Pierre Moulon
Alessandro Bezzi

Python Photogrammetry Toolbox: A free solution for Three-Dimensional Documentation

Abstract

The modern techniques of Structure from Motion (SfM) and Image-Based Modelling (IBM) open new perspectives in the field of archaeological documentation, providing a simple and accurate way to record three-dimensional data. In the last edition of the workshop, the presentation "Computer Vision and Structure From Motion, new methodologies in archaeological three-dimensional documentation. An open source approach. " showed the advantages of this new methodology (low cost, portability, versatility …), but it also identified some problems: the use of the closed feature detector SIFT source code and the necessity of a simplification of the workflow.

The software Python Photogrammetry Toolbox (PPT) is a possible solution to solve these problems. It is composed of python scripts that automate the different steps of the workflow. The entire process is reduced in two commands, calibration and dense reconstruction. The user can run it from a graphical interface or from terminal command. Calibration is performed with Bundler while dense reconstruction is done through CMVS/PMVS. Despite the automation, the user can control the final result choosing two initial parameters: the image size and the feature detector. Acting on the first parameter determines a reduction of the computation time and a decreasing density of the point cloud. Acting on the feature detector influences the final result: PPT can work both with SIFT (patent of the University of British Columbia - freely usable only for research purpose) and with VLFEAT (released under GPL v.2 license). The use of VLFEAT ensures a more accurate result, though it increases the time of calculation.

Python Photogrammetry Toolbox, released under GPL v.3 license, is a classical example of FLOSS project in which instruments and knowledge are shared. The community works for the development of the software, sharing code modification, feed-backs and bug-checking.

1. Introduction

3D Digital copy can be done by various technology, laser (ground), lidar (aerial), structured light, photogrammetry... They have their pros and cons. Laser and Lidar are accurate (millimeter precision) but expensive, even in rental agency, and their use requires formation. Photogrammetry is more and more accessible with the recent progress in electronics that make compact digital camera cheaper but less precise (centimeter precision). Photogrammetry with consumer camera does not reach the same performance as the laser but is accessible to anybody. Today Computer Vision algorithms are mature to be used by non technical users. It's a very active research domain and a lot of progress have been done in the last decade. Such progress are visible with web-service like the Microsoft Photosynth

Project[1].

Our objective consist in providing a tool-chain in order to make 3D digital copy easy. This tool-chain should be Free, Open-source and Cross-Platform to be accessible without constraint. Our pipeline draws largely from existing solution that have proven to be functional and adequate.

2. Related Work

The reconstruction of a scene captured from different viewpoints by a set of 2D images is a computer vision problem that have been studied for decades. Such 3D reconstructions are able to describe the structure (3D points) of the scene and the configuration (motion) of the camera for the registered pictures.

The 3D reconstruction problem could be decomposed into three main steps:

1. Correspondences between pairs of images are found and 3D configuration of image pairs are estimated (estimation of relative camera pose),
2. The two-view geometries are fused in a common coordinates system (estimation of global camera pose),
3. Having a complete camera calibration, a homogeneous dense model of the scene surfaces is computed using all images.

While the two-view camera calibration is a well-studied problem, the multi-view camera calibration remains a challenging task. This multi-view calibration is crucial as it will determine the precision of the scene reconstruction and the quality of the resulting dense 3D model.

The most impressive progress in SfM and MVS make possible to compute a 3D representation of a city from web images[2], with aerial or ground images. But only a few free or open solution exists. Recent research gave birth to companies that have efficient products such as Pix4d[3] and Acute3D[4].

A short history 3D reconstruction from pictures starts with project like Façade[5], Canoma[6] and PhotoModeler[7] that use corresponding points selected by the user in various pictures to determine the 3D position of cameras and thus provide the user with an interface to model the pictured scene by hand . The process was long and required a lot of expertise. Progress in image matching and wide baseline matching allows now to determine the correspondences automatically thanks to contributions like SIFT[8] or SURF[9].

