



Towards Open-Source Web-Based 3D Reconstruction for Non-Professionals

Oliver Dietz and Jens Grubert*

Department of Electrical Engineering and Computer Science, Coburg University of Applied Sciences and Arts, Coburg, Germany

Structure-from-motion and multi-view stereo can be used to create 3D models from image sequences, but are not widely adopted by consumers. We study how to address two challenges of such systems for non-professional users: 1) their technical complexity, and, 2) the computational demand needed for processing. To this end, we embed an open-source pipeline in a scalable cloud environment and create a user interface aimed at non-professional users of photogrammetry systems. Finally, we evaluate both the cloud-based infrastructure and the user interface and demonstrate its usability.

Keywords: 3D reconstruction, structure-from-motion, usability, cloud computing, user interface, multi-view stereo, extended reality

OPEN ACCESS

Edited by:

Christian Richardt,
University of Bath, United Kingdom

Reviewed by:

Weiya Chen,
Huazhong University of Science and
Technology, China
Mansi Sharma,
Indian Institute of Technology Madras,
India

*Correspondence:

Jens Grubert
jens.grubert@hs-coburg.de

Specialty section:

This article was submitted to
Technologies for VR,
a section of the journal
Frontiers in Virtual Reality

Received: 30 September 2021

Accepted: 14 December 2021

Published: 03 February 2022

Citation:

Dietz O and Grubert J (2022) Towards
Open-Source Web-Based 3D
Reconstruction for Non-Professionals.
Front. Virtual Real. 2:786558.
doi: 10.3389/frvir.2021.786558

1 INTRODUCTION

The ability to create 3D models from real-world objects and environments is a key enabler for content production for Extended Reality. Traditional structure-from-motion and multi-view stereo approaches are technically advanced and have the potential to be used for 3D content creation with millions of existing mobile devices without the need for specialized sensors beyond RGB cameras. Specifically, the majority of capture devices available to consumers, such as smartphones or tablets still employ monoscopic cameras (with exceptions such as the Apple iPhone 12 Pro). While research on extracting 3D information from single views is on the rise Fahim et al. (2021); Watson et al. (2021) classic photogrammetry pipelines are still worthwhile to be considered. However, while both open-source and commercial photogrammetry software exists, they are not widely adopted by consumers.

Two challenges arise, when trying to support non-expert users in creating 3D models from image sequences, namely *capturing the image sequences* and *processing them to a 3D model*. Within this work, we concentrate on the later challenge. Specifically, we study how to address two challenges of processing image sequences into a 3D model for non-professional users: 1) the technical complexity of the process, and, 2) the computational demand needed for processing, typically requiring access to GPU accelerated computing hardware.

To address those challenges, we embed an open-source pipeline in a scalable cloud environment and create a user interface aimed at non-professional users of photogrammetry systems and evaluate both the user interface and cloud environment. Our work indicates that it is possible to create usable 3D reconstruction interfaces for novice users, but also that the computational costs are substantial, which can become an obstacle when trying to offer 3D reconstruction services pro bono publico.

The remainder of this paper is structured as follows. First, we embed our approach into related work. Then, we derive, describe and evaluate the user interface for non-expert users. We then present our cloud-based reconstruction architecture and evaluate its running costs and conclude the paper with a discussion.

2 RELATED WORK

Our work is embedded in the large area of 3D reconstruction (Gomes et al. (2014); Iglhaut et al. (2019); Jiang et al. (2020); Zhao et al. (2020)).

When aiming at making photogrammetry workflows accessible to non-expert users, they typically need support in *capturing the image sequences* and *processing them to a 3D model*. The first issue is commonly addressed using view planning or guidance techniques Hoppe et al. (2012); Langguth and Goesele (2013); Locher et al. (2016); Andersen et al. (2019); Isabelle and Laurendeau (2020) and available in selected commercial applications such as SCANN3D1 or TRNIO2. For the scope of our work, we assume that such guidance techniques would be available for capturing image streams. The second issue (processing image sequences into a 3D model) relies on underlying technical complex structure-from-motion and multi-view stereo pipelines. While modern smartphone can locally process 3D models using structure-from-motion and multi-view-stereo, the limited computational resources compared to a dedicated compute server typically limit the scope of 3D reconstructions. Also, many open-source solutions such as AliceVision or OpenMVG as well as commercial solutions such as Metashape, RealityCapture or Pix4DMapper exist, which can be deployed on dedicated computing hardware such as PCs or compute servers with powerful GPUs. However, they are typically cumbersome to use for beginners due to the sheer complexity of parameters that can influence the 3D reconstruction outcome as well as hardware requirements for computing complex 3D models consisting of potentially hundreds of images.

Hence, prior work has investigated on how to simplify the reconstruction process for non-expert users through web-based services. In Heller et al. (2015), a Structure from Motion pipeline was provided as a web service which was developed at the Center of Machine Perception (CMP) of the Czech Technical University in Prague. The servers for the web service were hosted at the same location. The service can be used either *via* a CLI client or web browser. With the latter users can upload images to the web page interface and start a 3D reconstruction job. There are 21 different job types each with a different task for the reconstruction. A “One-Button” feature allows to skip the configuration of all parameters and use default settings. A similar approach was proposed in Laksono (2016). There, a new Structure from Motion pipeline was developed with existing libraries and combined with a single page application as the interface. Registered users can upload images, specify options such as feature detector and camera intrinsics, and perform the 3D reconstruction on the server. The work elaborates on the term “cloud computing” but does not take advantage of workload distribution. Yet another web-based approach was presented in Vergauwen and Van Gool (2006). Though, in this case, the 3D reconstruction is not accessed *via* a web browser but with two separate desktop applications: An upload tool and a model-viewer tool. The upload tool lets the user select, preview, and finally upload the images to start a 3D reconstruction on a remote cluster of servers. The remote reconstruction consists only of

camera calibration, depth map, and quality map. The results can be downloaded as a bundle via FTP and viewed via the model-viewer tool. Users can also triangulate the 3D model in the model-viewer. While 3D reconstruction is feasible on modern smartphones Ondruska et al. (2015) multiple approaches combined a server-based reconstruction back-end, with capturing front-ends on mobile devices (e.g., Muratov et al. (2016); Fleck et al. (2016); Poiesi et al. (2017)) due to the (relatively) larger computational power of dedicated servers.

