3DNOW: IMAGE-BASED 3D RECONSTRUCTION AND MODELING VIA WEB

Y. Tefera[a], F. Poiesi[a,*] D. Morabito[b], F. Remondino[b], E. Nocerino[b,c,d], P. Chippendale[a]

[a] Technologies of Vision - Fondazione Bruno Kessler - Povo (TN), Italy - {ytefera, poiesi, chippendale}@fbk.eu
[b] 3D Optical Metrology - Fondazione Bruno Kessler - Povo (TN), Italy - {remondino, morabito, nocerino}@fbk.eu
[c] LSIS, I&M Team, Aix-Marseille Universite, Polytech Luminy, Marseille, France - erica.nocerino@univ-amu.fr
[d] Theoretical Physics, ETH Zurich, Zurich, Switzerland - erican@phys.ethz.ch

Commission II / WG1

KEY WORDS: Photogrammetry, Structure-from-Motion, Dense Image Matching, 3D modeling, Web service.

ABSTRACT:

This paper presents a web-based 3D imaging pipeline, namely 3Dnow, that can be used by anyone without the need of installing additional software other than a browser. By uploading a set of images through the web interface, 3Dnow can generate sparse and dense point clouds as well as mesh models. 3D reconstructed models can be downloaded with standard formats or previewed directly on the web browser through an embedded visualisation interface. In addition to reconstructing objects, 3Dnow offers the possibility to evaluate and georeference point clouds. Reconstruction statistics, such as minimum, maximum and average intersection angles, point redundancy and density can also be accessed. The paper describes all features available in the web service and provides an analysis of the computational performance using servers with different GPU configurations.

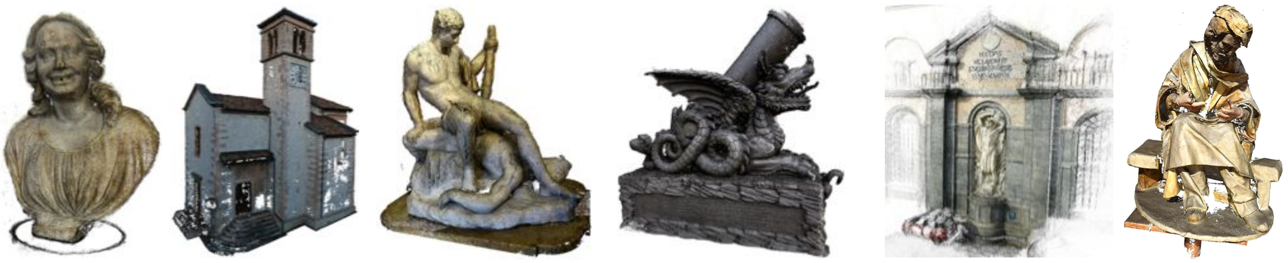


Figure 1. 3D reconstructions of various objects achieved using 3Dnow.

1. INTRODUCTION

We are entering a period in which the demand for quality and novel 3D content is outstripping supply capacity. Uses for such 3D media are diversifying, spanning from 3D documentation, 3D modeling or social media to 3D printing, and Augmented/Virtual Reality (AR/VR). The bottom line is, handcrafted 3D content is expensive to produce and this is holding up innovation.

Improved wireless connectivity and operating system diversity, often means that web (and possibly free) services that can enable non-expert users to produce their own 3D content are becoming a hot topic. Examples of such services include: Autodesk Recap (Autodesk, 2018), PHOV (PHOV, 2018), Skycatch (Skycatch, 2018), PrecisionMapper (PrecisionMapper, 2018) and the pioneering ARC 3D service (ARC3DWeb, 2018), (Vergauwen and Gool, 2006). Thanks to these tools, the interest of users in generating their own 3D content has been sparked, resulting in user generated 3D content being consumed in AR/VR applications using tools like Wikitude Studio (Wikitude, 2018). Although user generated models are typically of a lower quality than those produced with laser scanning or professional SLR cameras, a few years ago experts have shown an interest in using Cloud-based 3D reconstruction services as a resource, for example in architecture and heritage 3D reconstruction (Brutto and Meli, 2012, Abate et al., 2011, Menna and Troisi, 2010, Remondino and Menna, 2008, Kersten and Lindstaedt, 2012, Galizia and Santagati, 2013, Schoning and Heidemann, 2015).