In recent years, a lot of commercial products and web services make 3D reconstruction more accessible but computed 3D data are not always provided to the users. The main example is Photosynth. It only provides a 3D visualization service to travel through the set of photos but 3D data cannot be used for your own purpose. The web service uses the cloud as a storage and computational platform, so the pictures and computed data usage are not under your control. Some web services are free (ARC3D[10] or CMP SFM WebService[11]) but again they use a cloud for computation so there is no of the usage of the

1 http://photosynth.net
2 AGARWAL *et alii* 2009
FRAHM *et alii* 2010
3 http://www.pix4d.com/
4 http://www.acute3d.com/
5 http://ict.debevec.org/~debevec/Research/
6 http://www.canoma.com/
7 http://www.photomodeler.com/
8 LOWE 2004
9 BAY *et alii* 2008
10 http://homes.esat.kuleuven.be/~visit3d/webservice/v2/
11 http://ptak.felk.cvut.cz/sfmservice/

data (pictures).

Thanks to the emergence of OpenSource frameworks (Bundler[12], CMVS/PMVS[13]) that perform multi-view calibration and dense 3D point cloud computation, we aim to develop a free and easy to use pipeline in order to make 3D digital copy easy. We provide the user a self-contained solution that gives him control on the whole data flow. As pictures are the property of the user we chose a user-side pipeline. The main drawback of our approach is that computation speed depends from user's computer. It could have some drawbacks for large scenes or large images but a compromise between performance and quality can be made by reducing image size dynamically in the toolbox.

3. 3D from pictures, the basis

Building 3D "model" from pictures consists in recover 3D camera positions related to pictures and 3D positions of particular content of the images. It is done by identifying similar content between N views and solve 3D geometry problems. User input consist of an image collection and camera parameters. The computed output is a set of 3D camera positions and 3D points (fig. 1).

3D from pictures is an active research domain that rely on Computer Vision and more specifically, Image retrieval/matching, Structure from Motion (SfM) and Multiple View Stereovision (MVS).

- Image matching finds common local sub-image between two pictures.
- Structure from Motion estimates the relative camera position from anchor points computed at the previous step.
- Multiple View Stereovision estimate a dense representation of the 3D model (Dense point cloud).

3.1 Image Matching

Image matching identifies pictures that can be used to compute the relative orientation of 2 camera and thus to calibrate a network of images. This process of image matching is performed in 3 steps:

1. compute local content on each image(Feature and Descriptor computation, for instance SIFT),
2. find putative matches between two pictures (find the nearest descriptor in the other image of the pairs),
3. check the geometry of the putative matches (Epipolar geometry).

Once the image matching between all possible pair is performed, a geometric graph is built (fig. 2). An edge is added if a geometric connection exists between two pictures.

3.2 Camera Pose estimation

Camera pose estimation finds the camera positions by solving the relative pose estimation problem. Relative pose estimation consists in estimating a rigid motion between two cameras, a rotation R and a translation T (fig. 3). This relative geometry between two

12 SNAVELY 2008
13 FURUKAWA 2010

views is faithfully described by an "Essential" matrix[14].

This Essential 3×3 matrix relates corresponding points in stereo images assuming that the cameras satisfy the pinhole camera model. This E matrix could be computed from 8 points with a linear method or with 5 points[15]. The 5 points method is preferred because it is the minimal case and it allows to add more constraint on the estimated matrix and so provide more accurate results. The image matching step is thus crucial: the more common points we get between pictures we will get, the more images we can estimate 3D positions accurately.

The position of a camera can also be computed from correspondences between 3D points and corresponding projections in the image plane. This 3D-2D correspondence problem is known as Resection (fig. 4). It consists in estimating Pi (rotation, translation and internal parameters of the camera) with a ray constraint geometry. It finds the Pi configuration that minimize the re-projection errors between the rays passing through optical camera center to 3D points and the 2d image plane coordinates. Once two cameras are related with an Essential matrix and 3D points X are build, we can add incrementally new camera to the scene by using successive resection.

Based on those computations (Essential, Resection) we can perform Incremental Structure from Motion. It's the algorithm that is implemented in the 3D calibration software we use (Bundler).

3.3 Incremental Structure from Motion

Bundler is one of the state-of-the-art implementation of incremental SfM. It takes as input a image series and camera information (like focal values extracted from Exif jpg data and CCD sensor size available on camera manufacturer website or dpreview.com).

