


Article

# To 3D or Not 3D: Choosing a Photogrammetry Workflow for Cultural Heritage Groups

Hafizur Rahaman \* and Erik Champion 

School of Media, Creative Arts, and Social Inquiry, Curtin University, Perth, WA 6845, Australia

\* Correspondence: hafizur.rahaman@curtin.edu.au; Tel.: +61-8-9266-3339

Received: 22 May 2019; Accepted: 28 June 2019; Published: 3 July 2019



**Abstract:** The 3D reconstruction of real-world heritage objects using either a laser scanner or 3D modelling software is typically expensive and requires a high level of expertise. Image-based 3D modelling software, on the other hand, offers a cheaper alternative, which can handle this task with relative ease. There also exists free and open source (FOSS) software, with the potential to deliver quality data for heritage documentation purposes. However, contemporary academic discourse seldom presents survey-based feature lists or a critical inspection of potential production pipelines, nor typically provides direction and guidance for non-experts who are interested in learning, developing and sharing 3D content on a restricted budget. To address the above issues, a set of FOSS were studied based on their offered features, workflow, 3D processing time and accuracy. Two datasets have been used to compare and evaluate the FOSS applications based on the point clouds they produced. The average deviation to ground truth data produced by a commercial software application (Metashape, formerly called PhotoScan) was used and measured with CloudCompare software. 3D reconstructions generated from FOSS produce promising results, with significant accuracy, and are easy to use. We believe this investigation will help non-expert users to understand the photogrammetry and select the most suitable software for producing image-based 3D models at low cost for visualisation and presentation purposes.

**Keywords:** digital heritage; 3D Reconstruction; FOSS; photogrammetry; point clouds; cultural heritage

## 1. Introduction

Developing 3D digital models of heritage assets, monuments, archaeological excavation sites, or natural landscapes is becoming commonplace in areas such as heritage documentation, virtual reconstruction, visualization, inspection of a crime scene, project planning, augmented and virtual reality, serious games, and scientific research. Conventional geometry-based modelling approaches using software like Maya, Blender or 3D Studio Max etcetera, typically involves a steep learning curve and requires a considerable amount of time and effort. Advancements in hardware (laser scanners, UAV, etc.), especially for 3D reconstruction of real-world objects, have made it easier for professionals to virtually reconstruct 3D scenes. However, tools like laser scanners and structured lighting systems are often costly. Additionally, this technology has limitations in regards to rendered material properties and environmental conditions, such as strong sunlight [1].

There is a critical need for tools, which allow non-expert users to comfortably and efficiently create 3D reconstruction models, especially for visual and documentation purposes. To answer this demand, some commercial, as well as free and open source software (FOSS) based on image-based modelling (IBM) or photogrammetry, have emerged. Image-based 3D reconstruction software creates 3D point cloud with camera poses derived from uncalibrated photographs. The software determines the geometric properties of objects from photographic images. This process requires comparing reference points or matching pixels across a series of photographs. The quality and specific number of

photographs are needed to allow the surface to process, match and triangulates visual features and further generating 3D point-cloud.

Structure from Motion (SfM) is one of the most common techniques in image-based modelling approach implemented in different software. This technology enables non-experts to quickly and easily capture high-quality models through uncalibrated images captured from cheap setups, without requiring any specialized hardware or carefully designed illumination conditions. Generally the workflow follows six steps to produce 3D reconstructions/3D models:

- (1) Image acquisition (or adding photos)
- (2) Feature detection, matching, triangulation (or align photos)
- (3) Sparse reconstruction, bundle adjustment (or point cloud generation)
- (4) Dense correspondence matching (or dense cloud generation)
- (5) Mesh/surface generation, and
- (6) Texture generation.

A few software packages also offer cloud/mesh editing within a single package [2]. The whole process of 3D reconstruction can be done with the support of cloud-based computation or a local PC; based of the service/application used. This paper does not cover the cloud processing method and only focuses on those software/applications that run on a local PC workstation.

Ample studies have been published so far on image-based modelling software analyzing their performance [3–5], accuracy in 3D production [6–8], used algorithms [9], and scalability [1,9,10]. A few of them also discuss workflows [11,12]. However, it is rare to find studies on either best practice or optimized workflows, which can be adopted by the general public to create 3D reconstruction models easily and free or at little cost.

This paper presents a comparison study of four 3D reconstruction FOSS (free and open source software) packages selected on the basis of their price, platform independence, scalability, output format, accuracy, ease of use and installation, and most importantly, their required processing time (suitable for home consumer viewpoints). Popular commercial, and free and open source software (FOSS) was studied first. Agisoft's Metashape (version 1.3.3) commercial package and four applications, i.e., Visual SFM (version 0.5.26), Python Photogrammetry Toolbox with the graphical user interface (PPT GUI) (version 0.1), COLMAP (version 3.1) and Regard3D (version 0.9.2) were selected to represent the FOSS group. This article will later describe and illustrate their workflows for a clearer understanding of their working methods, including differences at each step of their production pipelines.

Two datasets were used to evaluate the selected FOSS group; and compared with the reconstruction achieved from a commercial software (as ground truth data), by using CloudCompare software to assess the average deviation of their produced point clouds. This article concludes with a discussion and assessment of their limitations and strengths, especially in regards to their ease of use, workflow, performance, and any required learning curves (from a non-expert end-user's perspective). Please note, this study mostly used the default settings offered by the selected FOSS to produce the reconstruction; results may vary for customised settings.

## 2. Related Works

Recently, image-based modelling has become a growing area of interest in academic circles; a considerable amount of literature has also become available on relevant topics. 3D point clouds can only be produced when the relations between photographs are appropriately established. Several articles describe the overall production of 3D models as a sequence of calibrated or uncalibrated photographs [13–15], including details of different techniques for achieving a high degree of accuracy [1,6].

Researchers have successfully worked on developing various methods of producing 3D models from photographs. For example, Debevec et al. [16] developed the 'Façade system' and captured a 3D model of MIT campus through the 'MIT City Scanning Project'. From a sequence of street view images, Xiao et al. [17] developed a semi-automatic image-based approach to reconstruct 3D façade

models. Brown et al. [18] later presented another technique based on image-based modelling, which used ‘recover camera’ parameters to develop 3D scene geometry. Snavely et al. [19] introduced Photo Tourism (later named as Photo Synth by Microsoft) based on the work of Brown, with more scalability and robust engine. Agarwala [20] proposed another technique based on panoramas of roughly planar scenes to produce 3D. Some researchers have worked on evaluating photogrammetry software [11,14], however their study mostly concentrated on modelling accuracy and performance.