Focusing on open-source 3D reconstruction, Stathopoulou et al. compare three common open-source image-based 3D reconstruction pipelines (OpenMVG, COLMAP, AliceVision, which we also employ in our project) Stathopoulou and Remondino (2019). The aim of their research was to investigate algorithm reliability and performance on large data sets. Julin et al. propose to combine photogrammetry and terrestrial laser scanning, focusing on generating 3D models that are compatible with web-based 3D viewers Julin et al. (2019). To this end, they employ a commercial product (RealityCapture) and evaluate its performance for photogrammetry-based, laser-scanning based and hybrid 3D reconstruction. Bouck-Standen et al. introduce the *NEMO Converter 3D* to reconstruct 3D objects from annotated media. In addition, they introduce a web-based interface to edit the resulting 3D models (cropping for removing unwanted parts of the reconstruction, and reorienting) but do not evaluate it in a user study Bouck-Standen et al. (2018). Sheng et al. introduce a set of algorithms focusing on creating compact surface reconstructions suitable for 3D printing through a computationally efficient process Sheng et al. (2018). The data is stored in a cloud-based server and the point cloud data can be viewed through an web-based interface. Their evaluation concentrates on reconstruction quality. No formal user study was carried out.

Our approach complements these prior works by specifically investigating challenges relevant for non-expert users.

3 AN OPEN-SOURCE WEB-BASED 3D RECONSTRUCTION PIPELINE

Existing software for creating 3D models using structure-from-motion and multi-view stereo are typically aimed at expert users who are aware of relevant parameters (such as number of keyframes, or descriptor types) that can influence the reconstruction as well as have access to sufficient computational resources.

Our goal was to develop and evaluate a scalable and open-source 3D reconstruction pipeline that can be easily used by non-professionals who do not have the technical knowledge required for photogrammetry software. Our goals were 1) to ease the use of AliceVision for non-experts and 2) reduce the local computational demand needed for processing.

Our system consists of three main components: 1) photogrammetry software, 2) a web-based user interface, and 3) a cloud based computing environment.

As photogrammetry pipelines are already matured, we decided on relying on an existing pipeline as a technical basis. To this end, we chose AliceVision, an open-source photogrammetry pipeline

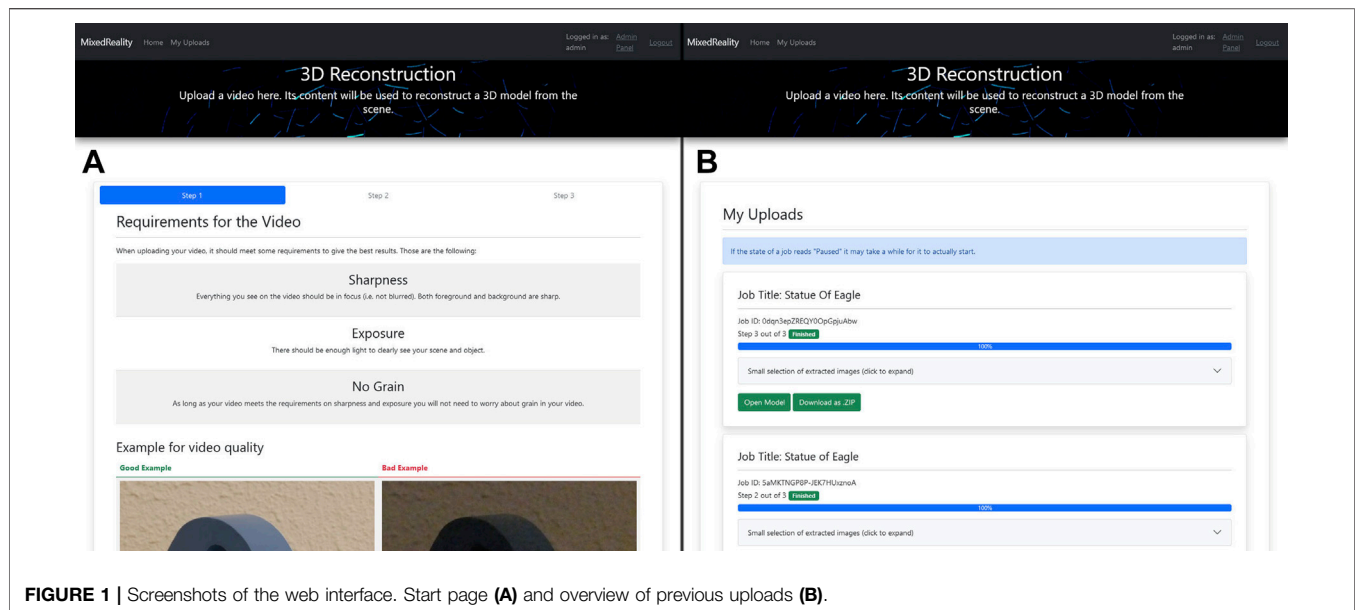


FIGURE 1 | Screenshots of the web interface. Start page (A) and overview of previous uploads (B).

under the Mozilla Public License, which allows for both sparse and dense 3D reconstruction using videos or a set of images as input. The development and evaluation of the two other components are described in the following.

4 USER INTERFACE FOR NON-EXPERTS

AliceVision is a photogrammetry pipeline consisting out of multiple steps (typically, camera initialization; feature extraction, description and matching; incremental or global structure-from-motion; depth-map estimation; meshing; texturing). Individual modules can be added, exchanged and parameterized. To ease the use of AliceVision, Meshroom can be used as the graphical user interface. It exposes the underlying photogrammetry workflow in a visual programming environment (using a dataflow graph) and allows for visualization of the (intermediate and final) result. In addition, it provides a standard parameterization of a photogrammetry workflow. To find out if the interface affords easy usage, we conducted a usability inspection of the interface.