These above factors, together with recent advances in image-based 3D reconstruction tools, have been the driving force behind the implementation of a new freely available photogrammetry web service - 3Dnow[1]. 3Dnow is developed with the idea of encouraging users to upload their own sets of images that can be acquired with any camera device and then generate 3D content (downloadable as point clouds or meshes). Differently from other 3D reconstruction web services (e.g. ARC3DWeb, Recap), 3Dnow targets both experts and non-experts, offering them a host of functionalities, all embedded inside a friendly and freely available web service. No additional software needs to be installed other than a standard web browser. The image processing and 3D reconstruction modules are based on the open source pipeline COLMAP (Schonberger, 2016), which have been customised with new analysis features and editing algorithms. The innovative aspects of 3Dnow are:

- Fully automatic and web-based 3D digitisation pipeline, from images to simplified meshes;
- Intuitive web viewer for 3D reconstruction;
- Fully-automated cleaning procedure of point cloud and meshes;

[*] Corresponding author.

[1] Website: http://3dnow.fbk.eu.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-1097-2018 | © Authors 2018. CC BY 4.0 License.

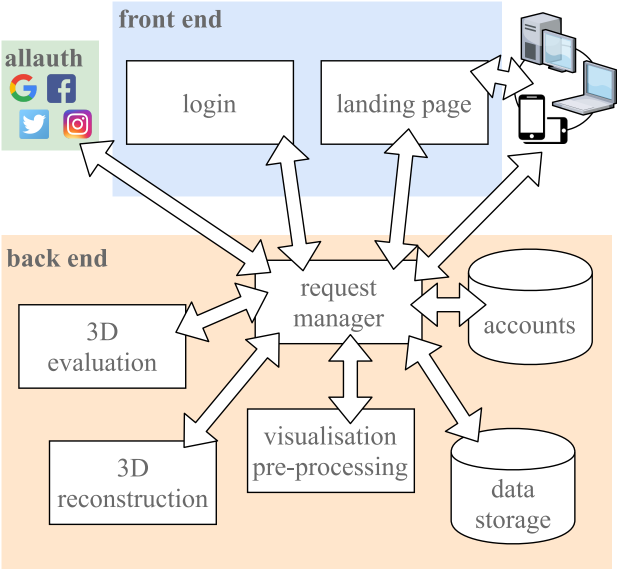


Figure 2. 3Dnow infrastructure.

- Feedback statistics (and visualisation) of the 3D reconstruction processing (e.g. intersection angle, tie points redundancy, RMS reprojection error);
- Scaling and georeferencing.

3Dnow will let users move beyond a typical black-box approach of photogrammetric tools and help them analyse acquisition principles, data processing algorithms and standard metrics to assess the quality of their results (Remondino et al., 2017).

The paper presents the developed infrastructure (Sec. 2), the processing pipeline (Sec. 3), analysis and visualisation functionalities (Sec. 4) and a series of use-cases and applications that have benefited from this web service (Sec. 5). Lastly, we will report challenges, limitations and further developments that will be addressed in future developments of 3Dnow (Sec. 6).

2. INFRASTRUCTURE

3Dnow infrastructure is composed of a front end to facilitate the user to control the reconstruction process and of a back end where the heavy computational load (e.g. 3D reconstruction) is performed (Fig. 2).

2.1 Front end

The front end is divided into two parts: a login page where the user authenticates and a landing page where the user interacts with the 3D reconstruction features. In order to enable users to log in 3Dnow using their own social media or email accounts, we implemented a third-party authentication process that goes through a request manager to validate their credentials. This framework is developed using Allauth (Allauth, 2018). Allauth provides a set of functionalities to handle registrations and account management through social accounts. The first authentication for a user requires an explicit authorisation by the 3Dnow’s administrator. The landing page is the interaction web interface to configure, execute, visualise and analyse 3D reconstructed objects.

The front end is cross-platform and runs on WebGL-enabled web browsers (WebGL, 2018). In order to make the landing page responsive (i.e. suitable for both desktop and mobile devices), we developed the front end using Bootstrap, an open source toolkit based on HTML, CSS and JS (Bootstrap, 2018). The visualisation interface to display 3D reconstruction results is embedded in the landing page and uses Potree, an open-source progressive visualisation tool based on WebGL able to handle large point clouds (Potree, 2018). Potree uses a set of visualisation data that are typically pre-processed on the back end to optimise the rendering performance and then transferred on the front end for use. Fig. 3 shows a screenshot of the front end where we can observe options to execute and analyse different types of reconstructions.

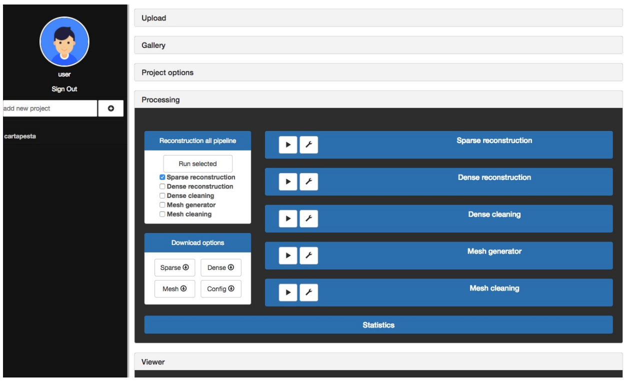


Figure 3. Landing page where the user can configure, launch and visualise reconstruction processes.