It is rare to find articles that dealt with comparing workflows on determining best practice, exploring effective learning methods and uses intended for the general public who have limited technical knowledge. This significant issue has also been previously noted by Remondino et al. [21], Deseilligny et al. [7], and De Reu et al. [22]. However, Ben Nausner [23] worked on the project ‘Virtual Nako’ and discussed the workflow used for 3D acquisition and visualisation of the scene that can be finally embedded with Google Earth for public viewing. Sauerbier [24] described the workflow for photogrammetric processing of areal images for generation of a textured 3D model of Tucume, Peru. Koutsoudis et al. [12] presented a ‘versatile’ workflow for making 3D reconstruction models based on a set of open source software. He used eight different software applications to integrate the workflow and claimed this approach could be useful for those who are on a level similar to a “computer graphics enthusiast experienced with 3D graphics”. Similarly, Deseilligny et al. [7] described another workflow for an automated image-based 3D modelling pipeline for the accurate and detailed digitization of heritage artefacts based on open-source software.

3D documentation of cultural heritage poses prime importance regarding historic preservation, tourism, educational and spiritual values [25,26]. Image-based reconstruction software also claims to be cost-effective as compared to traditional laser scanning methods and can provide an automated system with considerable accuracy in the 3D model generation [15,27]. However, costs are incurred for acquiring commercial software licenses, and a level of technical skill and knowledge is necessary. Boochs et al. [28] and Kersten and Lindstaedt [29] demonstrated free or low-cost 3D reconstruction methods for archaeological and heritage objects but the presented methods were targeted at technical users.

Structure from Motion (SfM) based 3D reconstruction software has become widely used in recent years [30]. Articles those covers benchmarking and compare SfM based reconstruction, however, merely covers FOSS solutions. While few studies cover VisualSfM [8,11,21,31] and Bundler/PMVS [11,13,21], however these studies are rarely tailored for novice users or non-expert users in supporting their involvement in making 3D reconstruction.

Considering the interest of local communities in showcasing their heritage assets, the issue of the ownership, and the significant amount of time and money to train, appoint and retain 3D reconstruction specialists; finding a suitable FOSS with an optimized workflow would be of benefit to these communities. This paper aims to examine various FOSS-based applications and their related workflow and presents a comprehensive comparison of their modelling accuracy to help general users with limited relevant technical knowledge, and who are interested in heritage documentation and visualization.

### 3. Selection of the Software

A wide variety of 3D modelling programs are available based on SfM; ranging from simple home-brew systems to high-end professional packages [30]. Keeping the target groups in mind who have limited budget and entry level of skills and requires a robust, easy to learn, scalable model building environment, the following selection criteria are used in selecting FOSS package

- Free or low cost.
- Simple workflow; easy to install, learn and use.
- Supports Graphic User Interface (GUI).
- Supports close-range photogrammetry and Structure from Motion (SfM).

This selection process, however, has excluded free commercial products with limited functions/capabilities, and cloud-based online services. A list of image-based 3D reconstruction software has been found from Wikipedia at the time of writing ([https://en.wikipedia.org/wiki/Comparison\\_of\\_photogrammetry\\_software](https://en.wikipedia.org/wiki/Comparison_of_photogrammetry_software), visited, 30.09.2018). This site has described the presence of ninety-eight (98) existing software, including standalone programs and plugins that can build 3D models from photographs. While we are wary of Wikipedia articles being used as reference literature, in this case, we know of no comparable study that lists such a large number of image-based modelling software (for older reviewers please see [http://www.pvts.net/pdfs/ortho/photosftw\\_purch.pdf](http://www.pvts.net/pdfs/ortho/photosftw_purch.pdf), visited 02.10.2018).

During the selection process, some commercial software such as 3Dsom, Autodesk ReMake, PhotoModeler, Metashape, Aspect3D, 3DF Zephyr, and RealityCapture were reviewed. As budget is one of the primary concerns, Agisoft's Metashape has been selected due to its low price (education standard version) and accuracy in production of 3D point cloud [14,32]. On the other hand, based on the selection criteria four popular Free and Open Source (FOSS) software, i.e. VisualSfM, Python Photogrammetry Toolbox (PPT GUI), COLMAP, and Regard3D were selected as they also support SfM (Structure from Motion) for 3D reconstructions. Table 1 presents the various basic modelling features, which are offered by the selected software.

We note here that the study is not a comprehensive comparison of output 3D environments. Mostly the default settings have been used while excluding some other crucial aspects. For example, features such as the capacity to handle large datasets, GPU support, multiple image frames captured at different time moments with alternative settings were not considered in this study.

**Table 1.** Operationalizing the considerations.

| Software & Developer                     | License Type              | Supported OS         | Basic Features           |                                |  | Supported Modelling Features |             |      |         |         |
|--|---------------------------|----------------------|--------------------------|--------------------------------|--|------------------------------|-------------|------|---------|---------|
|  |                           |                      | Type of Photography      | Price                          | Export Format  | Point Cloud                  | Dense Cloud | Mesh | Texture | Editing |
| <b>Metashape</b>                         | Commercial                | Linux, Windows, OS X | Aerial, Close-range, UAS | 3499~179 \$Edu. Ver.: 549~59\$ | .obj, .3ds, .dae, .ply, .stl, .dxf, .fbx, .wrl, .pdf, .u3d | Yes                          | Yes         | Yes  | Yes     | Yes     |
| <b>COLMAP</b>                            | GNU GPL v3                | Linux, Windows, OS X | Aerial, Close-range, UAS | Free                           | Ply, .nvm, .out, VRML,                                     | Yes                          | Yes         | Yes  | No      | No      |
| <b>Python Photogrammetry-try Toolbox</b> | Open-source               | Linux, Windows, OS X | Close-range              | Free                           | .ply, .out   | Yes                          | Yes         | No   | No      | No      |
| <b>VisualSfM</b>                         | Open-source (partly)      | Linux, Windows, OS X | Close-range              | Free                           | .ply, .out   | Yes                          | Yes         | No   | No      | No      |
| <b>Regard3D</b>                          | Open-source (MIT license) | Linux, Windows, OS X | Close-range              | Free                           | .ply, obj, .pcd,   | Yes                          | Yes         | Yes  | Yes     | No      |

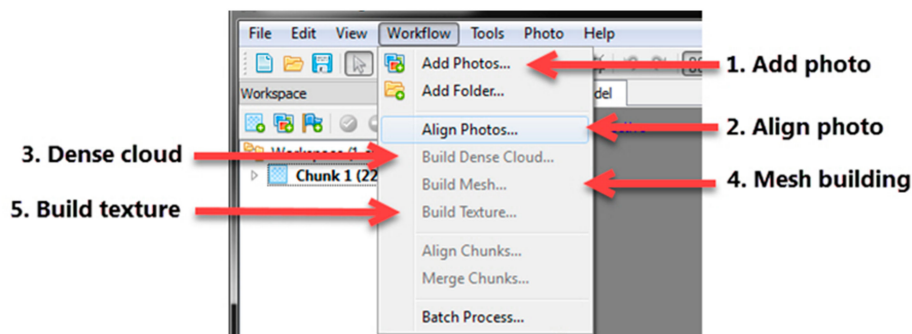
## 4. Workflow Study

For the convenience of the reader, a basic review of the workflow or production pipelines of the selected application is described first. A typical critical attribute is presented later in Section 6.

