4. Curved Pipe Modeling

Figure 7 compares the modeling of curved pipes in Metashape, 3-Sweep, and MoReLab. In general, the reconstruction of curved pipes is difficult due to the lack of features. Figure 7b shows the result of modeling curved pipes using Metashape. The result is extremely low-quality because background walls are merged with the pipes, and visually similar pipes produce different results. The result of using 3-Sweep is shown in Figure 7c. As shown in Figure 7d, the mesh obtained with 3-Sweep hardly reflects the original pipe. Due to discontinuous outline detection and curved shape, multiple straight cylinders are estimated to model a single curved pipe.





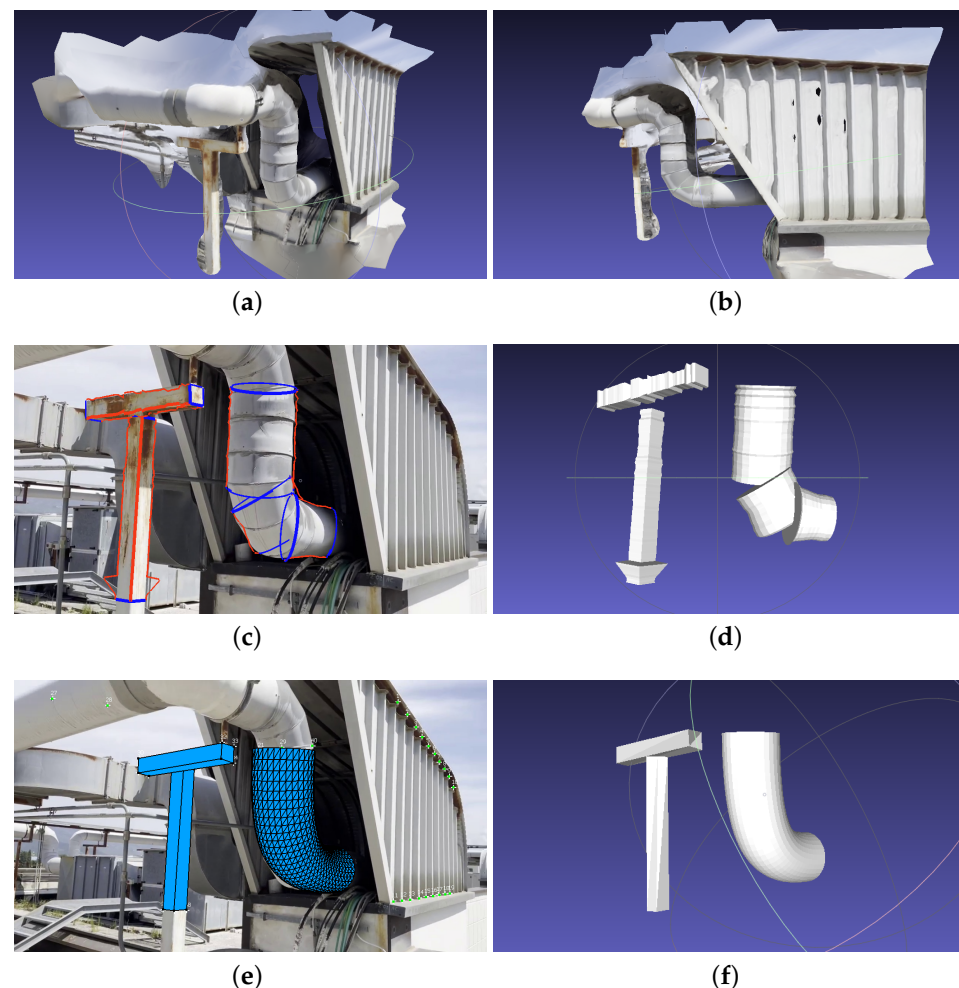
**Figure 7.** An example of modeling a curved pipe: (a) A frame of input video; (b) Modeling curved pipes in Metashape; (c) Paint strokes snapped to curved cylinder outlines; (d) Estimation of curved pipes using 3-Sweep visualized in MeshLab; (e) Bézier curve is drawn on a frame; (f) Bézier curve is drawn on another frame; (g) Curves on multiple frames are optimized to obtain the final Bézier curve shown by red color; (h) A cylinder around the curve is created; (i) A copy of the first cylinder is placed on the second pipe; (j) Estimated curved cylinders are visualized in MeshLab.

Figure 7e–j show the steps of modeling two curved cylinders in MoReLab. The results are quite good, even if there is a small misalignment between the pipes and the underlying frame.