2.2 Back end

The back end integrates a request manager that schedules and handles the processes of 3D reconstruction and evaluation, data pre-processing for visualisation, account information fetching and data storage. The back end is multi-GPU and implemented in Django, an open-source Python-based web framework that uses MVT (Model-View-Template, 2018) architectural pattern. MVT is a development framework that promotes code maintenance and expansion. Because multiple users can launch processes simultaneously (e.g. reconstruction, automatic editing, mesh simplification), we integrated a First In First Out (FIFO) multi-queue scheduling strategy to handle concurrent reconstruction processes. We separate queues for CPU and GPU processes. Based on the number of available CPU cores and GPUs on the server, we can evenly distribute the resources for each user. When all the resources are occupied, new processes are queued using the FIFO strategy. The queue manager is developed using Celery framework (Celery, 2018).

3. PHOTOGRAMMETRIC PIPELINE

In order to accommodate users with different expertise, knowledge and needs, digital reconstructions can be executed using default or personalised modalities. The former uses the default COLMAP parameters of camera calibration and image orientation (Structure from Motion - SfM), dense image matching (DIM) and meshing (COLMAP Web, 2018). This modality assumes that each image is taken with a different imaging device and thus SfM estimates different camera parameters for each image. DIM assumes that objects are fairly textured. Meshing uses a octree depth value that is directly related to the size of the dense point cloud. Because the method adapts the parameters based on the sampling density, the default parameters are suitable for large-scale reconstructions or for small-sized objects acquired from a close distance.

The personalised modality allows the user to configure the parameters prior the processing of the images. Amongst the configurable parameters for SfM, the user can choose the camera model to compute interior orientation parameters, the image resolution for the feature extraction stage, minimum number of keypoints necessary to triangulate a 3D point, the minimum intersection angle to create a 3D point and the algorithm to match keypoints between

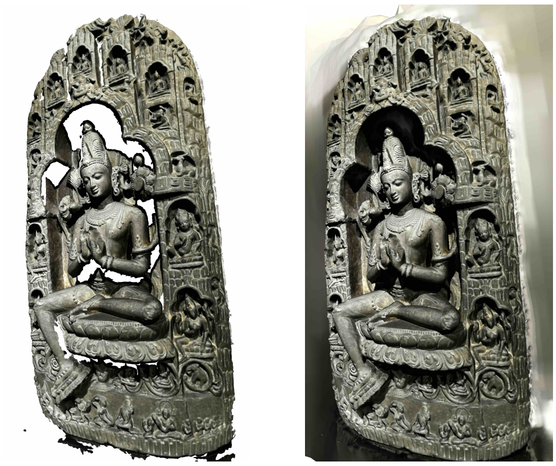

(a) (b)

Figure 4. Meshing performed using Poisson reconstruction (a) with default parameters and (b) with custom parameters.

images. The available matching methods to detect tie points are exhaustive, sequential, vocabulary tree, sequential and transitive (Hartmann et al., 2016). For the DIM step, the user can choose the image resolution and the window radius for the normalised cross correlation (NCC). The window radius specifies the size of the patch that is used to compute the cost of the matching for the disparity estimation. Weakly-textured objects need a large value of the radius, leading to greater computational time.

As far as the meshing is concerned, the user can choose the octree depth and the surface trim value. The octree depth d corresponds to the voxel grid max resolution defined as $2^d \times 2^d \times 2^d$ to which the Poisson meshing is applied to. The trim value is a threshold that defines the minimum number of cloud points within an octree node that is needed for the computation of the surface in that region. As AR/VR applications need downsampled mesh model, in addition to the high-quality mesh produced by Poisson reconstruction, 3Dnow offers the option to reduce the number of polygons using a mesh simplification approach based on quadratic error metrics (Garland and Heckbert, 1997). It is possible to select the desired percentage of polygon reduction that the mesh simplification algorithm will target to achieve. The original (full resolution) version of the meshed reconstruction is retained in the database. Downloadable point clouds and meshes are *ply* and *obj* files, respectively, while the camera poses and processing parameters are *nvm* and metadata file.

Fig. 4 shows an example of meshing performed using Poisson reconstruction with default parameters (surface trim value equal to 10) and with personalised parameters (surface trim value equal to 5). Because a larger trim values discard a larger number of octree nodes, we can observe that the inner surfaces, affected by the casted shadow, have been filtered out in (a). On the other hand, the inner surfaces are maintained in (b), but the noisy 3D points outside the target object lead to noisier surfaces.