4.1 Usability Inspection of Meshroom

We conducted a usability inspection in order to identify potential usability challenges when operating the open-source software Meshroom. To this end, we asked three usability experts (2 male, 1 female, mean age 32 years, $sd = 5.09$) to conduct a cognitive walkthrough of Meshroom. The expert users were asked to test the application with the mindset of an inexperienced user. This way, usability issues could be identified without the danger of aborting the workflow when inexperienced users would not know how to proceed. The cognitive walkthrough was conducted after the model of Rieman et al. (1995), with the following list as a guideline for each step in the analysis: 1) The user sets a goal to be accomplished with system. 2) The user searches the user interface for currently available actions (e.g., menu items, buttons,

command-line inputs). 3) The user selects the action that seems likely to make progress towards the goal. 4) The user performs the selected action and evaluates the system's feedback for evidence that progress is being made towards the current goal. The intended goals were to create a 3D reconstruction using a set of images as well as using a video.

The experts found the interface to be used easily in case of using image sets with the default reconstruction parameters (which involves importing a set of images into the user interface, pressing a button and waiting for the reconstruction process to finish). However, both processing videos as well as changing reconstruction parameters was found to have a low usability for laymen. For the former, the flow graph of the reconstruction needs to be exchanged with a keyframe selection node (performing image selection based on sharpness measures and frame distances), requiring an understanding of the concepts of keyframe extraction as well as additional steps to locate and re-import the extracted frames. For the latter, domain knowledge about photogrammetry (such as feature types or bundle adjustment) is required.

4.2 Web-Based User Interface

Based on the insights of the expert study, we decided to address the two main issues of the existing user interface: Support for video processing and ability to parameterize the 3D reconstruction without expert knowledge. Hence, we developed a web-interface following responsive-web design principles (to support access for a wide variety of devices). The process of the 3D reconstruction is split into three steps. Step one allows the user to upload a video whose frames will later be used for the 3D reconstruction. The according web page shows prerequisites concerning video quality and informs the user that the video quality is roughly determined by sharpness, exposure and noise which all influence the quality of the 3D model (see **Figure 1A**, and **Figure 2**). Based on the insights of the cognitive walkthrough and given that this interface was targeted at non-

Step 1 Step 2 Step 3

Requirements for the Video

When uploading your video, it should meet some requirements to give the best results. Those are the following:

Sharpness

Everything you see on the video should be in focus (i.e. not blurred). Both foreground and background are sharp.

Exposure

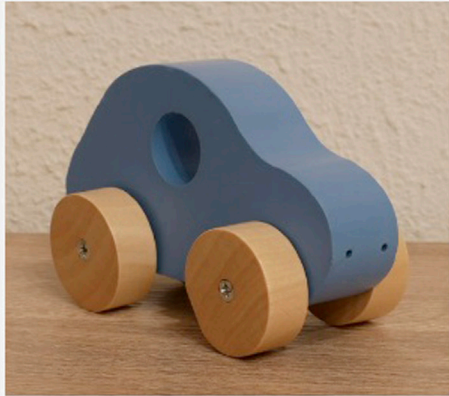
There should be enough light to clearly see your scene and object.

No Grain

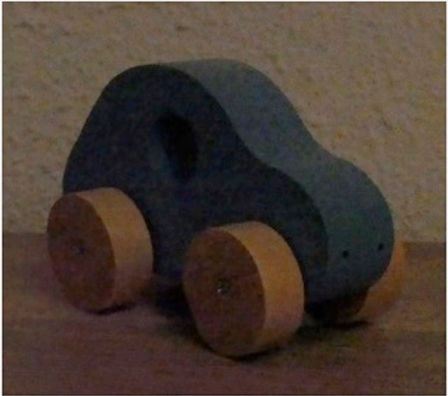
As long as your video meets the requirements on sharpness and exposure you will not need to worry about grain in your video.

Example for video quality

Good Example



Bad Example



Upload Video

Titel zum Video

Quality settings for the 3D model

Low quality will be the fastest option of all (usually within minutes) but with the worst result. High quality will give better results but can take up to several hours.

Low Quality
 Medium Quality
 High Quality

Advanced Options (for experts) ▼

Datei auswählen Keine ausgewählt

Upload

FIGURE 2 | Start page of the web interface including an example of suitable and unsuitable image frames.



FIGURE 3 | Sample frames of the video used for the user study.

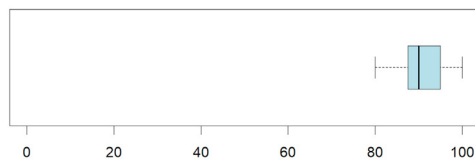


FIGURE 4 | Boxplot of the SUS scores. Mean: 90, sd = 5.7.

professionals, we deliberately chose to not expose the individual reconstruction parameters. Instead, the upload form consists of a title and a quality setting divided into low, medium and high quality (see **Appendix** for parameter mapping in AliceVision). After submitting, the user is redirected to step two which extracts the frames of the uploaded video. The extracted frames are filtered with a combination of blur detection and keyframe selection. A progress bar indicates the progress of that process. When finished, the user receives a message whether the extracted frames are likely usable or not. In both cases, the user can opt to try again with a different video or continue to step three. In this step, a selection of the extracted frames are represented to the user. A click on a “Start” button initiates the 3D reconstruction. The user is informed that the reconstruction can take up to several hours depending on the video length and quality settings. When started, the user is redirected to the “My Uploads” page which lists all previously uploaded videos as “jobs” (see **Figure 1B**). The current progress is shown for every job. The 3D model of every job that successfully finished can either be viewed in an integrated online 3D viewer (based on Three.js) or be downloaded as a ZIP file.

The website was developed using the Django Framework. It was chosen because it enforces high security standards by default, offers a built-in authentication system, templates, and supports object-relational mapping (ORM). It works with the Model-View-Controller (MVC) pattern, i.e., all pages (Step 1, Step 2, Step 3, My Uploads) are mapped to their own separate view-function in

Python which handles its logic and context. In each view-function, a template is assigned to the page. In Django a template is an extension of HTML which adds support for inheritance, variables, and simple functions. In this project, all templates are based off HTML5. All pages inherit from a master template which defines the design as well as the previously mentioned header section and content section. For the design the CSS framework Bootstrap was used to support responsive design. Updating the progress bars on the pages Step 2 and “My Uploads” occurs through polling the current status in fixed intervals via AJAX and helper functions from jQuery. The authentication system as well as the jobs are backed by a SQL database.