From an initial image network, Bundler chooses a pair of images, computes the relative pose with the Essential Matrix and try to add incrementally the remaining images in the 3D scene by using successive resections. In order to avoid incremental error, bundle adjustments[16] is used to refine non linearly the estimated camera parameters and 3D point positions, and thus reduce the error across computed data whose size is growing.

This pseudo algorithm can be pictured as in figure 5:

Input:

- image network of geometrically coherent pictures,
- internal camera parameters (Focal length, CCD sensor size).

Output:

- camera position,
- sparse point cloud.

Bundler suffers from some defaults. It's code is not very clean and sometimes the 3D reconstruction fails due to drift error. But it have the advantage of being nearly the only Open-Source viable solution over internet with such performance. Recent community initiative like the libmv project[17] is a prelude to a cleaner implementation of "Bundler clones". This bricks could be replaced in the tool-chain in a near future.

3.4 Multiple View Stereovision

14 http://en.wikipedia.org/wiki/Essential_matrix
15 NISTER 2004
16 http://en.wikipedia.org/wiki/Bundle_adjustment
17 http://code.google.com/p/libmv/

Multiple View Stereovision (MVS) consists in mapping image pixel to 3D points fcposes, images point cloud. This dense representation can be a dense point cloud or a dense mesh. In order to find a 3D position for each corresponding pixel of the image sequence, MVS uses multiple image to reduce ambiguities and estimate accurate content (fig. 6).

One of the interesting state-of-the-art method is the Patch approach called PMVS (Patch MultiView Stereo)[18]. It is based on a seed growing strategy. It finds corresponding patches between images and locally expand the region by an iterative expansion and filtering steps in order to remove bad correspondences (fig. 8). Such an approach finds additional correspondences that were rejected or not found at the image matching phase step.

Figure 9 shows benefit of using PMVS (empty 3D zones correspond to poorly textured or too ambiguous image zones):

4. The Python Photogrammetry Toolbox

The Python Photogrammetry Toolbox (PPT)[19] implements a pipeline to perform 3D reconstruction from a set of pictures. It design follows the classic reconstruction process. It takes pictures as input and performs automatically the 3D reconstruction for the images for which 3D registration is possible. PPT hides from the user the boring task of data conversion and files listing that are required to communicate through the various software components of the chain. Open-source software has been chosen to perform the intensive computational parts of the reconstruction pipeline, Bundler for the camera pose estimation and CMVS/PMVS for the dense point cloud computation.

Initially Bundler and CMVS/PMVS are provided with some shell scripts that automates launching tasks, but one of the main drawback of shell is that it is not cross-platform. It cannot be used under Windows. Compilation of those software are not managed through the same basic interface (Makefile on Linux and vcproj on Windows) and so requires double maintenance for smooth compilation on both platform. Design choices in PPT make it cross-platform:

- it uses Python[20] as a cross-platform script language to handle communication and software launching operations. It handles all the tasks that are required for our purpose (directory listing, file listing, images conversion, Exif reading, Sqlite database management);
- it uses Cmake[21] for the compilation configuration of the chosen Open-Software that is available under the Open Source Photogrammetry code repository**[22].**

PPT provides a tool-chain that is easier to maintain and use than the previous approaches. It defines a clear pipeline to handle 3D reconstruction. This pipeline is designed as python module with a High Level API in order to be extensible in the future. It results in a 3-level application: Interface, Python modules and Software.

A graphic wrapper has been developed to hide the command-line calls that are required to use the chain through python modules. It provides a 2-step reconstruction workflow.

The multi-level application makes maintenance easier. Each bottom module can be updated as long it respects the designed High Level API. It makes the interface easily extensible. For example the python wrapper use a design pattern interface in order to have various feature detection/description algorithm for the image matching step (the user can use

18 FURUKAWA 2010

19 Source code is accessible from http://code.google.com/p/osm-bundler/

20 http://www.python.org/

21 http://www.cmake.org/

22 https://github.com/TheFrenchLeaf

the David Lowe SIFT[23] or the Open-source implementation VLFEAT[24]).