4. 3D ANALYSIS AND VISUALISATION

4.1 Metadata and logs

Each reconstruction process generates metadata that a user may want to analyse and visualise for specific applications. Therefore, we developed a procedure to gather this information and include it in a downloadable file. This file contains information about: (i) number of images uploaded and successfully processed, (ii) camera and lens model, (iii) parameters selected for the 3D digitisation pipeline, (iv) number of points triangulated (i.e. sparse point cloud), (v) number of points of the final dense point cloud, (vi) number of vertices and triangles of the 3D mesh model, (vii) bundle adjustment reprojection error and (viii) min/max/average intersection angle.

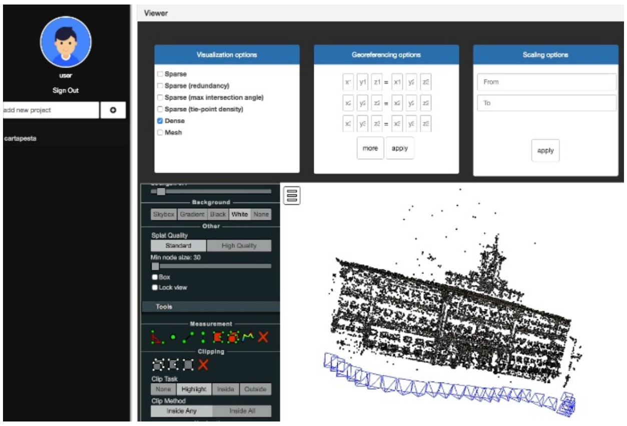

Visualization options
□ Sparse
□ Sparse (redundancy)
□ Sparse (max intersection angle)
□ Sparse (tie-point density)
☑ Dense
□ Mesh

Figure 5. 3D visualisation, scaling and georeferencing tools.

4.2 Visualisation interface

The visualisation uses Potree as it can render point clouds and meshes (Potree, 2018). The converter (on the back end) receives a point cloud as input (i.e. *ply* file) and generates a set of files that can be efficiently visualised with a viewer (on the front end) inside a browser. Within this process of conversion, 3D points are reorganised in efficient data structures that can be loaded by the viewer and visualised progressively while the user navigates within the point cloud. On the other hand, meshes are not converted with this method as they have a different structure, although they can still be visualised in Potree. In addition to the navigation and exploration of 3D models, Potree viewer also offers a set of useful functionalities such as selection and visualisation of point cloud profiles, point cloud clipping, measurements of angles and distances between points.

4.3 Point cloud evaluation tool

3Dnow’s point cloud evaluation has been embedded in the visualisation interface and enables the user to visualise a colour-coded version of the reconstructed sparse point cloud; the colours are based on an online evaluation procedure that is performed without using ground-truth (reference) data.

This point cloud evaluation tool can display density, intersection angle and redundancy associated to each 3D point. The density feature is defined as the number of 3D points contained in a sphere of radius R centred on a certain 3D point. In (Mauro et al., 2014), authors used a fixed-sized radius which does not generalise for point clouds of arbitrary size, which is a key element to make the evaluation applicable to generic point clouds. Therefore, in order to make the evaluation kernel point cloud independent, for each 3D point we compute the Euclidean distance between their closest 100 points, and calculate the value corresponding to the 80^{th} percentile of this distances (including all 3D points) to have an estimate of distances in volumes populated by 3D points. We finally set the radius R as three times the value of the 80^{th} percentile assuming a normally-distributed set of points lying inside three standard deviations. The intersection angle is computed as the maximum angle between the viewing directions of the evaluated cloud point. The smaller this angle, the more uncertain a 3D point. Lastly, the redundancy is defined for each 3D point and indicates the number of cameras that see a 3D point. Fig. 6 shows how these values are visualised using the colour-coded visualisation method.

https://doi.org/10.5194/isprs-archives-XLII-2-1097-2018 | © Authors 2018. CC BY 4.0 License.