### 4.1. Metashape/PhotoScan

Agisoft Metashape is a low-cost commercial 3D reconstruction software from Agisoft LLC, Russia. Metashape automatically builds precise textured 3D models by using digital photos (both metric and non-metric) of an object or scene and is available in Standard and Pro versions. This program works on Windows, Mac OS and Linux operating systems on a local PC, and therefore all data remains with the user. In some cases, it is difficult or even impossible to generate a 3D model of the whole object in a single attempt. To overcome this difficulty Metashape offers options for splitting the set of photos into several separate "chunks" within the project. This way, the default processing/steps can be performed on each chunk separately, and then the resulting 3D models can be combined.

To use Metashape, users must capture the images or photographs all around the object (to get all possible views, mostly in a circular fashion). Either masked or unmasked photographs can be added to the workflow (step 1), and image alignment is required before computing (step 2) (Figure 1). However, Metashape recommends masking all irrelevant elements on the source photos (such as the background and any accidental foreground) for better reconstruction results. Step 3, Metashape computes the photographs and builds the geometry (create a point cloud) of the scene. Users can edit and clean the unnecessary point cloud by cropping or removing extra floating objects before creating the mesh. The density of the point cloud (generated from step 3) can vary from normal, medium to ultra-high format, and later, the mesh (step 4) can be calculated by following any setting (from fine to high pre-set). Unwanted geometry/faces can be edited at this stage before building the texture (step 5). A user can use additional features (optional) to close holes (step 6) and export the model.



**Figure 1.** A sample of typical workflow offered by Metashape (PhotoScan) GUI.

The resulting mesh can be textured with minimum user effort by leaving the default setting (which can be generic, average, fill holes, 2048 × 2048, and standard). 3D models can be exported in various formats (OBJ, 3DS, VRML, COLLADA, PLY, FBX, DXF, and PDF) for further editing and rendering.

### 4.2. Visual SfM

VisualSfM is an academic open-source software solution that supports Linux, Windows and Mac OS, and developed by Changchang Wu after he had combined several of his previous projects (more information can be found on Dr Wu's website <http://ccwu.me/vsfm>, visited 02.10.2017). VisualSfM does not require the input of any camera information; instead, it provides a GUI, which includes SiftGPU and PMVS. SiftGPU finds the camera positions while PMVS creates a point cloud from the matched photos. VisualSfM does not create a complete reconstruction, but it basically provides a point cloud that requires post-processing.

The workflow starts with adding image file (step 1); either by loading a .txt file containing relative image paths (NView Match) or by directly importing of multiple images. VisualSfM can automatically determine all the used parameters of the camera to acquire the photos. The next step (step 2), VisualSfM detect features in each image and find matches. It provides a variety of different algorithms for feature detection including Scale Invariant Feature Transform (SIFT) and SiftGPU (a GPU implementation of SIFT) [11]. Matches found in the previous step are later converted to points in 3D space (step 3) achieved through Bundle adjustment.

This step can be done by going to the SfM menu and selecting Reconstruct Sparse or by using the Compute 3D Reconstruction shortcut (Figure 2). A denser point can be achieved by using the PMVS/CMVS tool (step 4). Select Reconstruct Dense or click on the Run Dense Reconstruction shortcut. This command prompts to save the output as a \*.nvm file, which VisualSfM creates and saved in a folder titled \*.nvm.cmvs and runs the CMVS (Clustering view for Multi-view Stereo). A \*.ply file is also automatically saved in the same location.

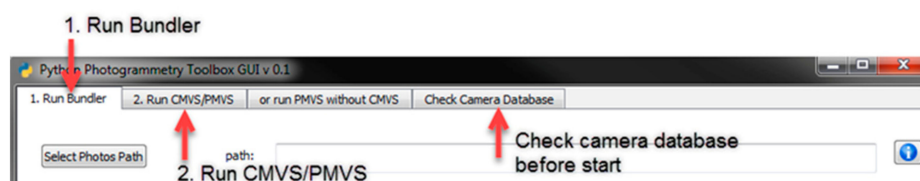


**Figure 2.** A sample of typical workflow offered by VisualSfM GUI (adopted from <http://ccwu.me/vsfm/doc.html>).

#### 4.3. Python Photogrammetry Toolbox (PPT GUI)

The Python Photogrammetry Toolbox (PPT) is free and open-source software that runs on various platforms (Mac OS, Linux, and Windows). The software was initially developed by Pierre Moulon and Alessandro Bezzi. The toolbox is composed of python scripts that automate the 3D reconstruction process from a set of pictures. The reconstruction process is mainly performed in two parts: camera pose estimation/calibration and dense point cloud computation. Open-source software such as Bundler for the calibration; and CMVS/PMVS for the dense reconstruction is employed to perform these intensive computational tasks. PPT GUI provides a 2-step reconstruction workflow. However, before starting step 1, we recommend checking the camera database. The terminal window will prompt for the camera model and sensor (CCD) width size if it is not in the PPT's database.

Step 1: Run Bundler performs the camera calibration and computes the 3D camera pose from the set of images. Despite automation, the user can control the result by choosing from two initial parameters: the image size and the feature detector. Step 2: Run CMVS/PMVS or run PMVS without CMVS, it takes the output of the previous step as input, and perform the dense 3D point cloud computation. However, running CMVS before PMVS is highly recommended, but not strictly necessary. It is also possible to use PMVS directly (Figure 3) [33]. The software generates the outputs (\*.ply file) automatically to a 'temp' directory, which prompts through the terminal window.



**Figure 3.** A sample of typical workflow of PPT GUI.

#### 4.4. COLMAP

COLMAP is a general-purpose Structure from Motion (SfM) and Multi-View Stereo (MVS) pipeline which supports both graphical and command-line interface. The software was developed by Johannes L. Schoenberger and is licensed under the GNU General Public License v3 or later (source: <https://colmap.github.io/license.html>, visited: 03Oct.2018). COLMAP offers a single click Automatic Reconstruction with an inbuilt default setting. This automatic process is faster, when compared to the step-by-step process, however, it has a trade-off in terms of the reconstruction quality. On the other hand, a manual step-by-step process may provide more flexibility on settings and accuracy in dense reconstruction.