#### 4.5. Additional Experiments

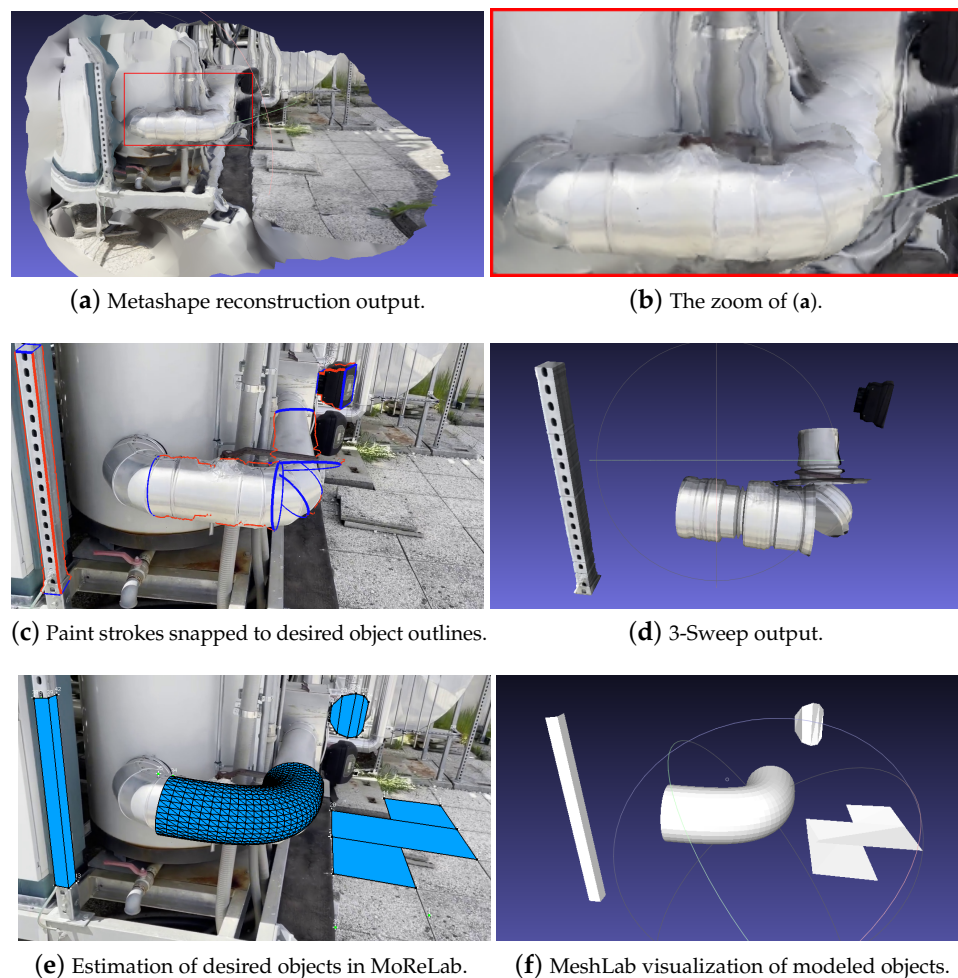
After observing the results of the data provided by the utility company, we captured a few more videos to conduct additional experiments and better evaluate our approach. These videos are captured on the roof of our research institute, which is full of steel pipes and other featureless objects.

Figure 8 shows the result of modeling a video with Metashape, 3-Sweep, and MoReLab. While the overall 3D model obtained with Metashape (Figure 8a) looks good, a visual examination of the same model from a different viewpoint (Figure 8b) shows that the T-shaped object and curved pipe lack a surface from behind. This can be due to the lack of a sufficient number of features and views at the back side of the T-shaped object and curved pipe. 3-Sweep output in Figure 8d shows gaps in 3D models of T-shaped object and curved pipe. As shown in Figure 8e,f, MoReLab is able to model desired objects more accurately, and a fine mesh can be exported easily from MoReLab.



**Figure 8.** Modeling cuboids and a curved pipe with tested software programs: (a) Metashape reconstruction output visualized in MeshLab; (b) A different view of the Metashape reconstruction visualized in MeshLab; (c) Paint strokes snapped to desired object outlines; (d) 3-Sweep output visualized in MeshLab; (e) Estimation of desired objects in MoReLab; (f) Estimated objects are visualized in MeshLab.