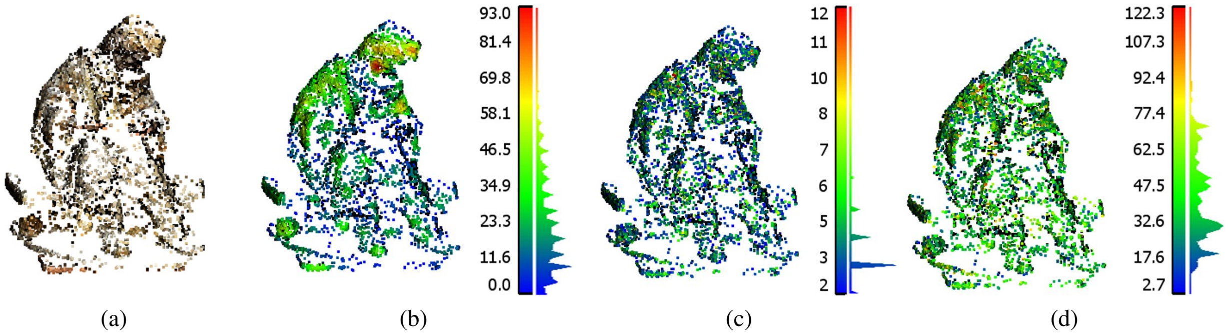


Figure 6. Point-cloud evaluation through colour-coded sparse point cloud: (a) original RGB, (b) 3D point density (3D points per volume), (c) redundancy (tie points per 3D point) and (d) intersection angle (degrees).

In addition to the colour-coded point cloud evaluation, it is possible to visualise graphical statistics of (i) tie points per image, (ii) redundancy and (iii) max intersection angle for 3D points. The graph visualisation is developed using the javascript charting engine Highcharts (Highcharts, 2018). Fig. 7 shows an example of these statistics where we can observe from graph (b) that the majority of 3D points have been triangulated using three images.

4.4 Scaling and georeferencing

The scale and datum ambiguity of the photogrammetric processing is addressed via a dedicated tool and using two solutions: (i) importing a known distance between two points manually picked in the point cloud or (ii) importing the known 3D coordinates of some Ground Control Points marked in the point cloud. The measure of these distance or of the 3D points can be performed directly in the visualisation interface. When the distance has been measured, the user can insert the corresponding values in the appropriate fields above the visualisation interface and apply the changes to rescale the point cloud. Differently from the original implementation of COLMAP, where the georeferencing is performed through camera positions, we developed a tool to enable the user to select at least three 3D points to georeference either the sparse or the dense point cloud.

5. RECONSTRUCTION USE-CASES

To evaluate the performances and reliability of 3Dnow, we processed several datasets. Tab. 1 reports the reconstructed scenes with their main characteristics, while Fig. 8, 10 and 9 show screenshots of the reconstructed scenes. The 3D reconstruction tests were also performed using different server configurations to provide figures about the computational time. The reconstruction algorithm embedded in 3Dnow has been numerically evaluated on a few use-cases in (Nocerino et al., 2017, Knapitsch et al., 2017), where it showed state-of-the-art performances.

5.1 Experiment analysis

We run several experiments with different scales of observation and using images that were captured from diverse types of cameras, such as professional, compact, mobile phones and aerial (Tab. 1). Acquisitions range small (Buddha Thailand - Fig. 8a,b) to large objects (Dortmund - Fig. 8c), and single monuments (Chigi - Fig. 8d) to large squares (Trento - Fig. 8e).

The reconstruction, editing and visualisation algorithm embedded in 3Dnow showed evidence to handle the tested cases, even when images characterised by different scales or multiple cameras (e.g. terrestrial and UAV images for the Neptune Paestum dataset Fig. 10).

Noisy points and unwanted artifacts, such as the sub-clusters of points separated from the main “body” of Sarcophagus, can be eliminated through the automatic editing procedure embedded in 3Dnow (Fig. 9). These noisy sparsely distributed points can be further pruned via the selection of an appropriate “trim” value if meshing is performed.

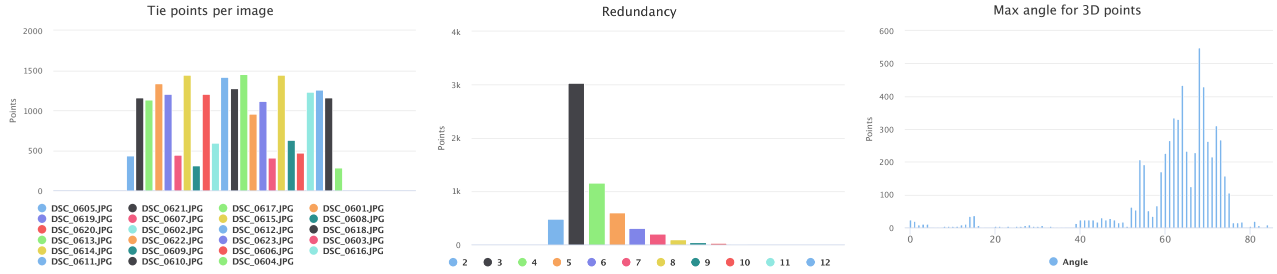
5.2 Computational performance

We performed the dense reconstruction on two different server configurations in order to test the performances using the multi-GPU architectures. The dataset we used for this experiment contained 55 images of size 5472×3648 pixels. In Tab. 2 we can observe

Table 1. 3D reconstruction use-cases. Key: TI(OI): Total Images(Oriented Images); GSD: Ground Sample Distance.