4.3 User Study

To evaluate if the designed web-interface addresses the complexity of photogrammetry software and if it offers an intuitive, easy-to-use interface for non-experts we conducted a usability study with 10 participants (8 male, 2 female, mean age 27.6, sd = 7.4). None of the participants had prior experience with 3D reconstructions. Participants rated their experience with photography on average as 3.6 (sd = 1.2) and their experience with online platforms on average as 4.4 (sd = 2.4) on a seven-item Likert scale ranging from 1: very inexperienced to 7: very experienced.

4.3.1 Procedure and Apparatus

The study took place in online meetings *via* Zoom due to pandemic restrictions. In the beginning, the subjects were given a handout containing general information about the study such as the topic, the goal, and instructions. The instructions stated that the web application takes a video as input and outputs a reconstructed 3D model representing the object recorded in that video. It also stated that the subjects are provided with a pre-recorded video of 10 seconds. With that, they were asked to iterate the steps of the web application. **Figure 3** shows sample frames of the employed video. A video of 10 seconds was chosen to keep the reconstruction time to a manageable level.

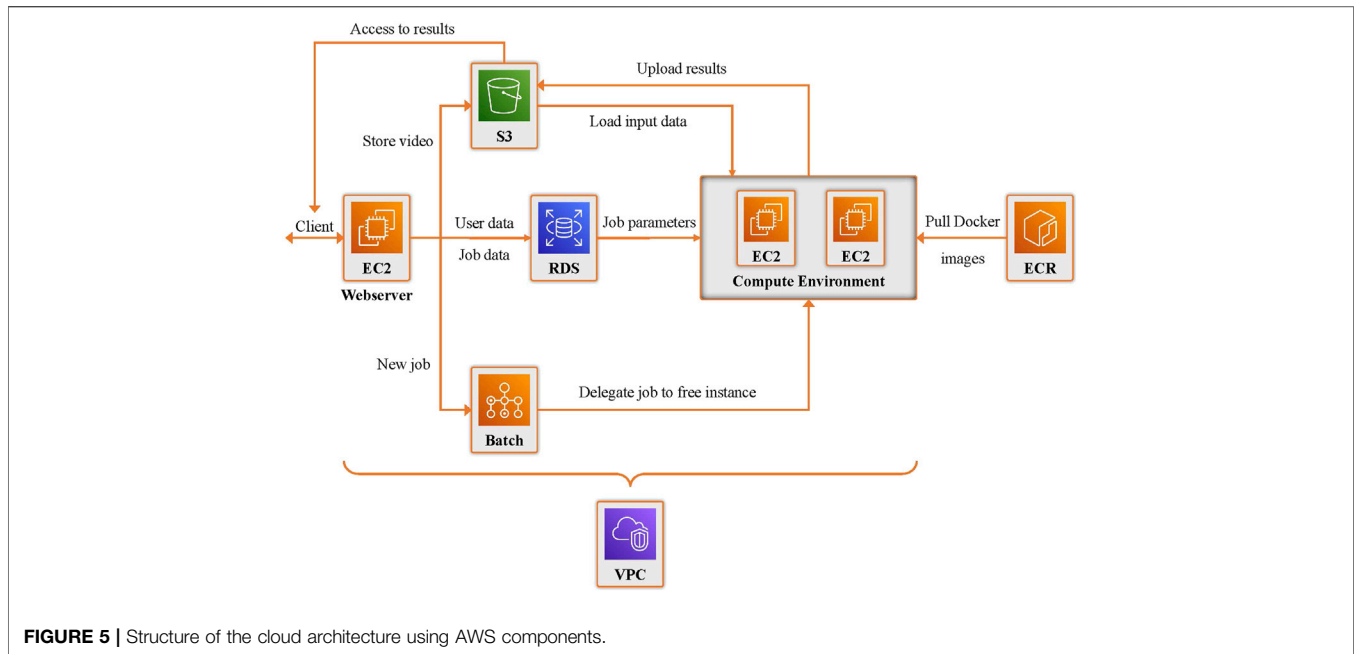


FIGURE 5 | Structure of the cloud architecture using AWS components.

During the study, subjects were asked to constantly verbally express what they think and feel about operating the web interface (think out loud). For the whole process the participant's screen and audio were recorded for later evaluation and notes were taken about their behavior while operating the interface. At the end, they were asked to fill out the System Usability Scale (SUS) Brooke (1996). After the SUS, subjects were asked to additionally fill out the following questions for feedback: (1) *This is what I liked . . .*, (2) *This is what I did not like . . .*, (3) *My suggestions for improvement . . .*, and (4) *I think my use cases could be . . .*

4.3.2 Results

According to Brooke (1996) the average score is 68 and the score rating is as follows: < 51 Awful, 51–68 Poor, 68 Okay, 68–80.3 Good, > 80.3 Excellent. The average score among the ten participants was 90 with a standard deviation of 5.7, therefore indicating an excellent rating, see also Figure 4.

By evaluating the recordings with the participants' behavior and expressions of the participants, several observations could be made. All participants immediately started reading the instructions stated on the website and were quick to grasp the structure and functionality of it. All of them were able to complete the steps for reconstructing a 3D model on their own without needing any type of hint or support. All participants looked at the expert settings, recognizing that they require more technical knowledge and assumed that those are presumably right the way they are. Nine participants quickly learned the use of the 3D viewer by experimenting with the controls without having to read the instructions for control. Most of the participants expressed that they found the given instructions useful, with P01 stating "Everything is comprehensible". Eight participants explicitly

expressed, that they found the text about the requirements of the video brief but still easy to understand, while some also mentioned the same about the overall texts of the website. Six participants liked the example images for comparing good and bad video quality and thought them useful. Furthermore, two participants stated that the small selection of preview images of the extracted frames provided more insight for them.