Data workflow is organized in a temp directory created at the beginning of the process. All the required data to process the 3D reconstruction is located in this directory. Data is updated by the different element of the tool-chain and showed at the end to the user via a directory pop-up. The main workflow is illustrated in figure 10. It's interesting to take a closer look to the 2-step process workflow (RunBundler and RunCMVS) to better see the job of the python scripts.

RunBundler (fig. 11) performs the camera calibration step. It computes the 3D camera pose from a set of image with corresponding "camera model"/"CCD width size" embedded in an Sqlite database. In figure 11, orange coloured items (bottom squares) are the created files. We recognize image matching tools (sift, matchFull) and the 3D pose estimation software (Bundler).

RunCMVS (fig. 12) takes as input the images collection, cameras poses and perform the dense 3D point cloud computation. Data conversion from Bundler format to CMVS/PMVS format is done by using Bundle2PMVS and RadialUndistort. Dense computation is done by PMVS as well as CMVS, that is an optional process to divide the input scene in many smaller instance that make the process of dense reconstruction faster.

PPT-Gui (fig. 7) is the graphical interface to interact easily with the photogrammetry toolbox[25]. The Gui part is powered by PyQt4[26], a multi-platform gui manager. The interface is designed in two different parts: a main window composed by numbered panels which allows the user to understand the steps to perform, and a terminal window in which the process is running. The GUI is deliberately simple and it is build for people who are not familiar with command-line scripts. The four panels lead the user to the end of the process through only two steps: Run Bundler (panel 1) and Run CMVS\PMVS (panel 2). Running CMVS before PMVS is highly recommended, but not strictly necessary: there is also the possibility to use directly PMVS (panel 3). Panel 4 provides a fast solution to integrate the SQL database with the CCD width (mm) of the camera, without using external software.

5. Application

Archaeological field activity is mainly a working process which ends, in most cases, with the complete destruction of the site. Usually a ground layer is excavated to investigate the underlying level. In the lack of particular expensive equipment (laserscanner, calibrated camera) or software (photogrammetric applications), field documentation is composed by pictures (digital or films), manual drawings, total station measurements and bi-dimensional photo-mosaics. At best all the data are connected together inside a Geographical Information System (GIS).

The last years' progresses of Computer Vision open new perspectives, giving to everybody the possibility to record three-dimensional data. The benefits of this technique are different:

- it is a software-based technology in continuous developing (data analysed today could be processed in future ending in better results),
- it is well represented by Free and Open Source solutions,
- it needs only the equipment which is normally used in an excavation (digital camera and total station),
- easily portable hardware components allow archaeologists to work under critical or extreme conditions (e.g. in high mountain, underwater or inside a cave),

23 http://www.cs.ubc.ca/~lowe/keypoints/

24 http://www.vlfeat.org/

25 Source code is accessible from https://github.com/archeos/ppt-gui/

26 http://wiki.python.org/moin/PyQt4

- the flexibility of this technique facilitates the documentation of a wide range of situations.

The next chapters introduce some examples of application in different scales: from macro (layers, structure) to micro (finds).

5.1. Layers

Normally archaeological layers are documented using photomapping techniques, a bi-dimensional projection of reality which produce rectified images starting from zenith pictures and ground control points (GCP). Using the same instruments (digital camera and total station) it is possible to record also the morphology of the level (figg. 13-14). The data acquisition is fast and simple: it consists exclusively in taking pictures of the area of interest (both horizontal surfaces and vertical sections) paying attention to include at least three measured marks, which will be used in data processing to georeference the final model. The same rules of the traditional photography are to be followed: centre the desired object in each picture, avoid extreme contrast shadow/sun, use a tripod in low-light condition.

There are no limits around the number of images: it depends on the complexity of the surface and on the power of the hardware (RAM) which will process the data.

5.2. Architectonic Structure

This technique is particularly indicated for architectonic monuments (figg. 15-16). In this case the main difficulty is to cover the whole object with a good photo set, in order to avoid holes in the final mesh. Most of the time it is possible to solve logistical problems related to the complexity of the structure, using specific hardware like telephoto lenses or remote sensing devices (UAV).