Dataset	Camera type	TI(OI)	Image res. [pixel]	GSD [cm]	Examples
Sarcophagus	Nikon D3X	516(516)	6048×4032	0.01	
Neptune Paestum (UAV+Terrestrial)	NikonD3X Canon550D	680+111 (680+111)	5184×3456 6048×4032	0.3 5	
Buddha Thailand	Nikon D3X	40(40)	6048×4032	0.5	
Trento Duomo	Nikon D3X	369(369)	6000×4000	0.5	
Palazzo Chigi	Canon G10 PowerShot	26(26)	4416×3312	1	
Saranta Kolones	Huawei P9	210(210)	1920×1080	3	
Dortmund (aerial nadir/oblique)	IGI Pentacam	59(59)	8176×6132	8-12	

https://doi.org/10.5194/isprs-archives-XLII-2-1097-2018 |



(a) (b) (c)

Figure 7. Statistics extracted from an example of 3D reconstruction and available to view in 3Dnow.

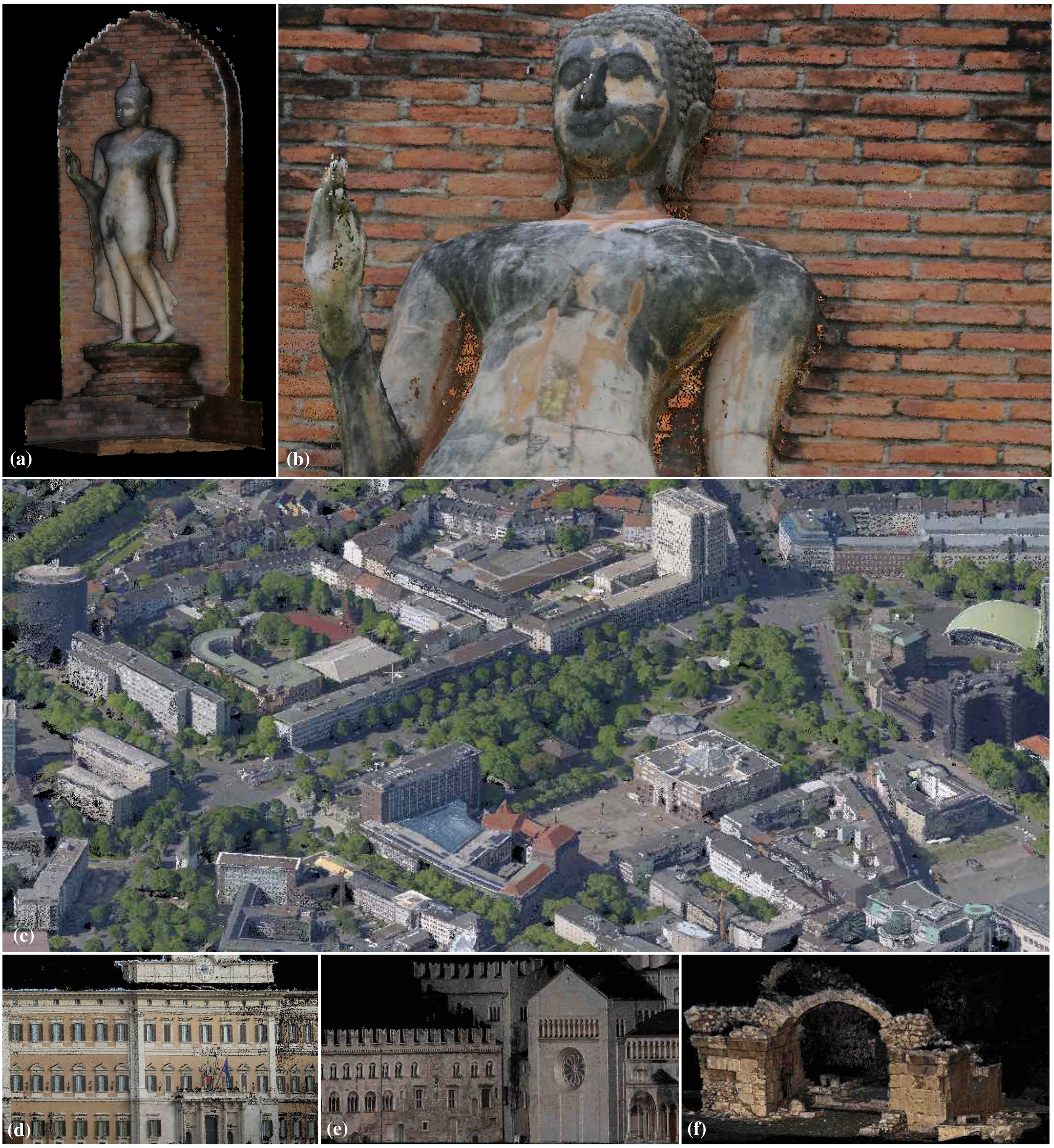


Figure 8. Dense point cloud examples from Tab. 1 datasets: (a,b) Buddha Thailand, (c) Dortmund, (d) Chigi, (e) Trento and (f) Saranta Kolones.