Figure 9 shows the result of modeling another video. Metashape output (see Figure 9a) shows a high level of approximation. The red rectangular region represents the curved pipe in the frame, and Figure 9b shows the zoom-in of this rectangular region. The lack of a smooth surface reduces the recognizability of the pipe and introduces inaccuracies in the measurements. Figure 9d shows gaps in the 3D output model of a curved pipe. However, outputs obtained with MoReLab are more accurate and represent the underlying objects more accurately.



**Figure 9.** Modeling cuboids and a curved pipe with Metashape, 3-Sweep, and MoReLab.

#### 4.6. Discussion

The results obtained with SfM packages (e.g., see Figures 4b, 6b, 7b, 8b, and 9a) elicit the need to identify features manually and develop software for user-assisted reconstruction. The reason for low-quality output models obtained using 3-Sweep can be attributed to low-quality border detection. This is due to dark light conditions in these low-resolution images. 3-Sweep modeled high-resolution images in their paper and reported high-quality results in their work for high-quality images. However, our experiments indicate that 3-Sweep is not suitable for low-resolution images and industrial scenarios mentioned in Figure 1. In these difficult scenarios, 3-Sweep suffers from low robustness and irregularity in the shapes of meshes.

MoReLab does not rely on the boundary detection stage and hence generates more robust results. After computing sparse 3D points on the user-provided features, our software provides tools to the user to quickly model objects of different shapes. Figures 4f, 5e, 6e, 7i, 8e, and 9e demonstrate the effectiveness of our software by showing the results obtained with our software tools.

#### 4.7. Measurement Results

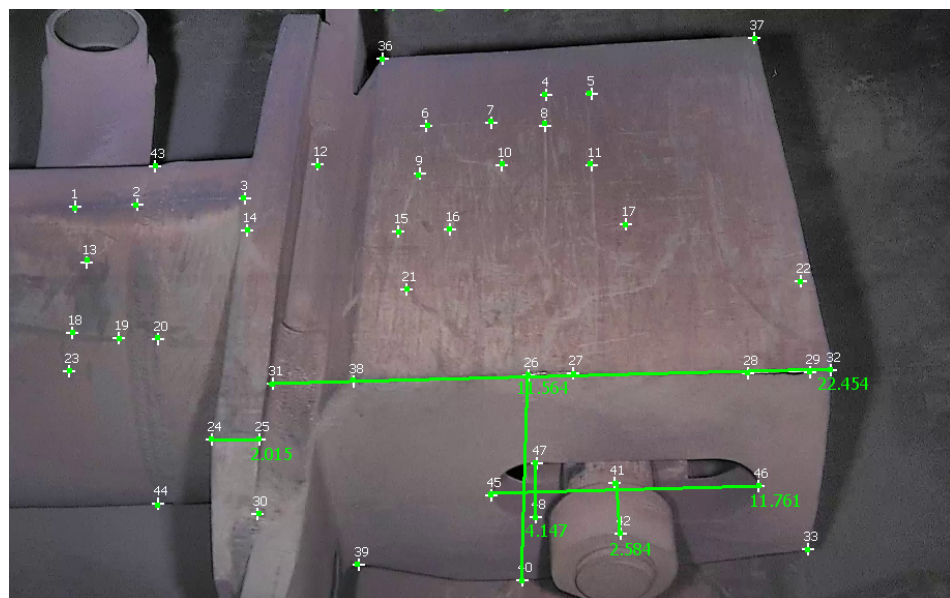
Given the availability of ground-truth data for two videos in the first dataset, we performed a quantitative analysis. The evaluation metric being used for quantitative analysis, is a relative error,  $E_{rel}$ :

$$E_{rel} = 100 \times \left( \frac{|M_e - M_g|}{M_g} \right), \quad (6)$$

where  $M_g$  is the ground-truth measurement, and  $M_e$  is a measure length from the estimated 3D model.

##### 4.7.1. 1-Measurement Calibration

In this section, we perform calibration with one ground-truth measurement. In all experiments, the longest measurement was taken as ground truth to have a more stable reference measure. This helps in mitigating the error of the calculated measurements. Table 2 reports measurements obtained with the different approaches on a video of the first dataset, and Figure 10 shows these measurements taken in MoReLab.



**Figure 10.** Measurements are taken in MoReLab. The distance of 22.454 cm between features 31 and 32 is the measurement provided for calibration. The other distances are calculated according to this reference distance.

The selection of measurements has been done according to the available ground-truth measurements from diagrams of equipment. Table 2 also presents a comparison of relative errors with these three software packages. Among the five measurements under consideration, MoReLab achieves the lowest errors in three measurements and the lowest average relative error.