5.3 Finds (artefacts and ecofacts)

Archaeological finds can be documented “in situ”, taking pictures moving around the object (fig. 17), or in laboratory using a turntable and a black background (fig. 18). In this last case the position of the camera during data acquisition was fixed, but will be split in more poses during data processing (fig. 19). Good results were reached taking a picture each 10 degrees, 36 photos for a complete revolution of the object. It is possible to use macros to obtain model of a small artefact.

5.4 3D-data recovering from old or historical photo sets

One the most interesting approaches of SfM is the ability to extract three-dimensional information from old or even historical photographs taken by amateurs for other purposes. The critical point of this application is to reach the minimal number of images needed to start the reconstruction process (3). It is much easier to find an appropriate photographic documentation since digital cameras have become a widespread phenomenon (figg. 20-21). If a monument in the present is not longer in his original conservation status or even completely destroyed (e.g. the Banyan Buddhas), Computer Vision is a valid method to recover the original morphology.

6. Conclusion

Python Photogrammetry Toolbox (PPT) is an user-friendly application to perform 3D digital copies of pictured scenes. It provides a low-cost, portable solution that opens a direct access to Structure from Motion (SfM) and Image-Based Modelling (IBM) to every owner of a consumer camera. It opens particularly interesting perspective in the field of archaeological documentation due to the fact that a reflex camera is much cheaper than a laser scanner. A possible drawback of the current solution is that it uses a feature detector/descriptor for image matching (SIFT algorithm) that is under Patent in the USA. A rewritten and optimized version of SIFT is included inside VLFeat (an Open and Portable Library of Computer Vision Algorithms, released under GPL v. 2). This can't completely solve the license problem but most of the users are aware of those constraints.
A direct comparison between SfM/IBM and hardware technology (laser scan) or other photogrammetric applications is certainly possible, but it is not the objective of this article. All these techniques are different in their approach, but they lead to similar results. The choice of one of these methods depends on various factors: environmental characteristics of the site, economic budget of the project and technical skills of the staff. Anyway SfM/IBM is able to satisfy some of the basic needs of a typical archaeological project: the reduction of costs related to equipment, a fast and simple data collection process and a low-interactive and easy data processing. For these reasons SfM/IBM is a viable alternative to more expensive (laserscan) or more technically complex (stereo-photogrammetric restitution) methods. From an archaeological point of view, the final intent is to acquire three-dimensional morphology of the layers that the excavation irreparably destroyed and to create a virtual copy of the archaeological record to allow continuous monitoring and further analysis. The good results achieved in a such fast way can be used to extract 3D volume (voxel) of each stratigraphic level, applying free software like GRASS, Blender and ParaView[27].

PPT is an open source solution, that make 3D reconstruction from images easier, in which user contribution will produce benefit for all the community. It is a example of how the combination of FLOSS projects can contribute to scientific and methodological progress. The future implementation will consider multi-thread computation, performance improvement and functionality addition.

Acknowledgements

This work have been made possible due to many individual Open-Source initiative. We thanks particularly Noah Snavely for Bundler sources, Yasutaka Furukawa for CMVS/PMVS sources and Vladimir Elistratov for the osm-bundler initiative.

Bibliography

AGARWAL 2009
S. Agarwal - N. Snavely - I. Simon - S. M. Seitz - R. Szeliski, *Building Rome in a day,* in ICCV 2009, 72-79.

BAY 2008
H. Bay – A. Ess - T. Tuytelaars – L. Van Gool, *SURF: Speeded Up Robust Features,* in CVIU 2008, 346-359.

BEZZI 2006
A. Bezzi – L. Bezzi – D. Francisci – R. Gietl, *L'utilizzo di voxel in campo archeologico*, in Geomatic Workbooks, 6, 2006.

FRAHM 2010
J. -M. Frahm – P. Georgel - D. Gallup – T. Johnson – R. Raguram – C. Wu – Y.-H. Jen – E. Dunn - B. Clipp - S.

27 Bezzi 2006.

Lazebnik, *Building Rome on a Cloudless Day,* in ECCV 2010, 368-381.

FURUKAWA 2010
Y. Furukawa – B. Curless – S. M. Seitz – R. Szeliski. *Towards Internet-scale multi-view stereo*, in CVPR 2010, 1434-1441.