The user needs to run the \*.bat file to open the program followed by the file menu to open/create a new project (Figure 4). The user must show the program the present location of the images and where to locate the database (step 1). The user needs to start with the feature extraction (step 2) under the Processing tab; followed by the 'feature matching' (step 3). Then the user needs to reconstruct the camera positions, and produce a sparse point cloud. The start reconstruction command is located under the pull-down menu Reconstruction (step 4). COLMAP produces the 3D view while depicting cameras being added to the scene while it simultaneously forms the sparse point cloud. After completing this stage, the sparse cloud can be exported. A Bundle adjustment can be run before densification. Dense reconstruction (step 4) contains three steps, i.e. un-distortion, stereo, and fusion. The final step (step 5) is Meshing. All models can be exported as \*.nvm, \*.out, \*.ply, and \*.wrl file format.

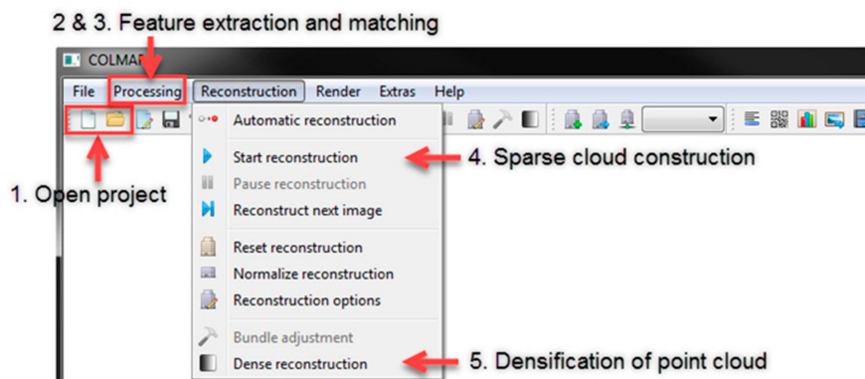


Figure 4. A sample of typical workflow offered by COLMAP GUI.

#### 4.5. Regard3D

Regard3D is another free and open source structure from motion program that supports multiple platforms (Windows, OS X, and Linux). Regard3D has a simple and straightforward GUI. The details of the executed tasks are highlighted in the left tree view (Figure 5).

Experimenting with settings thereby is more accessible, since the user only has to click on a completed task to see a list of the arguments used to generate it, as well as view the running time of that selected step. Similar to other software applications, the user needs to set a project path and a name to start a project. Photographs are required to be set (step 1) for the software to compute the matches (step 2). Next up is camera registration. In other words, the process of determining each camera's position and orientation in the scene (step 3), can be done by selecting the match results and clicking 'Triangulation'.

Based on this simple sparse point cloud (which consists of points that are unevenly distributed over the scene and generated from the corresponding camera positions), users can "densify" the triangulation result (step 4). From the tree view, it is possible to highlight the results of step 4 and choose 'Create dense point cloud'. The dense cloud (\*.ply, \*.pcd) can be exported at the end of this step. Users can also generate a mesh by clicking Create Surface (step 5). Please note that if users select



CMVS/PMVS Poisson reconstruction is the only option offered. On the other hand, two colorization methods are offered: coloured vertices or texture. The user can now export the generated surface as a \*.obj file or directly export to MeshLab as \*.mtl file format (step 6).

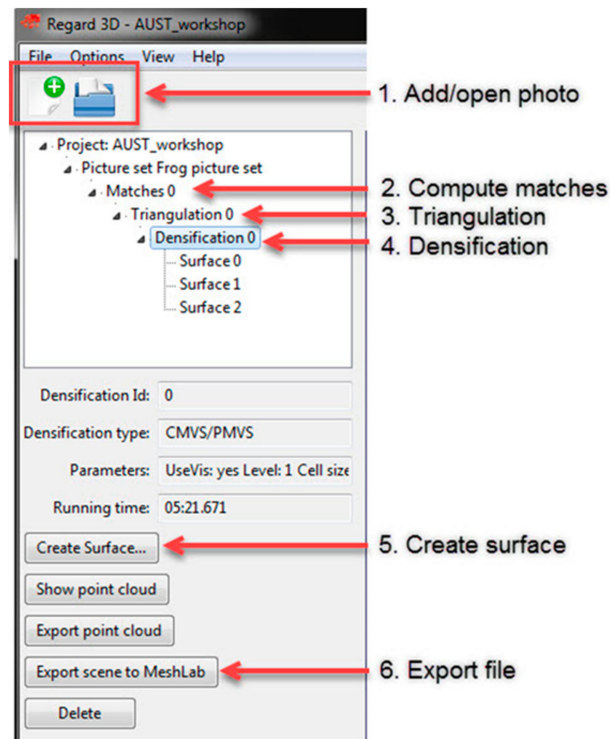


Figure 5. A sample of a typical workflow in Regard3D GUI.

## 5. Performance Study

This section presents a comparative study of the four pre-selected FOSS based 3D reconstruction applications on the basis of their produced point cloud, computation time, and reconstruction accuracy.

### 5.1. Dataset and Computation

The primary goal of this article is to evaluate and compare the efficiency, accuracy and constraints of the selected software to provide insight and help in choosing the most appropriate software, in general. To conduct this evaluation, a repository of seven objects (or data sets) was used for a pilot study. Two objects were selected for the final comparison based on image acquisition response time and a satisfactorily dense point cloud (with minimum holes). Two data sets, i.e., 22 photographs of a sculpture (frog) and 50 photographs of a historic building elevation (Kidogo Arthouse, Perth, Western Australia) were used to run the full test. The photographs were captured with a resolution of  $3456 \times 2304$  pixels, in an outdoor setting by a Canon EOS 600D camera (sensor size  $22.3 \times 14.9$  mm, sensor type CMOS, lens 10.0–20.0 mm). The reconstructions were computed on a standalone PC with Intel i7-6700 CPU, 3.4GHz system with 16GB RAM, and Quadro K620 graphic card with 2GB VRAM. The operating system was Windows 7 Enterprise.




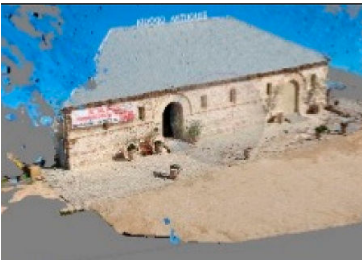
### 5.2. Comparing Methods

To determine the accuracy of the 3D point clouds derived from the different software the two different data sets (sculpture and building façade) were used for comparison. The comparison was made between the point cloud produced by the FOSS and a reference mesh surface produced from the commercial software (i.e., Metashape formerly known as PhotoScan).