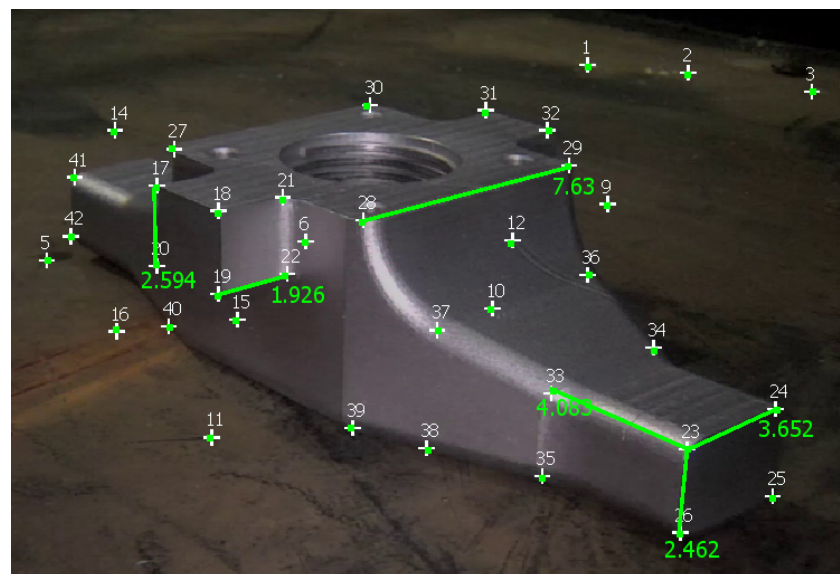
Table 3 reports measurements obtained with Metashape, 3-Sweep, and MoReLab on another video of the first dataset, and Figure 11 shows these measurements taken in MoReLab. Given the availability of a CAD model for the jet pump, we take meaningful measurements between corners in a CAD file and use these measurements as ground truths. Table 3 also presents a comparison of relative errors with these three software packages. Among the five measurements under consideration, MoReLab achieves the lowest errors in three measurements and the lowest average relative error.



**Table 2.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the first video (see Figure 10).

Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	14.046	13.472	4.087
	12.395	9.809	20.857
	4.115	2.664	35.201
	2.616	6.644	23.136
	2.057	1.852	9.889
	Average Relative Error		
3-Sweep	14.046	13.858	<b>1.447</b>
	12.3953	12.669	<b>2.213</b>
	4.115	4.475	8.765
	2.616	3.731	42.621
	2.057	1.338	34.938
	Average Relative Error		
MoReLab	14.046	11.564	17.672
	12.3953	11.761	5.117
	4.115	4.147	<b>0.783</b>
	2.616	2.584	<b>1.231</b>
	2.057	2.015	<b>2.061</b>
	Average Relative Error		

Table 4 reports measurements and calculations for a video of the second dataset, and Figure 12 illustrates these measurements in MoReLab. We took some meaningful measurements to be used as ground truth for measurements with Metashape, 3-Sweep, and MoReLab. Relative errors are also calculated for these measurements and reported in Table 4. All software programs have achieved more accurate measurements in this video with respect to videos of the first dataset. This can be due to more favorable lighting conditions and high-resolution frames containing a higher number of recognizable features. Similar to Tables 2 and 3, five measurements have been considered and MoReLab achieves the lowest relative errors in three measurements and the lowest average relative error in comparison to other software programs.