LOWE 2004
D. G. Lowe, *Distinctive image features from scale-invariant keypoints,* in IJCV 2004, 91-110.

NISTER 2004
D. Nister, *An Efficient Solution to the Five-Point Relative Pose Problem*, in IEEE Trans. Pattern Anal. Mach. Intell. 2004, 756-777.

SNAVELY 2008
N. Snavely – S. M. Seitz – R. Szeliski. *Modeling the World from Internet Photo Collections,* in IJCV 2008, 189-210.

Figures

archeologici della Provincia Autonoma di Trento – N. Pisu).

Affiliation

- **Pierre MOULON** IMAGINE/LIGM, University Paris Est & Mikros Image
  http://imagine.enpc.fr http://mikrosimage.eu
  pmo@mikrosimage.eu
- **Alessandro BEZZI** ArcTeam s.n.c.
  http://www.arc-team.com/
  alessandro.bezzi@arcteam.com

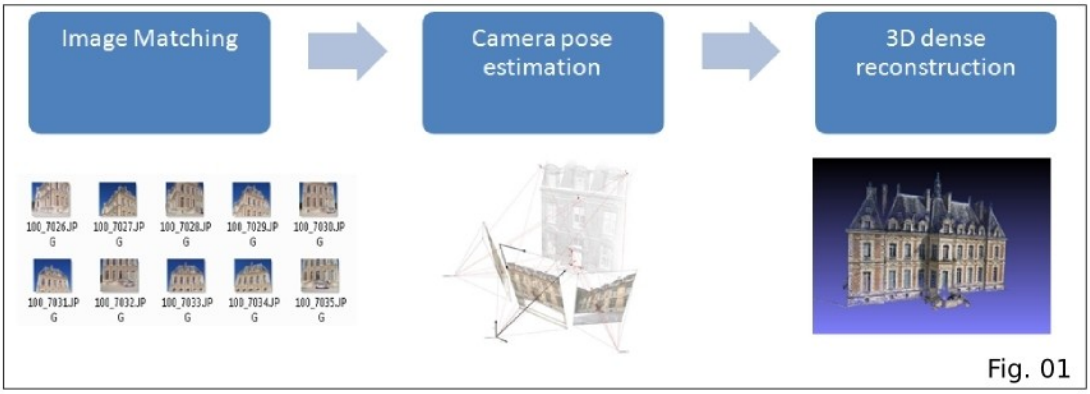

Fig. 01

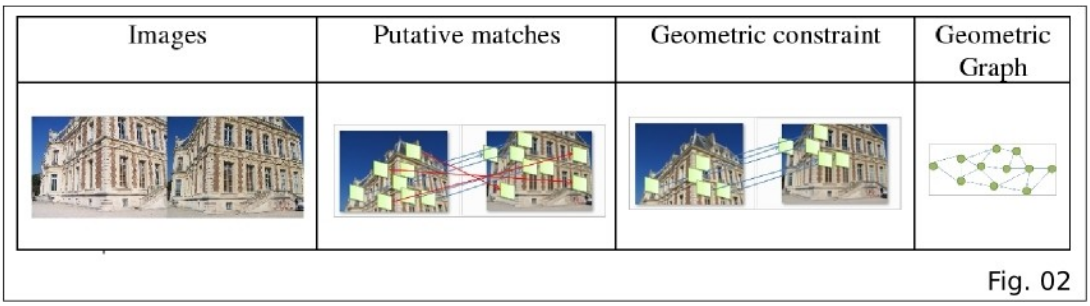