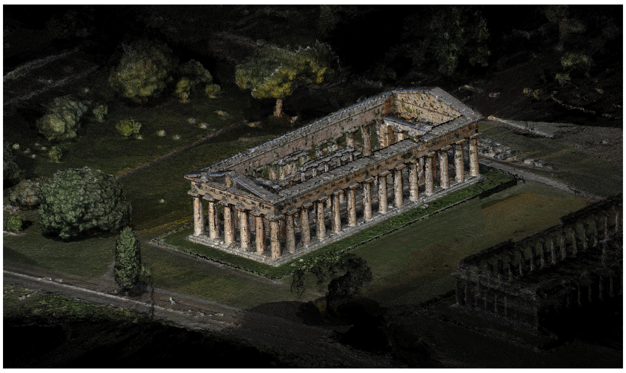


Figure 10. 3D survey of the Neptune temple in Paestum combining UAV and terrestrial images.

that the computational time decreases dramatically as the number of GPUs increases. The steps that benefit the most from the use of GPUs are feature matching and dense reconstruction, and this is because the operations performed during these steps are mostly linear operations. This suggests that for this type of processing, architectures for parallel processing can be more effective also to handle concurrent sessions from multiple accounts.

6. CONCLUSIONS

The paper presented the developments of a new web server for the 3D reconstruction of various scenarios from set of images. The server offers high performance processing capabilities for the orientation of any type of image block and the generation of dense point clouds. With respect to existing solutions, the main features of 3Dnow include: user friendly GUI, selection of computation parameters, 3D visualisation of generated point clouds and meshes, filtering and cleaning of 3D data, statistics about processing steps. The reported examples span from small to large datasets, from smartphone cameras to professional SLR and aerial cameras, from urban to heritage and architectural scenarios.

As future work, we are exploring mechanisms to predict (and suggest) processing parameters in order to generate more precise results with the automatic reconstruction process. Moreover, we will extend the current functionalities of 3Dnow with tools for the comparison of the generated 3D models with available ground truth data, as well as for the semantic segmentation of point clouds using machine learning approaches based on deep learning.

Lastly, we will include also educational material and tutorials related to 3D reconstructions from images. Indeed a proper understanding of the theoretical background of algorithms running in software applications is mandatory in order to obtain reliable 3D results and metric products. Leaving the black-box processing approach behind will ensure a better usability of achieved 3D results, long-lasting quality data, transferability of the methodology and a better diffusion of image-based 3D reconstruction methods.

Table 2. Computational time using different configurations for parallel computing.

	Number of GPUs		
	1	**2**	**3**
NVIDIA GTX 1080	132m47s	63m20s	33m54s
NVIDIA GTX 1080Ti	98m55s	54m32s	40m21s

ACKNOWLEDGEMENTS

This research has partially received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement number 687757.

REFERENCES

Abate, D., Furini, G., Migliori, S. and Pierattini, S., 2011. Project Photofly: new 3D modeling online web service (case studies and assessments). *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* XXXVIII-5/W16, pp. 391–396.

Allauth, 2018. http://django-allauth.readthedocs.io/en/latest/index.html. Accessed: Apr 2018.

ARC3DWeb, 2018. https://homes.esat.kuleuven.be/~visit3d/webservice/v2/. Accessed: Apr 2018.

Autodesk, 2018. https://www.autodesk.com/products/recap/overview. Accessed: Apr 2018.

Bootstrap, 2018. https://getbootstrap.com/. Accessed: Apr 2018.