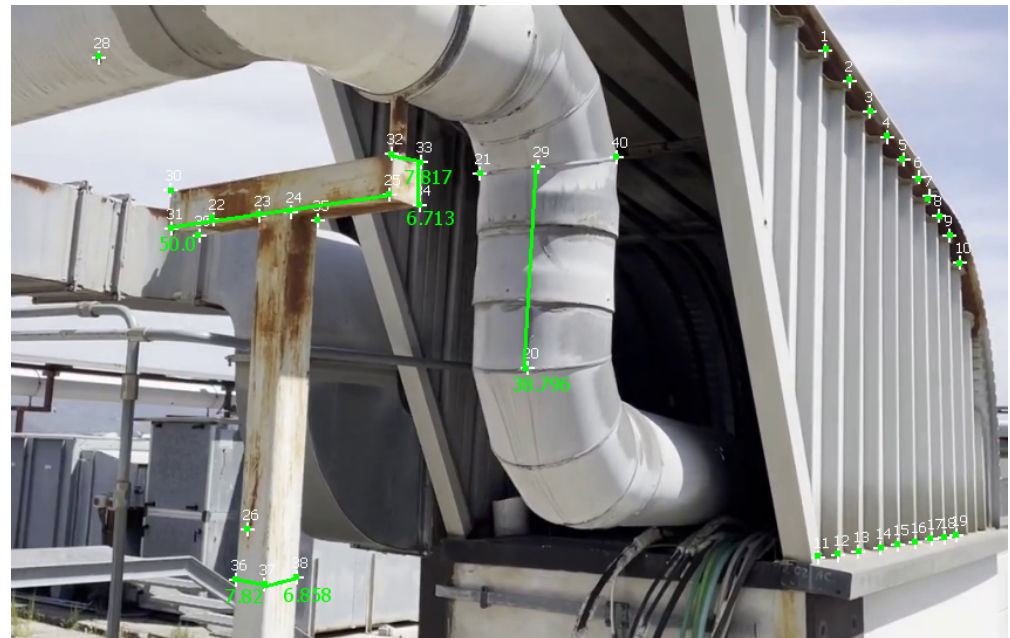


**Figure 11.** Measurements computed in MoReLab. The distance of 7.630 cm between features 28 and 29 is the measurement provided for calibration, and other distances are calculated.

**Table 3.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the second video (see Figure 11).

Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	3.355	4.161	24.011
	3.216	3.109	<b>3.316</b>
	2.365	2.532	<b>7.073</b>
	2.251	2.626	16.688
	1.923	2.045	6.344
	Average Relative Error		11.486
3-Sweep	3.355	2.388	28.833
	3.216	2.868	10.817
	2.365	1.954	17.374
	2.251	1.905	15.359
	1.923	1.264	34.249
	Average Relative Error		21.326
MoReLab	3.355	4.083	<b>21.687</b>
	3.216	3.652	13.570
	2.365	2.594	9.695
	2.251	2.462	<b>9.401</b>
	1.923	1.926	<b>0.15</b>
	Average Relative Error		<b>10.902</b>

Table 5 reports measurements obtained with Metashape, 3-Sweep, and MoReLab on another video of the second dataset, and Figure 13 illustrates these measurements in MoReLab. Among the five measurements under consideration, MoReLab achieves minimum error in four measurements and the lowest average relative error.



**Figure 12.** Measurements computed in MoReLab. The distance of 50 cm between features 25 and 31 is the measurement provided for calibration, and other distances are calculated.



**Table 5.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the fourth video (see Figure 13).

Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	24	24.45	<b>1.873</b>
	17.5	15.959	8.843
	5	3.558	28.852
	4.2	3.528	15.974
	3.5	4.016	14.741
	Average Relative Error		
3-Sweep	24	21.618	9.927
	17.5	12.287	29.817
	5	3.685	26.289
	4.2	5.228	24.461
	3.5	3.815	9.221
	Average Relative Error		
MoReLab	24	21.51	10.375
	17.5	16.739	<b>4.349</b>
	5	4.592	<b>8.16</b>
	4.2	3.575	<b>14.881</b>
	3.5	3.621	<b>3.457</b>
	Average Relative Error		

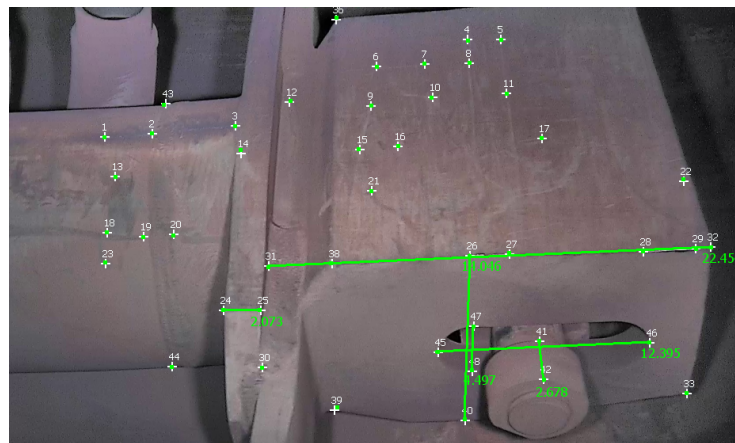
**Table 6.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the first video seen in Figure 14. The 1-measurement calibration table corresponding to this one is Table 2.

Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	4.115	2.918	29.085
	2.616	2.213	15.412
	2.057	1.864	9.4
	Average Relative Error		
3-Sweep	4.115	4.549	10.552
	2.616	3.077	17.613
	2.057	1.632	20.677
	Average Relative Error		
MoReLab	4.115	4.497	<b>9.288</b>
	2.616	2.678	<b>2.362</b>
	2.057	2.073	<b>0.758</b>
	Average Relative Error		

#### 4.7.3. Limitations

From our evaluation, we have shown that our method performs better than other approaches for our scenario of industrial plants. However, users need to be accurate and precise when adding feature points and to use a high-quality methodology when performing measurements. Overall, all image-based 3D reconstruction methods, including ours, cannot achieve a precision of millimeters (at our scale) or less for many factors (e.g., sensor resolution). Therefore, if an object has a small scale the error introduced by the tolerance is lower than the reconstruction error.

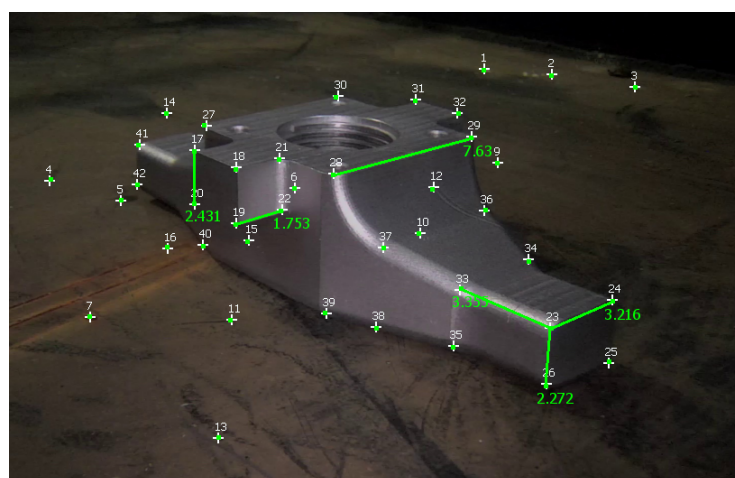




**Figure 14.** Measurements computed in MoReLab. The distances of 22.454, 14.046, and 12.395 cm are provided for calibration, and other distances are calculated.

**Table 7.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the second video seen in Figure 15. The 1-measurement calibration table corresponding to this one is Table 3.

Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	2.25	2.535	12.645
	1.923	2.07	<b>7.644</b>
	2.365	2.512	6.227
	Average Relative Error		8.839
3-Sweep	2.25	2.375	5.535
	1.923	1.554	19.189
	2.365	2.202	6.878
	Average Relative Error		10.534
MoReLab	2.25	2.272	<b>0.958</b>
	1.923	1.753	8.84
	2.365	2.431	<b>2.802</b>
	Average Relative Error		<b>4.20</b>



**Figure 15.** Measurements computed in MoReLab. The distances of 7.63, 3.355, and 3.216 cm are provided for calibration, and other distances are calculated.

**Table 8.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the third video seen in Figure 16. The 1-measurement calibration table corresponding to this one is Table 4.

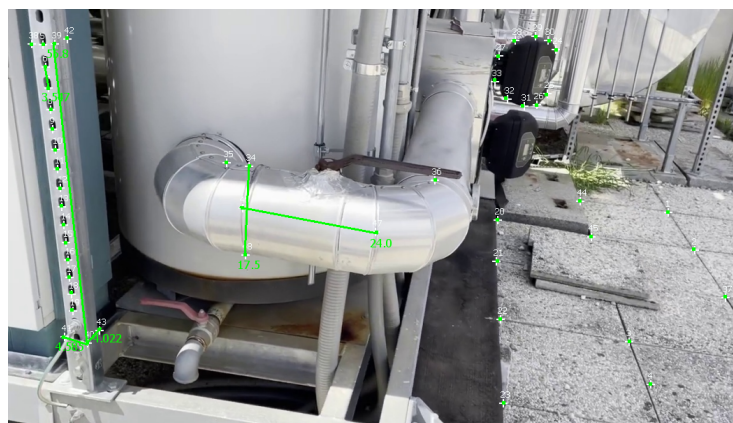
Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	6.9	5.649	18.13
	6.8	6.482	4.676
	6.7	6.447	3.776
	Average Relative Error		8.861
3-Sweep	6.9	6.962	<b>0.899</b>
	6.8	7.940	16.765
	6.7	6.332	5.493
	Average Relative Error		7.719
MoReLab	6.9	7.41	7.391
	6.8	6.553	<b>3.632</b>
	6.7	6.776	<b>1.134</b>
	Average Relative Error		<b>4.052</b>



**Figure 16.** Measurements computed in MoReLab. The distances of 50, 35, and 7 cm are provided for calibration, and other distances are calculated.