Fig. 02

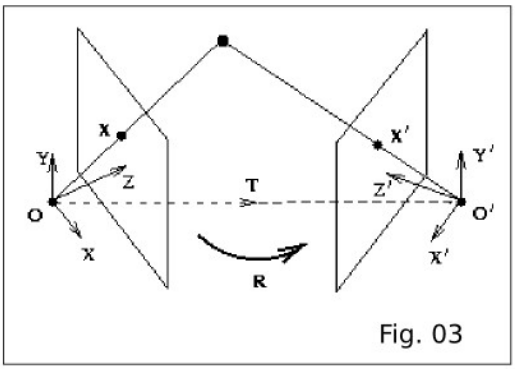

Fig. 03

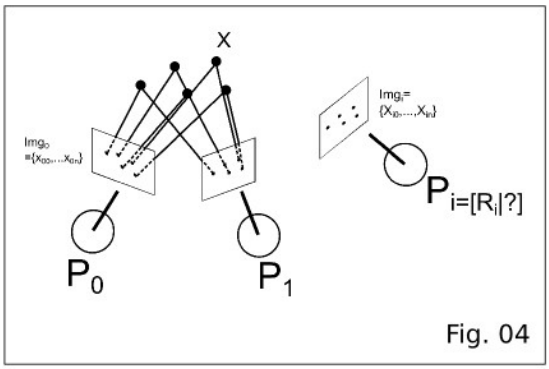

Fig. 04

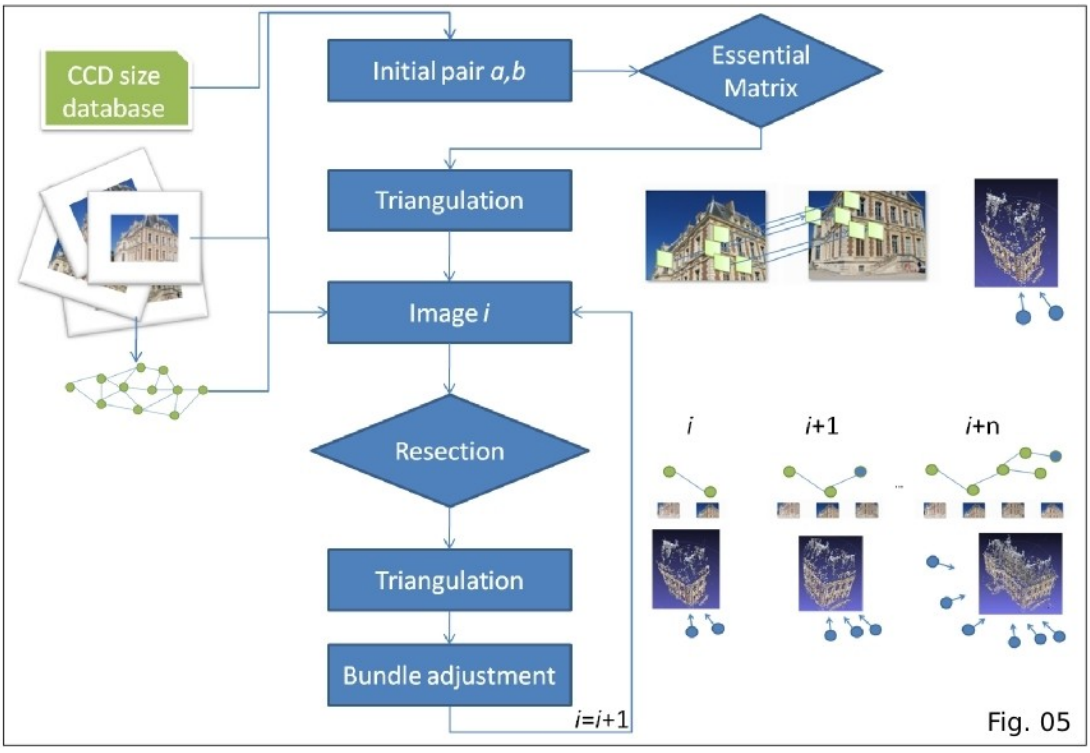

Fig. 05

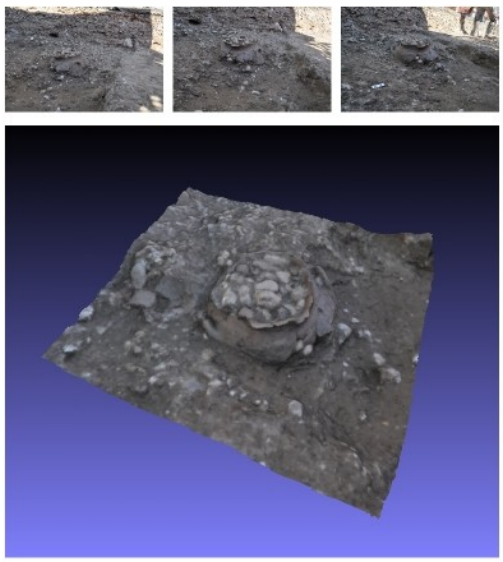

Fig. 17

Fig. 18

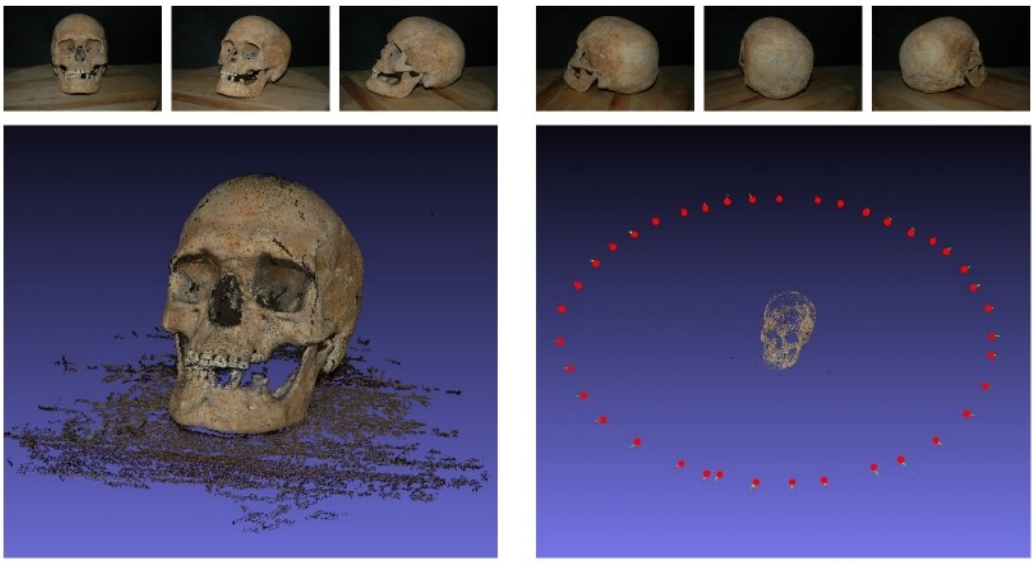

Fig. 19

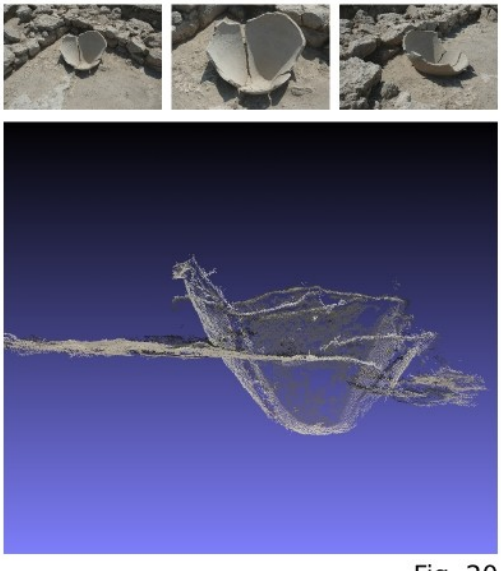

Fig. 20

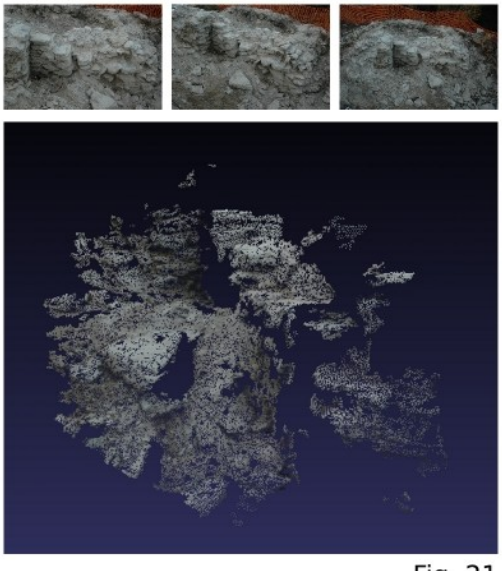

Fig. 21