Three participants were confused about the quality settings. They assumed that it referred to the quality of their input video instead of the quality of the output 3D model. Two of them mentioned that the requirements of the video and the good and bad example of a video conditioned them to think the setting referred to the input video as well. Two participants were confused that the job status on the "My Uploads" page briefly read "Paused" and P6 mentioned, "When it says 'Paused', to me that means something might have gone wrong.". P6 also stated that "the job is cryptic and does probably not help in understanding". Another two participants had wished for a display of the remaining time for the video preprocessing and 3D reconstruction steps with P7 stating "It would be nice to see when it is going to be finished".

5 OPEN-SOURCE PHOTOGRAMMETRY IN THE CLOUD

The 3D reconstruction is a computationally expensive process. Modern 3D reconstruction pipelines also benefit from GPU acceleration, in case of AliceVision utilizing Nvidia CUDA. We relied on Amazon Web Services (AWS) for transplanting the AliceVision pipeline to the cloud, as it is one of the major cloud service providers.



FIGURE 6 | Selected frames of the video used for the cost evaluation.

TABLE 1 | Computation time and costs of the 3D reconstruction.

Video variation	Computation time (hh:mm:ss)	Costs on g4dn.xlarge (USD)
4K-51 frames	02:06:04	1.38
4K-26 frames	00:52:55	0.58
Full-HD-51 frames	00:37:27	0.41
Full-HD-26 frames	00:14:44	0.16

Using AWS, a number of compute instances can be launched to handle multiple pending 3D reconstructions in parallel. AWS is billed on a per usage basis. Hence, to optimize runtime costs, ideally, no compute instances should be running idle and instead be launched only when the need arises and be terminated whenever all tasks are finished. Therefore, we introduced a job management system, i.e., a logic that collects new jobs in a queue, handles the start and termination of compute instances, and distributes the queued jobs among these compute instances.

While, strictly speaking, only the actual 3D reconstruction process would need to be run on (accelerated) compute instances (with web services, databases and job management being able to be hosted elsewhere), we decided to utilize additional AWS services in order to derive a coherent infrastructure. One challenge in using cloud services is the vast amount of potential products and services that could be used (e.g., as of September 2021 AWS offers over 150 core services and over 15 thousand services and products *via* the AWS marketplace). Through comparisons with prior web-based reconstruction approaches and internal design workshops, we derived the AWS structure shown in **Figure 5**.

Elastic Compute Cloud (EC2) provisions scalable computing capacity in form of virtual servers based on different types of instances. Batch is a job management system which launches and terminates new EC2 instances and distributes the pending jobs that run *via* Docker accordingly. The jobs are executed *via* Docker images. Simple Storage Service (S3) offers an unlimited amount of online storage in form of buckets with primitive access *via* PUT and GET queries. Relational Database Service (RDS) provisions

SQL databases. Elastic Container Registry (ECR) offers a Docker image repository similar to Docker Hub. Virtual Private Cloud (VPC) creates a virtual network for all services to communicate among each other. For an extensive description of all Amazon Web Services, refer to the official [Amazon \(2021\)](#).

The starting point is the webserver based on an EC2 instance. When a user starts with step one and submits a new video, it will be stored in a S3 bucket (a cloud storage resource). The content of the upload form (title, video name, and quality settings) is saved *via* RDS in an SQL database. A new job for preprocessing is submitted to Batch which launches a new EC2 instance. The EC2 instance pulls its Docker image from ECR, retrieves the content of the upload form from RDS, and fetches the video from the S3 bucket. When the preprocessing is finished, the extracted and filtered frames of the video are stored in the S3 bucket. Once the user reaches step three and starts the 3D reconstruction, the same procedure with Batch repeats with a job for 3D reconstruction. All content on the S3 bucket can be publicly accessed *via* unique links.

5.1 Implementation

The proposed concept was implemented by mainly using the AWS Free Tier, which offers a selection of services for free under certain conditions. The webserver and web interface were hosted on the `t2.micro` general purpose instance which offers 1 vCPU core, 1 GB of memory and 30 GB of general purpose SSD (“gp2”). The back-end of the web interface was supported by MariaDB hosted *via* RDS on a separate `t2.micro` instance with 20 GB of free tier storage. The database could be manually set up on the same compute instance as the web interface but for the sake of simplicity and scalability the database is hosted *via* RDS. The S3 bucket was configured to be publicly accessible *via* URLs. Uploaded videos are stored in the root of the bucket with the naming pattern “video_<job-id>.<extension>”, extracted frames are stored in the path “/Extracted/*”, and results of 3D reconstructions are stored in “/Results/*”. The compute environment of Batch was configured to use accelerated compute instances of the instance family `g4dn`. All instances of this family provision at least one Nvidia GPU with

different dimensions of vCPU cores, memory, and storage space. Lastly, the Docker images, which are utilized by Batch, and contain the runtime environment for preprocessing and 3D reconstruction are stored on ECR. No special configurations were made to VPC as AWS automatically created the relevant configuration.

Both types of jobs, preprocessing and 3D reconstruction, are handled by Batch, and, therefore, a Docker image is required for each. AliceVision offers a KeyframeSelection node which can extract frames from videos and optionally filter them by the amount of blur and “spatial distance”. While AliceVision includes a frame extraction module, it was replaced by a custom solution due to its empirically determined unreliability. The preprocessing consists of a *Python* script that handles frame extraction *via* FFmpeg, blur detection *via* CPBD (Narvekar and Karam (2011)), and keyframe selection *via* feature matching with ORB (Rublee et al. (2011)). The Docker image was built on the “python:3.9” image and supplemented with the required dependencies of the *Python* script. The Docker image for 3D reconstruction contains AliceVision v2.3.0 and was built using the instructions in the official AliceVision GitHub repository. It is based on the image “nvidia/cuda:11.2.0-devel-ubuntu20.04” for Nvidia CUDA support. Additionally, a *Python* script was supplemented which handles the configuration and execution of the AliceVision nodes. The docker image is available under <https://hub.docker.com/repository/docker/mlrlabcoburg/alicevision>.