The idea presented by Schönig and Heidemann [14] was used for the comparison study. Since a ground truth is required for benchmarking the FOSS applications, and as an alternative to LIDAR data and related technology, the point cloud generated from Metashape was used as ground truth (Table 2). This also means that the errors in Metashape reconstruction were not taken into consideration. This accuracy should suffice for general purposes of modelling such as documentation and visualization (but not for scientific analysis). The Metashape reconstruction will be referred to as ‘ground truth’ objects, while the outputs from each FOSS will be referred as ‘reconstructed’ objects. The comparisons were made with free open source software CloudCompare, and the calculated results are summarized in Tables 3 and 4.

The different point clouds were co-registered manually by CloudCompare. Additionally, we note that the image-based models were not scaled, meaning that the model does not reflect the real world structure’s size. The reconstruction models were scaled in relation to the ground truth and registered by using an ‘iterative closest point algorithm’ (ICP) [34] with a target error difference (RMS) of  $1.0 \times 10^{-20}$ , and a random sampling unit 60,000 was applied. Once the models were registered, the minimal distance between every point to any triangular face of the meshed model (i.e. ground truth) was computed. Using the normals of the meshes, the distance was calculated as indicated. These distances were visualized using a ‘pseudo schematic colour heat map’ (or heat maps), where the range is based on a blue-green-red scheme. The generated heat maps of the given datasets are presented in Tables 2 and 3. From the distances, the mean and standard deviation of the distance distribution for the whole object was also calculated. A Chi-square distribution was assumed for the modelling of the distance distribution between the ground truth and the reconstruction. The computation time of each solution is measured and also presented in Tables 2 and 3.

**Table 2.** Point cloud data generated from Metashape (used as ground truth).

| Original Object   | Generated Point-Cloud   | Process Time  |
|---|---|---|
|  |  | Align photo: 68.10 s<br>Dense cloud: 247.77 s<br>Build Mesh (high): 142.44 s<br>Texture: 54.48 s<br>Total time: 513 s (8.55 min.)                   |
|  |  | Match/Align photo: 316.62 s<br>Dense cloud: 4406.43 s<br>Build Mesh (high): 596.28 s<br>Texture: 103.2 s<br>Total time: 5390.4 s<br>(89 min 50.4 s) |





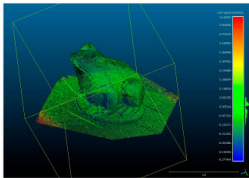
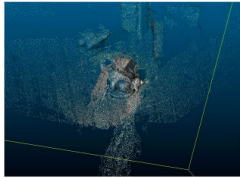
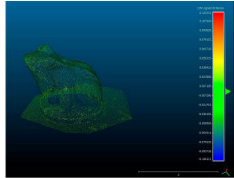
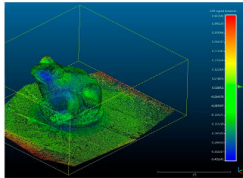
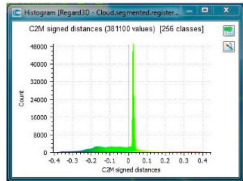
### 5.3. Result

For the first dataset (frog sculpture, 22 photographs are used), three out of four FOSS managed to compute the data, and later these three results were used to compare with the ground-truth (Table 2). This figure shows that all tested software tools yield useful results with this data set. PPT GUI produced a considerable amount of point cloud data, with three partial sets, which however prohibited comparing its accuracy with the ground truth data. The heat map and the Chi-square/Gaussian histogram distributions in Table 3, shows that the closest match was produced by COLMAP, followed





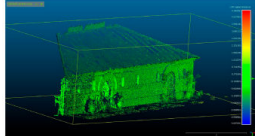
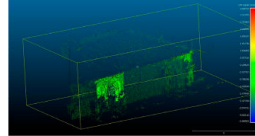
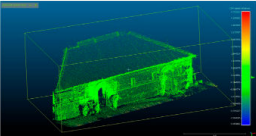
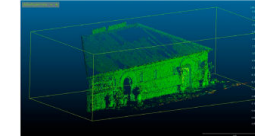
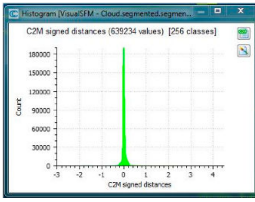
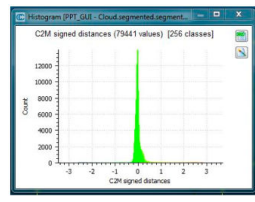
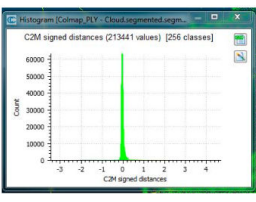
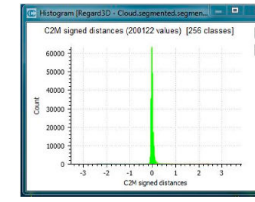
by Regard3D and VisualSfM. While VisualSfM produced a larger deviation distance from the ground truth, it took considerably less computation time. COLMAP generated the closest match with the ground truth; however, it took the longest time and did not produce any textures. Overall, Regard3D seems to be, overall, the most well-rounded performer here.

50 photographs were used for the second data set, but the acquired mesh developed holes in the reconstruction (Table 4). Except for PPT GUI, which produced the noisiest clouds, the other three software produced consistent results and had similar distance distributions with reference to the ground truth. The heat map also indicates the minor deviation between the produced results. COLMAP and Regard3D managed to capture most of the details without much noise as found on the histogram. On the other hand, VisualSfM failed to produce the roof of the model. Similar to the previous experiment, COLMAP took three times longer for computation in comparison to Regard3D, and did not produce any texture.