**Table 9.** Results of comparing MoReLab against Metashape and 3-Sweep in terms of relative error in measurements on the fourth video seen in Figure 17. The 1-measurement calibration table corresponding to this one is Table 5.

Method	Ground Truth (cm)	Measured Distance (cm)	Relative Error
Metashape	5	3.965	20.7
	4.2	3.164	24.667
	3.5	3.894	11.257
	Average Relative Error		18.875
3-Sweep	5	3.787	24.26
	4.2	4.991	18.833
	3.5	3.767	7.629
	Average Relative Error		16.907
MoReLab	5	4.585	<b>8.3</b>
	4.2	4.022	<b>4.238</b>
	3.5	3.547	<b>1.343</b>
	Average Relative Error		<b>4.627</b>



**Figure 17.** Measurements computed in MoReLab. The distances of 55.8, 24, and 17.5 cm are provided for calibration, and other distances are calculated.

## 5. Conclusions

We have developed a user-interactive 3D reconstruction tool for modeling low-quality videos. MoReLab can handle long videos and is well-suited to model featureless objects in videos. It allows the user to load a video, extract frames, mark features, estimate the 3D structure of the video, add primitives (e.g., quads, cylinders, etc.), calibrate, and perform measurements. These functionalities lay the foundations of the software and present a general picture of its use. MoReLab allows users to estimate shapes that are typical of industrial equipment (e.g., cylinders, curved cylinders, etc.) and measure them. We evaluated our tool for several scenes and compared results against the automatic SfM software program, Metashape, and another modeling software, 3-Sweep [16]. Such comparisons show that MoReLab can generate 3D reconstructions from low-quality videos with less relative error than these state-of-the-art approaches. This is fundamental in the industrial context when there is the need to obtain measurements of objects in difficult scenarios, e.g., in areas with chemical and radiation hazards.

In future work, we plan to extend MoReLab tools for modeling more complex industrial equipment and to show that we are not only more effective than other state-of-the-art approaches in terms of measurement errors but also more efficient in terms of the time that the user needs to spend to achieve an actual reconstruction.

**Author Contributions:** Conceptualization, A.S., F.B., M.C. and P.C.; methodology, A.S., F.B., M.C. and P.C.; experiments, A.S.; validation, A.S., F.B. and M.C.; formal analysis, A.S., F.B. and M.C.; investigation, A.S., F.B. and M.C.; resources, F.B., M.C., D.S. and C.J.; data curation, A.S., F.B., P.C., D.S. and C.J.; writing—original draft preparation, A.S.; writing—review and editing, F.B., M.C., D.S. and C.J.; supervision, F.B., M.C. and P.C.; project administration, F.B., P.C., D.S. and C.J.; funding acquisition, P.C., D.S. and C.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been supported by European Union’s Horizon 2020 Programme EVOCA-TION: Advanced Visual and Geometric Computing for 3D Capture, Display, and Fabrication (Grant Agreement No. 813170) and a research contract with Electric Power Research Institute.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The first dataset presented in this study is not available; but, the second dataset can be provided on request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