5.2 Cost Evaluation

The 3D reconstruction is outsourced to EC2 of the Amazon Web Services. Usage of EC2 is billed on a per use basis and the price depends on the type of instance and its hardware specifications. 3D reconstructions need to be computed on accelerated compute instances, i.e., instances with GPUs which fall in a higher price range. The computation time depends heavily on the hardware, resolution of the images, quality of the images, and quality settings of AliceVision. The number of images also have a great influence which again depend on the preprocessing and its parameters. The primary goal of this evaluation was to get an estimate on the likely costs of running a 3D reconstruction *via* AWS.

For the evaluation, a video was recorded showing a 360° walk-around of a statue of an eagle as seen in **Figure 6**. The preprocessing step was omitted in all tests to get a predictable number of extracted images, and to focus on the core 3D reconstruction steps. The video has a duration of ca. 24 s and was converted to four different variations: 1) 4 K with 2 frames per second (fps) resulting in 51 frames, 2) 4 K with 1 fps resulting in 26 frames, 3) Full-HD with 2 fps resulting in 51 frames, and 4) Full-HD with 1 fps resulting in 26 frames. A 3D reconstruction was started with all four variations on “high” quality settings. The computation was handled by a `g4dn.xlarge` compute instance with 4 vCPU cores, 16 GB of RAM, 125 GB of SSD storage, and 1 Nvidia Tesla T4 GPU. At the time of this evaluation the price for this instance type running Linux amounted to 0.658 USD per hour.

Table 1 indicates the runtime and associated costs. While the computation time (and costs) increases approximately linear in the number of frames, the increase in resolution leads to a substantially higher computation time.

6 DISCUSSION

Through our study of an existing user interface of the open-source 3D reconstruction software AliceVision, we identified potential challenges for non-expert users. Hence, we designed and developed a user interface aimed at non-professional users. The usability evaluation indicated that user interface was indeed useful and usable. Our evaluation also indicated, that even making available some reconstruction parameters, requires users to acquaint certain domain knowledge about the 3D reconstruction process, and, hence, might not be advisable for non-expert users. However, as with many other user interface software, there is always a trade-off between ease-of-use and the range of accessible functions in software user interfaces which have an effect on learnability and usability of the user interface (c.f. Grossman et al. (2009)). For example, the output quality (and success) of a 3D reconstruction depends on many factors such as the set of input images, the scene content and complexity as well as on the reconstruction parameters. Additional steps taken by the users would, for example, be required in case 1) the reconstruction fails or 2) the reconstruction quality is not in line with user expectations. While in both cases, a reconstruction could be restarted with a different set of parameters, for case 2) post-processing of the 3D reconstruction (e.g., removing unwanted artifacts) might also be useful. One option could be to integrate the proposed user interface as an interactive software wizard into the existing Meshroom interface, which would allow for these options. But, still, the migration of a non-expert user to an occasional, advanced, and professional user would need to be better understood through future research. This issue is further complicated through the rise of AI-based methods, which, without explainability (c.f. Abdul et al. (2018)) could lead to even higher user frustration. Finally, the input set of images or even the recorded scene (e.g., a translucent object such as a glass bottle) could be unsuitable for 3D reconstruction. While our interface supports users in selecting suitable images through a help screen, guidance during the capture process can complementary increase chances of successful reconstructions Hoppe et al. (2012); Langguth and Goesele (2013); Locher et al. (2016); Andersen et al. (2019); Isabelle and Laurendeau (2020).

Regarding, the cloud-based reconstruction pipeline, we decided for one amongst many possible design solutions. Specifically, relying solely on cloud software components from a single provider (in this case Amazon Web Services) has the benefit of a coherent and replicable configuration, but comes at an increased cost. The core components needed for computation are the EC2 instances. All other components could be implemented and hosted on third party providers for likely lower costs. Also, the choice of loading Docker images each time a new reconstruction job is started was done to avoid unnecessary running costs for (relatively expensive) GPU instances. However, this comes at the cost of an delayed start of the

reconstruction process, after the user has submitted a job. Empirically, it took about 5 minutes to start a new EC2 instance, load and initialize the Docker image from ECR. If a certain computational load (e.g., through a regular user base) is expected, one could, alternatively, always keep a minimum number of EC2 GPU instance running to allow for an immediate start of the reconstruction process, and start further instances as required.

Further, we reported the costs of reconstruction only for a single GPU compute instance. We experimented with further, more powerful instances, in order to potentially arrive at a better cost per job ratio. However, by default AliceVision utilizes all available cores for feature extraction, which can quickly lead to a buffer overflow (with one image processed by core; we experimented with up to 2 TB of RAM). The number of cores could be restricted to a fixed number to avoid this problem. However, this could lead to an unfair comparison between different hardware configurations (as only a fraction of the available cores could be utilized on a more powerful instance for parts of the pipeline) and a likely worse price to job ratio on those more powerful instances.

In future work, one could study the parallelization of the reconstruction pipeline further. As of now, the whole reconstruction process for a single job runs on a single instance. In contrast, one could derive a setup, where each step in the reconstruction process is executed on a separate instance. This way, one could potentially further optimize the cost structure as not all reconstruction steps require more expensive GPU instances.

Finally, while we employed an open-source software for making 3D reconstruction accessible in the cloud, the associated costs for executing the reconstruction are high (ca. 1.38 USD for a reconstruction of solely 51 frames in 4 K). For example, Autodesk ceased their free Photo-to-3D-Cloud service and charges per 50 photos to be reconstructed. If a cloud-based reconstruction service should be established on a broad scale, a funding model (e.g., based on donations) would need to be derived.

REFERENCES

- Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., and Kankanalli, M. (2018). "Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An Hci Research Agenda," in Proceedings of the 2018 CHI conference on human factors in computing systems (New York, NY: Association for Computing Machinery), 1–18.
- Amazon (2021). Aws Documentation. Available at <https://docs.aws.amazon.com/index.html>.
- Andersen, D., Villano, P., and Popescu, V. (2019). Ar Hmd Guidance for Controlled Hand-Held 3d Acquisition. *IEEE Trans. Vis. Comput. Graphics* 25, 3073–3082. doi:10.1109/tvcg.2019.2932172
- Bouck-Standen, D., Ohlei, A., Höfler, S., Daibert, V., Winkler, T., and Herczeg, M. (2018). Reconstruction and Web-Based Editing of 3d Objects from Photo and Video Footage for Ambient Learning Spaces. *Intl. J. Adv. Intell. Syst.* 11, 94–108.
- Brooke, J. (1996). Sus: A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*. Editors P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland. London: Taylor and Francis, 189–194.