Table 3. FOSS benchmarking with dataset 01 (sculpture).

|                     | Visual SfM  | PPT GUI  | COLMAP   | Regard3D   |
|---------------------|---|--|--|--|
| Pointcloud          |   |                            |   |    |
| CloudCompare result |    |                           |    |   |
|                     |   | Failed to compare because of noisy cloud   |  |   |
| Processing time     | Loading image data: 5 s<br>Image pairwise matching: 12 s.<br>3D Reconstruction: 13 s.<br>Dense reconstruction: 274.77 s.<br>Total time: 304.8 s (5.08 min., no texture) | Matching: 92.30 s<br>Processing time: Not shown by the tool.<br>Total time: Couldn't calculate. (no texture) | Feature extraction: 7.38 s<br>Feature matching: 60.24 s<br>Sparse reconstruction: 39.78 s<br>Dense reconstruction (Undistortion + Stereo + Fusion + Meshing): 1489.8 s<br>Total time: 1597.2 s (26 m 37.2 s, no texture) | Image matching: 132.65s<br>Triangulation: 34.89 s<br>Denisification: CMVS/PMVS— 267.98 s<br>Surface generation: 62.73 s<br>Total time: 438.25 s (8 m 30 s, with texture) |

**Table 4.** FOSS benchmarking with dataset 02 (Kidogo Arthouse).

|                     | Visual SfM  | PPT GUI   | COLMAP  | Regard3D   |
|---------------------|---|---|---|--|
| Pointcloud          |    |                                |   |   |
| CloudCompare result |    |                                |   |   |
|                     |    |                                |   |   |
| Processing time     | Loading image data: 12 s<br>Image pairwise matching: 58 s.<br>3D Reconstruction: 26 s.<br>Dense reconstruction: 474 s.<br>Total time: 570 s<br>(9 m 30 s, <i>no texture</i> ) | Matching: 220.8 s<br>Processing time: PMVS2: 175.8 s<br>Total time: 396.6 s<br>(6 m 36.6 s, <i>no texture</i> ) | Feature extract: 16.14 s<br>Feature matching: 330.9 s<br>Sparse reconstruction: 127.62 s<br>Dense reconstruction (Undistortion + Stereo + Fusion + Meshing): 5542.08 s<br>Total time: 6016.74 s<br>(100 m 16.8 s, <i>no texture</i> ) | Image matching: 1784 s<br>Triangulation: 275.74 s<br>Denification: CMVS/PMVS— 149.06 s<br>Surface generation: 88.95 s<br>Total time: 2297.75 s<br>(38 m 17.75 s, <i>with texture</i> ) |

## 6. Discussion

Photogrammetric 3D modelling for heritage documentation is a well-studied topic. This paper has studied the workflows and compared the results of the 3D reconstructions achieved from the four FOSS-based applications. This section briefs the understanding gained from the study.

### 6.1. Workflow

Metashape is one of the most popular commercial packages; hence we used it to produce the ground truth data. It has a robust but straightforward pipeline that can produce accurate results [14,32]. Metashape has the capability to split the set of photos, manually remove extra point clouds and also has some nice extra features such as the ability to automatically close holes and directly export to Sketchfab.

In general, VisualSfM is an excellent tool for taking something static, converting and then importing point clouds into a 3D environment, but it requires additional processing and human skill to make a perfect digital environment. Additional tools (PMVS/CMVS) are also required to run VisualSfM, they need to be downloaded from their respective sites and copied to the local folders. This might be tricky for new and non-technical users; however, the rest of the installation process is mostly automatic. If some cameras fail to align correctly, users are required to start again and go back and shoot (or collect) more photographs and recommence the computation process.

It is also possible to manipulate the initial reconstruction (i.e. the sparse reconstruction) while it is in progress, removing those bad points/cameras that have been added to the wrong position/orientation. The software atomically generates the sparse cloud and dense cloud to the user-designated folder including the SIFT and matching process (\*.sift and \*.mat file). Thus, the user does not need to re-sift or

re-match images that have already been analyzed. However, adding new images to improve previously completed reconstruction takes as much time, for it requires analysing a new data set. VisualSfM does not offer any inbuilt editing or noise cleaning option; therefore external software is required.

Python Photogrammetry Toolbox (PPT GUI) presents a relatively simple user interface but requires some time and effort to learn. It is a little confusing that the GUI suggests to start with the numeric sequence, i.e., '1. Run Bundler' as the first step. However, a user must insert the photo path and camera data (if it is not in the inbuilt database) before running the bundler, which is supposed to be the first step. The bundle adjustment automatically saves itself in the temporary directory (by default). Thus, it is a good idea to copy the OSM-\* directory to somewhere safe because it will be lost next time the computer boots. The main drawback is that the GUI does not provide any visual cues of the generated point cloud or mesh, so users must need to use an external viewer or editor such as MeshLab to view or edit the output.





COLMAP offers a simple one-step reconstruction, which may be convenient as a quick solution for novice users. The manual or step-by-step process, on the other hand, offers better reconstruction quality with many adjustment options. However, the 'stereo' step of the dense reconstruction process may crash the program if the video card memory is inadequate. Reduction of the memory use during a dense reconstruction process is therefore recommended (source: <https://colmap.github.io/faq.html#faq-dense-memory>, access date 23 Dec.2018). Downsizing the pixels of photographs to between 750 and 2000 may also solve this issue.

Additionally, the 'stereo' step takes quite a long time during computation. Despite this, COLMAP offers a plethora of variables to tweak, and it may be possible to minimise the computation time and achieve reasonable results. Aside from resolving the memory issue by lowering 'max\_image\_size', the process is mostly straightforward. COLMAP can produce point cloud movie animation as well. The only drawback to mention here is that the meshing process does not produce a texture; instead, it applies colours to each vertex. This output may appear little muddy, but a user can export cameras as bundle or as \*.nvm files, and import them to MeshLab and generate a texture.

Regard3D offers a simple one-step installation process, and the installer file can be downloaded from its official website. Regard3D is completely free to use with a simple workflow that offers a vertical menu to flow various steps. Additional options are offered on the right side of the GUI. Although it does not offer mesh or point cloud editing, it does offer surface reconstruction and texturing. The only drawback is that Regard3D uses a modified version of openMVG, which means that there is frequently a delay in getting the most recent version. However, this may not be an issue for non-technical users and beginners.

In general, based on which the computation method is chosen, image-based modelling software typically follows a six-step approach to produce 3D reconstructions or 3D models (Table 5). VisualSfM and PPT GUI produce point cloud automatically in a default location, whereas COLMAP, Regard3D, and Metashape ask for user input to export the cloud. Regard3D can produce mesh (using Poisson/FSSR) and can colour vertices or textures. COLMAP instead applies colours to each vertex and cannot produce textures.