SfM	Structure from Motion
MoReLab	Movie Reconstruction Laboratory

## References

- Vacca, G. 3D Survey with Apple LiDAR Sensor &mdash;Test and Assessment for Architectural and Cultural Heritage. *Heritage* **2023**, *6*, 1476–1501. [[CrossRef](#)]
- Rocchini, C.; Cignoni, P.; Montani, C.; Pingi, P.; Scopigno, R. A low cost 3D scanner based on structured light. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2001; Volume 20, pp. 299–308.
- Pollefeys, M.; Koch, R.; Vergauwen, M.; Van Gool, L. Hand-Held Acquisition of 3D Models with a Video Camera. In Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling, Ottawa, ON, Canada, 8 October 1999; pp. 14–23. [[CrossRef](#)]
- Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
- Rupnik, E.; Daakir, M.; Pierrot Deselligny, M. MicMac—A free, open-source solution for photogrammetry. *Open Geospat. Data Softw. Stand.* **2017**, *2*, 14. [[CrossRef](#)]
- Cernea, D. OpenMVS: Multi-View Stereo Reconstruction Library. *City* **2020**, *5*, 7.
- Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle adjustment—A modern synthesis. In Proceedings of the International Workshop on Vision Algorithms, Corfu, Greece, 21–22 September 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 298–372.
- Van Den Hengel, A.; Dick, A.; Thormählen, T.; Ward, B.; Torr, P.H. Videotrace: Rapid interactive scene modelling from video. *ACM Trans. Graph. (TOG)* **2007**, *26*, 86–es
- Sinha, S.N.; Steedly, D.; Szeliski, R.; Agrawala, M.; Pollefeys, M. Interactive 3D architectural modeling from unordered photo collections. *ACM Trans. Graph. (TOG)* **2008**, *27*, 1–10. [[CrossRef](#)]
- Xu, M.; Li, M.; Xu, W.; Deng, Z.; Yang, Y.; Zhou, K. Interactive mechanism modeling from multi-view images. *ACM Trans. Graph. (TOG)* **2016**, *35*, 1–13. [[CrossRef](#)]
- Rasmuson, S.; Sintorn, E.; Assarsson, U. User-guided 3D reconstruction using multi-view stereo. In Proceedings of the Symposium on Interactive 3D Graphics and Games, San Francisco, CA, USA, 5–7 May 2020; pp. 1–9.
- Habbecke, M.; Kobbelt, L. An Intuitive Interface for Interactive High Quality Image-Based Modeling. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2009; Volume 28, pp. 1765–1772.
- Baldacci, A.; Bernabei, D.; Corsini, M.; Ganovelli, F.; Scopigno, R. 3D reconstruction for featureless scenes with curvature hints. *Vis. Comput.* **2016**, *32*, 1605–1620. [[CrossRef](#)]
- Doron, Y.; Campbell, N.D.; Starck, J.; Kautz, J. User directed multi-view-stereo. In Proceedings of the Computer Vision-ACCV 2014 Workshops, Singapore, 1–2 November 2014; Revised Selected Papers, Part II 12; Springer: Berlin/Heidelberg, Germany, 2015; pp. 299–313.
- Töppe, E.; Oswald, M.R.; Cremers, D.; Rother, C. Image-based 3d modeling via cheeger sets. In Proceedings of the Computer Vision—ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010; Revised Selected Papers, Part I 10; Springer: Berlin/Heidelberg, Germany, 2011; pp. 53–64.
- Chen, T.; Zhu, Z.; Shamir, A.; Hu, S.M.; Cohen-Or, D. 3-sweep: Extracting editable objects from a single photo. *ACM Trans. Graph. (TOG)* **2013**, *32*, 1–10. [[CrossRef](#)]
- Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* **2021**, *65*, 99–106. [[CrossRef](#)]
- Tewari, A.; Thies, J.; Mildenhall, B.; Srinivasan, P.P.; Trevischi, E.; Wang, Y.; Lassner, C.; Sitzmann, V.; Martin-Brualla, R.; Lombardi, S.; et al. Advances in Neural Rendering. *Comput. Graph. Forum* **2022**, *41*, 703–735. [[CrossRef](#)]
- Chan, E.R.; Monteiro, M.; Kellnhofer, P.; Wu, J.; Wetzstein, G. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5799–5809.
- Tu, Z.; Huang, Z.; Chen, Y.; Kang, D.; Bao, L.; Yang, B.; Yuan, J. Consistent 3d hand reconstruction in video via self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9469–9485. [[CrossRef](#)] [[PubMed](#)]
- Longuet-Higgins, H.C. A computer algorithm for reconstructing a scene from two projections. *Nature* **1981**, *293*, 133–135. [[CrossRef](#)]
- Banterle, F.; Gong, R.; Corsini, M.; Ganovelli, F.; Gool, L.V.; Cignoni, P. A Deep Learning Method for Frame Selection in Videos for Structure from Motion Pipelines. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 3667–3671. [[CrossRef](#)]
- Nocerino, E.; Lago, F.; Morabito, D.; Remondino, F.; Porzi, L.; Poiesi, F.; Rota Bulò, S.; Chippendale, P.; Locher, A.; Havlena, M.; et al. A Smartphone-Based 3D Pipeline for the Creative Industry the Replicate Eu Project. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-2/W3*, 535–541. [[CrossRef](#)]



24. Branch, M.A.; Coleman, T.F.; Li, Y. A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems. *SIAM J. Sci. Comput.* **1999**, *21*, 1–23. [[CrossRef](#)]
25. Gordon, W.J.; Riesenfeld, R.F. Bernstein-BéZier Methods for the Computer-Aided Design of Free-Form Curves and Surfaces. *J. ACM* **1974**, *21*, 293–310. [[CrossRef](#)]
26. Cignoni, P.; Callieri, M.; Corsini, M.; Dellepiane, M.; Ganovelli, F.; Ranzuglia, G. Meshlab: An open-source mesh processing tool. In Proceedings of the Eurographics Italian Chapter Conference, Salerno, Italy, 2–4 July 2008; Volume 2008, pp. 129–136.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.