7 CONCLUSION

In this work, we aimed at introducing a scalable and open-source 3D reconstruction pipeline that can be easily used by non-professionals, who do not have the technical knowledge required for photogrammetry software. To this end, we studied how to ease the use of AliceVision for non-experts, developed and evaluated a user interface aimed at non-professional users, and implemented and evaluated a cloud-based reconstruction pipeline. In future work, we aim at also supporting more advanced and professional users by making the advanced reconstruction parameters of AliceVision available through further interfaces, studying how to further accelerate the computation of individual 3D reconstructions and to optimize the cost structure on AWS.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

OD, conceptualized and implemented both the user interface and the reconstruction pipeline. He also conducted the usability study. JG aided in conceptualization of the user interface, the reconstruction pipeline and the studies. Both authors wrote parts of the manuscript.

- Fahim, G., Amin, K., and Zarif, S. (2021). Single-view 3d Reconstruction: a Survey of Deep Learning Methods. *Comput. Graphics* 94, 164–190. doi:10.1016/j.cag.2020.12.004
- Fleck, P., Schmalstieg, D., and Arth, C. (2016). "Visionary Collaborative Outdoor Reconstruction Using Slam and Sfm," in In Proceeding of the 2016 IEEE 9th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), Greenville, SC, USA, 20-20 March 2016 (IEEE), 1–2. doi:10.1109/searis.2016.7551588
- Gomes, L., Regina Pereira Bellon, O., and Silva, L. (2014). 3d Reconstruction Methods for Digital Preservation of Cultural Heritage: A Survey. *Pattern Recognition Lett.* 50, 3–14. doi:10.1016/j.patrec.2014.03.023
- Grossman, T., Fitzmaurice, G., and Attar, R. (2009). "A Survey of Software Learnability: Metrics, Methodologies and Guidelines," in Proceedings of the sigchi conference on human factors in computing systems (CHI '09), Boston, MA (New York, NY: Association for Computing Machinery), 649–658. doi:10.1145/1518701.15188
- Heller, J., Havlena, M., Jancosek, M., Torii, A., and Pajdla, T. (2015). "3d Reconstruction from Photographs by Cmp Sfm Web Service," in Proceeding of the 2015 14th IAPR International Conference on Machine Vision

- Applications (MVA), Tokyo, Japan, 18-22 May 2015 (IEEE), 30–34. doi:10.1109/MVA.2015.7153126
- Hoppe, C., Klopschitz, M., Rumpler, M., Wendel, A., Kluckner, S., Bischof, H., et al. (2012). Online Feedback for Structure-From-Motion Image Acquisition. *BMVC* 2, 6. doi:10.5244/c.26.70
- Iglhaut, J., Cabo, C., Puliti, S., Piermattei, L., O'Connor, J., and Rosette, J. (2019). Structure from Motion Photogrammetry in Forestry: A Review. *Curr. For. Rep* 5, 155–168. doi:10.1007/s40725-019-00094-3
- Isabelle, J., and Laurendeau, D. (2020). "A Mixed Reality Interface for Handheld 3d Scanners," in *International Conference on Human Interaction and Emerging Technologies* (Cham: Springer), 189–194. doi:10.1007/978-3-030-55307-4_29
- Jiang, S., Jiang, C., and Jiang, W. (2020). Efficient Structure from Motion for Large-Scale Uav Images: A Review and a Comparison of Sfm Tools. *ISPRS J. Photogrammetry Remote Sensing* 167, 230–251. doi:10.1016/j.isprsjprs.2020.04.016
- Julin, A., Jaalama, K., Virtanen, J.-P., Maksimainen, M., Kurkela, M., Hyyppä, J., et al. (2019). Automated Multi-Sensor 3d Reconstruction for the Web. *Ijgi* 8, 221. doi:10.3390/ijgi8050221
- Laksono, D. (2016). CloudSfM: 3D Reconstruction Using Structure from Motion Web Service. Ph.D. thesis. doi:10.13140/RG.2.2.34441.29289
- Langguth, F., and Goesele, M. (2013). "Guided Capturing of Multi-View Stereo Datasets," in *Eurographics (Short Papers)*. Editors M.-A. Otaduy and O. Sorkine: The Eurographics Association. doi:10.2312/conf/EG2013/short/093-096
- Locher, A., Perdoch, M., Riemenschneider, H., and Van Gool, L. (2016). "Mobile Phone and Cloud—A Dream Team for 3d Reconstruction," in *Proceeding of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, 7-10 March 2016 (IEEE), 1–8.
- Muratov, O., Slynko, Y., Chernov, V., Lyubimtseva, M., Shamsuarov, A., and Bucha, V. (2016). "3dcapture: 3d Reconstruction for a Smartphone," in *Proceeding of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Las Vegas, NV, USA, 26 June-1 July 2016 (IEEE), 893–900. doi:10.1109/cvprw.2016.116
- Narvekar, N. D., and Karam, L. J. (2011). A No-Reference Image Blur Metric Based on the Cumulative Probability of Blur Detection (Cpbd). *IEEE Trans. Image Process.* 20, 2678–2683. doi:10.1109/TIP.2011.2131660
- Ondruska, P., Kohli, P., and Izadi, S. (2015). Mobilefusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on mobile Phones. *IEEE Trans. Vis. Comput. Graphics* 21, 1251–1258. doi:10.1109/TVCG.2015.2459902
- Poiesi, F., Locher, A., Chippendale, P., Nocerino, E., Remondino, F., and Van Gool, L. (2017). "Cloud-based Collaborative 3d Reconstruction Using Smartphones," in *Proceedings of the 14th European Conference on Visual Media Production (CVMP 2017)*, London, United Kingdom, December 11–13, 2017 (New York, NY, USA: Association for Computing Machinery). doi:10.1145/3150165.3150166
- Rieman, J., Franzke, M., and Redmiles, D. (1995). "Usability Evaluation with the Cognitive Walkthrough," in *Proceeding of the Conference companion on Human factors in computing systems (CHI '95)*, Denver, CO (New York, NY: Association for Computing Machinery), 387–388. doi:10.1145/223355.223735
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "Orb: An Efficient Alternative to Sift or Surf," in *Proceeding of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6-13 Nov. 2011 (IEEE)*, 2564–2571. doi:10.1109/ICCV.2011.6126544
- Sheng, B., Zhao, F., Yin, X., Zhang, C., Wang, H., and Huang, P. (2018). A Lightweight Surface Reconstruction Method for Online 3d Scanning point Cloud Data Oriented toward 3d Printing. *Math. Probl. Eng.* 2018, 4673849. doi:10.1155/2018/4673849
- Stathopoulou, E.-K., Welponer, M., and Remondino, F. (2019). Open-source Image-Based 3d Reconstruction Pipelines: Review, Comparison and Evaluation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XLII-2/W17, 331–338. in *6th International Workshop LowCost 3D-Sensors, Algorithms, Applications*. doi:10.5194/isprs-archives-xlii-2-w17-331-2019
- Vergauwen, M., and Van Gool, L. (2006). Web-based 3d Reconstruction Service. *Machine Vis. Appl.* 17, 411–426. doi:10.1007/s00138-006-0027-1
- Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., and Firman, M. (2021). "The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA: IEEE Computer Society, 1164–1174. doi:10.1109/cvpr46437.2021.00122
- Zhao, C., Sun, Q., Zhang, C., Tang, Y., and Qian, F. (2020). *Monocular Depth Estimation Based on Deep Learning: An Overview*. Science China Technological Sciences, 1–16.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Dietz and Grubert. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