**Table 5.** A typical workflow of the selected FOSS for 3D reconstruction.

| Typical Steps  |  |  |  |  | PPT GUI                       |
|--|---|---|--|---|-------------------------------|
|  | Metashape   | Regard3D  | COLMAP   | VisualSfM   | Python Photogrammetry Toolbox |
| 1 <b>Add Photos</b><br>(Image Acquisition)                                     | ✓   | ✓   | ✓  | ✓   | ✓                             |
| 2 <b>Align Photos</b><br>(Feature detection / matching / triangulation)        | ✓   | ✓   | ✓  | ✓   | ✓                             |
| 3 <b>Point Cloud generation</b><br>(Sparse reconstruction / bundle adjustment) | ✓<br>M  | ✓   | ✓  | ✓   | ✓                             |
| 4 <b>Dense Cloud generation</b><br>(Dense correspondence matching)             | ✓<br>M  | ✓<br>M  | ✓<br>M   | ✓<br>A  | ✓<br>A                        |
| 5 <b>Mesh/Surface generation</b>   | ✓   | ✓   | ×  | ×   | ×                             |
| 6 <b>Texture generation</b>  | ✓<br>M  | ✓<br>M  | ×  | ×   | ×                             |
| 7 <b>Cloud/Mesh Editing</b><br>(Optional cleaning features)                    | ✓<br>M  | ×   | ×  | ×   | ×                             |

\* M = Manual export, A = Auto export

## 6.2. Reconstruction

Three of the four software packages were able to compute models from the two datasets as presented in Tables 3 and 4. This paper presented a qualitative ranking based on the computation time and heat maps of distances. The heat maps make it clear that, in both cases, COLMAP and Regard3D, that they have produced the best results, i.e. few deviations from the ground truth (e.g., the Metashape model). However, PPT GUI produced the ‘noisiest’ clouds and failed to produce a single set of point clouds from the sculpture data set. The pipeline is often fragile and returned several unsuccessful outputs while dealing with a scale factor to run bundler. Additionally, lack of documentation and user support is still a significant issue and this might deter many novice users from selecting PPT GUI.

Being free software, COLMAP offers a pleasant GUI and plenty of tweaking options. However, in our experiment, it crashed several times (with both datasets) during the dense reconstruction phase. This dataset only worked when the ‘Stereo’ phase ‘max\_image\_size’ is set to 750. The automatic reconstruction, however, has not crashed but it produced less impressive results. Regard3D, on the other hand, was found to be more balanced in computation, documentation, GUI and tweaking options. Moreover, it is entirely free and can produce surface and texture. None of the FOSS applications offers any editing of point cloud or mesh, the user therefore needs to use an external application such as MeshLab (free software, can be downloaded from [www.meshlab.net](http://www.meshlab.net)) for further cleaning and editing 3D.

Based on this comparative study, Regard3D is our preferred application because of its academic licensing, runtime performance and quality of the produced 3D. COLMAP is our second most recommended application, followed by VisualSfM and PPT GUI. However, in regards to our benchmarking, a few issues need to be raised. First, the ‘ground truth data’, which was used to conduct the accuracy test, was obtained from another software application, Metashape. Nor was any laser scan LIDAR data used. Furthermore, we used the default settings of the FOSS applications; different settings would impact their output and any subsequent results.

## 7. Concluding Remarks

Defining and pointing to a specific 3D image-based modelling program as the ‘best’ free and open source (FOSS) solution is a difficult task. The four selected FOSS applications offer various options for handling data sets, tweaking options, computation time, the usability, GUI, and the learning curve.

We tested these applications with seven different challenging 3D objects (datasets) as a pilot study, among these two datasets were finally chosen for the final test as they produced the most successful results. Workflows of these applications were studied first, and later their reconstruction results were evaluated against ground truth objects on the basis of distance measurement and computation time.

The most promising thing we found during the study that, each of these programs have very similar basic workflows. PPT GUI seems less sophisticated for beginners in both the GUI and the outputted results. Both COLMAP and Regard3D offer a sophisticated and clean GUI, which could support a wide range of users' needs. However, both COLMAP and PPT GUI crashed several times during the study. On the other hand, VisualSfM produced relatively good results. However, the GUI is non-intuitive, and the learning curve seems a little steep for new users. Considering the process automation, the processing time, the GUI, the density of the point cloud and, not the least, the accuracy; we, therefore, rank the study software starting with the best one as Regard3D, COLMAP, VisualSfM and PPT GUI.

This paper set out to investigate 3D reconstruction (based on FOSS) solutions while keeping in consideration the potential benefits for small museums, heritage institutes, interested community, and local groups who are currently lacking high-end technological resources and related skills but interested in developing 3D heritage objects for documentation, visualisation, knowledge sharing and showcasing heritage assets. Regard3D, a free and open source software is therefore suggested as the most convenient solution because it is easy to install, requires no programming knowledge to use, can produce a significant good result with relatively low computation time, and offers a smooth learning curve. The official website also provides detailed documentation and tutorials. When the main purpose of the project is not primarily for scientific analysis or study, and where the project objectives demand only visualization and presentation, we believe this investigation will help non-expert users to understand and select the most suitable software for producing image-based 3D models at low cost.

**Author Contributions:** Conceptualization, Investigation, Resources & Writing: H.R.; Supervision, Review and Editing: E.C.

**Funding:** This research was funded by MCASI Small Grant 2017 by Curtin University, Australia.

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

## References

1. Nguyen, M.H.; Wünsche, B.; Delmas, P.; Lutteroth, C. 3D models from the black box: Investigating the current state of image-based modeling. In Proceedings of the 20th International Conference on Computer Graphics, Visualisation and Computer Vision (WSCG 2012), Pilsen, Czech Republic, 25–28 June 2012.
2. Rahaman, H.; Champion, E.; Bekele, M.K. From Photo to 3D to Mixed Reality: A Complete Workflow for Cultural Heritage Visualisation and Experience. *Digit. Appl. Archaeol. Cult. Herit.* **2019**, *13*. [[CrossRef](#)]
3. Durand, H.; Engberg, A.; Pope, S.T. *A Comparison of 3d Modeling Programs*; University of California is Editor, ATON Project/CREATE, D.o.M: Santa Barbara, CA, USA, 2011; pp. 1–9.
4. Grussenmeyer, P.; Al Khalil, O. A comparison of photogrammetry software packages for the documentation of buildings. In Proceedings of the Mediterranean Surveyor in the New Millennium, Malta, 18–21 September 2000.
5. Wang, Y.-F. *A Comparison Study of Five 3D Modeling Systems Based on the SfM Principles*; Technical Report 2011–01; Visualize Inc.: Goleta, CA, USA, 2011; pp. 1–30.
6. Bolognesi, M.; Furini, A.; Russo, V.; Pellegrinelli, A.; Russo, P. Accuracy of cultural heritage 3D models by RPAS and terrestrial photogrammetry. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 113–119. [[CrossRef](#)]
7. Deseilligny, M.P.; Luca, L.D.; Remondino, F. Automated image-based procedures for accurate artifacts 3D modeling and orthoimage generation. *Geoinf. FCE CTU* **2011**, *6*, 291–299. [[CrossRef](#)]
8. Oniga, E.; Chirilă, C.; Stătescu, F. Accuracy Assessment of a Complex Building 3d Model Reconstructed from Images Acquired with a Low-Cost Uas. In Proceedings of the ISPRS-International Archives of the