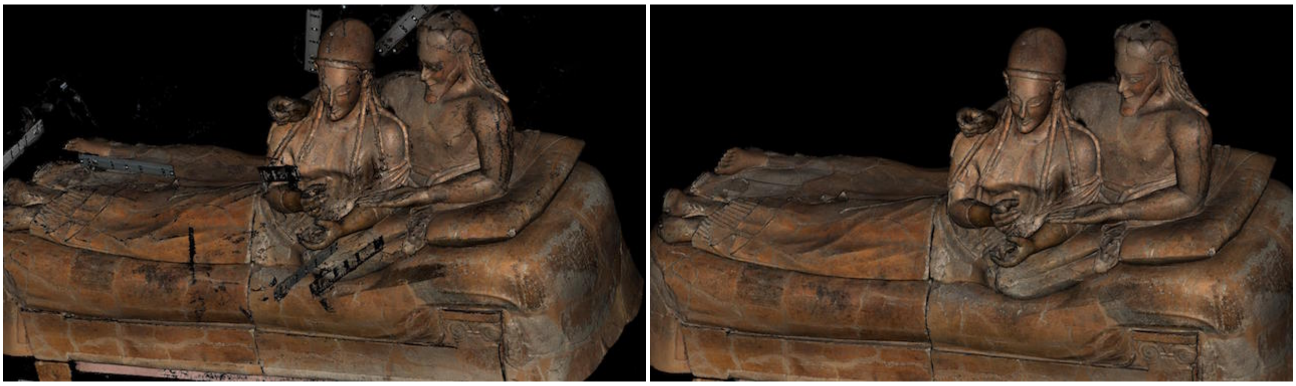
Brutto, M. L. and Meli, P., 2012. Computer vision tools for 3D modelling in archaeology. *International Journal of Heritage in the Digital Era,* 1, pp. 1–6.

Celery, 2018. http://www.celeryproject.org/. Accessed: Apr 2018.

COLMAP Web, 2018. https://colmap.github.io/tutorial.html#structure-from-motion. Accessed: Apr 2018.

Galizia, M. and Santagati, C., 2013. Low cost image-based modeling techniques for archaeological heritage digitalization: more than just a good tool for 3D visualization? *Virtual Archaeology Review,* 4(9), pp. 123–129.

Garland, M. and Heckbert, P., 1997. Surface simplification using quadric error metrics. In: *ACM Proc. of Computer Graphics and Interactive Techniques*, Los Angeles, US.



(a)

(b)

Figure 9. Sarcofagus dense point cloud (a) before and (b) after the automatic 3D point cleaning.

Hartmann, W., Havlena, M. and Schindler, K., 2016. Recent developments in large-scale tie-point matching. *ISPRS Journal of Photogrammetry and Remote Sensing,* 115, pp. 47–62.

Highcharts, 2018. https://www.highcharts.com/. Accessed: Apr 2018.

Kersten, T. and Lindstaedt, M., 2012. Potential of automatic 3D object reconstruction from multiple images for applications in architecture, cultural heritage and archaeology. *Int. Journal of Heritage in the Digital Era,* 1(3), pp. 399–420.

Knapitsch, A., Park, J., Zhou, Q.-Y. and Koltun, V., 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics,* 36(4), pp. 1–13.

Mauro, M., Riemenschneider, H., Signoroni, A., Leonardi, R. and Gool, L. V., 2014. A unified framework for content-aware view selection and planning through view importance. In: *British Machine Vision Conference*, Nottingham, UK.

Menna, F. and Troisi, S., 2010. Low cost reverse engineering techniques for 3D modelling of propellers. *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* 38, pp. 452–457.

Model-View-Template, 2018. ttp://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm. Accessed: Apr 2018.

Nocerino, E., Poiesi, F., Locher, A., Tefera, Y., Remondino, F., Chippendale, P. and Gool, L. V., 2017. 3D Reconstruction with a collaborative approach based on smartphones and a cloud-based server. *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* XLII-2/W8, pp. 187–194.

PHOV, 2018. http://www.phov.eu/about-phov/. Accessed: Apr 2018.

Potree, 2018. http://www.potree.org/. Accessed: Jan 2018.

PrecisionMapper, 2018. http://www.precisionhawk.com/precisionmapper. Accessed: Apr 2018.

Remondino, F. and Menna, F., 2008. Image-based surface measurement for close-range heritage documentation. *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* 37(B5-1), pp. 199–206.

Remondino, F., Nocerino, E., Toschi, I. and Menna, F., 2017. A critical review of automated photogrammetric processing of large datasets. *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* XLII-2/W5, pp. 591–599.

Schoning, J. and Heidemann, G., 2015. Evaluation of Multi-view 3D Reconstruction Software. In: *Proc. of Computer Analysis of Images and Patterns*, Valletta, MT.

Skycatch, 2018. https://www.skycatch.com/. Accessed: Apr 2018.

Vergauwen, M. and Gool, L. V., 2006. Web-based 3D reconstruction service. *Machine vision and applications* 17(6), pp. 411–426.

WebGL, 2018. https://www.khronos.org/webgl/. Accessed: Apr 2018.

Wikitude, 2018. http://studio.wikitude.com/. Accessed: Apr 2018.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-1097-2018 | © Authors 2018. CC BY 4.0 License.