APPENDIX

The quality setting “medium” on the web interface uses the default configuration of AliceVision which is as seen in **Table A1**. The configuration of the quality settings “low” and

“high” are the same as “medium” with three main differences. The quality “low” sets the “describerPreset” to “medium” and the “downscale” of DepthMap as well as Texturing to 4. The quality “high” instead, sets the “describerPreset” to “ultra” and the “downscale” of DepthMap and Texturing to 1.

TABLE A1 | Mapping of the medium quality setting to AliceVision parameters.

Node	Parameter	Value	Node	Parameter	Value
CameraInit	groupCameraFallback	folder	DepthMapFilter	minViewAngle	2.0
	defaultFieldOfView	45.0		maxViewAngle	70.0
	allowSingleView	1		nNearestCams	10
FeatureExtraction	describerTypes	sift		minNumOfConsistentCams	3
	describerPreset	medium		minNumOfConsistentCamsWithLowSimilarity	4
	forceCpuExtraction	True		pixSizeBall	0
ImageMatching	weights		Meshing	pixSizeBallWithLowSimilarity	0
	minNbImages	200		computeNormalMaps	False
	maxDescriptors	500		estimateSpaceFromSfM	True
	nbMatches	50		estimateSpaceMinObservations	3
FeatureMatching	describerTypes	sift		estimateSpaceMinObservationsAngle	10
	photometricMatchingMethod	ANN_L2		maxInputPoints	50,000,000
	geometricEstimator	acransac		maxPoints	5,000,000
	geometricFilterType	fundamental_matrix		maxPointsPerVoxel	1,000,000
	distanceRatio	0.8		minStep	2
	maxIteration	2048		partitioning	singleBlock
	geometricError	0.0		repartition	multiResolution
	maxMatches	0		angleFactor	15.0
	savePutativeMatches	False		simFactor	15.0
	guidedMatching	False		pixSizeMarginInitCoef	2.0
	exportDebugFiles	False		pixSizeMarginFinalCoef	4.0
	StructureFromMotion	describerTypes		sift	
localizerEstimator		acransac	contributeMarginFactor	2.0	
localizerEstimatorMaxIterations		4,096	simGaussianSizeInit	10.0	
localizerEstimatorError		0.0	simGaussianSize	10.0	
lockScenePreviouslyReconstructed		False	minAngleThreshold	1.0	
useLocalBA		True	refineFuse	True	
localBAGraphDistance		1	addLandmarksToTheDensePointCloud	False	
maxNumberOfMatches		0	colorizeOutput	False	
minInputTrackLength		2	saveRawDensePointCloud	False	
minNumberOfObservationsForTriangulation		2	removeLargeTrianglesFactor	60.0	
minAngleForTriangulation		3.0	keepLargeMeshOnly	False	
minAngleForLandmark		2.0	iterations	5	
maxReprojectionError		4.0	lambda	1.0	
minAngleInitialPair		5.0	textureSide	8,192	
maxAngleInitialPair		40.0	downscale	2	
useOnlyMatchesFromInputFolder		False	outputTextureFileType	jpg	
useRigConstraint		True	unwrapMethod	Basic	
lockAllIntrinsics		False	useUDIM	True	
interFileExtension	.abc	fillHoles	False		
PrepareDenseScene	outputFileType	exr		padding	5
	saveMetadata	True		correctEV	False
	saveMatricesTxtFiles	False		useScore	True
	evCorrection	False		processColorspace	sRGB
DepthMap	downscale	2		multiBandDownscale	4
	minViewAngle	2.0		multiBandNbContrib	1, 5, 10, 0
	maxViewAngle	70.0		bestScoreThreshold	0.1
	sgmMaxTCams	10		angleHardThreshold	90.0
	sgmWSH	4		forceVisibleByAllVertices	False
	sgmGammaC	5.5		flipNormals	False
	sgmGammaP	8.0		visibilityRemappingMethod	PullPush
	refineMaxTCams	6			
	refineNSamplesHalf	150			
	refineNDepthsToRefine	31			
	refineNIters	100			
	refineWSH	3			
	refineSigma	15			
	refineGammaC	15.5			
	refineGammaP	8.0			
	refineUseTcOrRcPixSize	False			
	exportIntermediateResults	False			
	nbGPUs	0			