- Photogrammetry, Remote Sensing and Spatial Information Sciences, Nafplio, Greece, 1–3 March 2017; pp. 551–558.
9. Knapitsch, A.; Park, J.; Zhou, Q.-Y.; Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph. (TOG)* **2017**, *36*, 78. [[CrossRef](#)]
  10. Santagati, C.; Inzerillo, L.; Di Paola, F. Image based modeling techniques for architectural heritage 3D digitalization: Limits and potentialities. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXIV International CIPA Symposium, Strasbourg, France, 2–6 September 2013.
  11. Hafeez, J.; Hamacher, A.; Son, H.-J.; Pardeshi, S.; Lee, S.-H. Workflow Evaluation for Optimized Image-Based 3D Model Reconstruction. In Proceedings of the International Conference on Electronics, Electrical Engineering, Computer Science (EEECS): Innovation and Convergence, Qingdao, China, 10–13 August 2016; pp. 62–65.
  12. Koutsoudis, A.; Arnaoutoglou, F.; Pavlidis, G.; Tsiafakis, D.; Chamzas, C. A versatile workflow for 3D reconstructions and modelling of cultural heritage sites based on open source software. In Proceedings of the Virtual Systems and Multimedia Dedicated to Digital Heritage Conference, Limassol, Cyprus, 20–26 October 2008; pp. 238–244.
  13. Fuhrmann, S.; Langguth, F.; Goesele, M. MVE-A Multi-View Reconstruction Environment. In Proceedings of the EUROGRAPHICS Workshops on Graphics and Cultural Heritage, Darmstadt, Germany, 6–8 October 2014; pp. 11–18.
  14. Schöning, J.; Heidemann, G. Evaluation of multi-view 3D reconstruction software. In Proceedings of the International Conference on Computer Analysis of Images and Patterns, Valletta, Malta, 22–24 August 2015; pp. 450–461.
  15. Skarlatos, D.; Kiparissi, S. Comparison of laser scanning, photogrammetry and SFM-MVS pipeline applied in structures and artificial surfaces. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *3*, 299–304. [[CrossRef](#)]
  16. Debevec, P.E.; Taylor, C.J.; Malik, J. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In Proceedings of the 23rd Annual Conference on Computer graphics and Interactive Techniques, Berkeley, CA, USA, 4–9 August 1996; pp. 11–20.
  17. Xiao, J.; Fang, T.; Tan, P.; Zhao, P.; Ofek, E.; Quan, L. Image-based Façade Modeling. *ACM Trans. Graph. (TOG)* **2008**, *27*, 161. [[CrossRef](#)]
  18. Brown, M.; Lowe, D.G. Unsupervised 3D object recognition and reconstruction in unordered datasets. In Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, Ottawa, ON, Canada, 13–16 June 2005; pp. 56–63.
  19. Snavely, N.; Seitz, S.M.; Szeliski, R. Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph. (TOG)* **2006**, *25*, 835–846. [[CrossRef](#)]
  20. Agarwala, A.; Agrawala, M.; Cohen, M.; Salesin, D.; Szeliski, R. Photographing long scenes with multi-viewpoint panoramas. In Proceedings of the ACM SIGGRAPH, Boston, MA, USA, 30 July–3 August 2006; pp. 853–861.
  21. Remondino, F.; Pizzo, S.D.; Kersten, T.; Troisi, S. Low-Cost and Open-Source Solutions for Automated Image Orientation—A Critical Overview. In Proceedings of the Progress in Cultural Heritage Preservation, Limassol, Cyprus, 29 October–3 November 2012; pp. 40–54.
  22. De Reu, J.; Plets, G.; Verhoeven, G.; de Smedt, P.; Bats, M.; Cherretté, B.; de Maeyer, W.; Deconynck, J.; Herremans, D.; Laloo, P.; et al. Towards a three-dimensional cost-effective registration of the archaeological heritage. *J. Archaeol. Sci.* **2012**, *40*, 1108–1121. [[CrossRef](#)]
  23. Nausner, B. Temple complex ‘Virtual Nako’—3D Visualisation of cultural heritage in Google Earth. In *True-3D in Cartography*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 349–356.
  24. Sauerbier, M. Image-Based Techniques in Cultural Heritage Modeling. In *Scientific Computing and Cultural Heritage*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 61–69.
  25. Dhonjua, H.; Xiaob, W.; Sarhosis, V.; Mills, J.; Wilkinson, S.; Wang, Z.; Thapa, L.; Panday, U. Feasibility Study of Low-Cost Image-Based Heritage Documentation in Nepal. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 237–242. [[CrossRef](#)]
  26. Bertocci, S.; Arrighetti, A.; Bigongiari, M. Digital Survey for the Archaeological Analysis and the Enhancement of Gropina Archaeological Site. *Heritage* **2019**, *2*, 848–857. [[CrossRef](#)]



27. Scianna, A.; La Guardia, M. Survey and Photogrammetric Restitution of Monumental Complexes: Issues and Solutions—The Case of the Manfredonic Castle of Mussomeli. *Heritage* **2019**, *2*, 774–786. [[CrossRef](#)]
28. Boochs, F.; Heinz, G.; Huxhagen, U.; Müller, H. Low-cost image based system for nontechnical experts in cultural heritage documentation and analysis. In Proceedings of the XXI International CIPA Symposium, Athens, Greece, 1–6 October 2007.
29. Kersten, T.P.; Lindstaedt, M. Image-based low-cost systems for automatic 3D recording and modelling of archaeological finds and objects. In Proceedings of the Euro-Mediterranean Conference, Barcelona, Spain, 2–3 April 2012; pp. 1–10.
30. Nikolov, I.A.; Madsen, C.B. *Benchmarking Close-Range Structure from Motion 3D Reconstruction Software under Varying Capturing Conditions*; Springer: Cham, Switzerland, 2016; pp. 15–26.
31. Chaiyasarn, K.; Bhadrakom, B. Automatic image-based reconstruction of historical buildings from Ayutthaya. In Proceedings of the 20th National Convention on Civil Engineering, Chonburi, Thailand, 8–10 July 2015; pp. 1–6.
32. Singh, S.P.; Jain, K.; Mandla, V.R. Image based 3D city modeling: Comparative study. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 537. [[CrossRef](#)]
33. Moulon, P.; Bezzi, A. *Python Photogrammetry Toolbox: A Free Solution for Three-Dimensional Documentation*; ArcheoFoss: Napoli, Italy, 2011; pp. 1–12.
34. Besl, P.J.; McKay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).