Integration of Depth Maps from Arcore to Process Point Clouds in Real Time on a Smartphone

Raphaël Haenel, Quentin Semler, Edouard Semin, Pierre Grussenmeyer, Emmanuel Alby

INTEGRATION OF DEPTH MAPS FROM ARCORE TO PROCESS POINT CLOUDS IN REAL TIME ON A SMARTPHONE

R. Haenel [1,2] *, Q. Semler [1], E. Semin [1], P. Grussenmeyer [2], E. Alby [2]

[1] Syslor SAS, 1 Square Camoufle, 57 000 Metz, France – (raphael.haenel, quentin.semler, edouard.semin) @ syslor.net
[2] Université de Strasbourg, CNRS, INSA Strasbourg, ICUBE Laboratory UMR 7357, Photogrammetry and Geomatics Group, 67000 France – (raphael.haenel, pierre.grussenmeyer, emmanuel.alby) @ insa-strasbourg.fr

Commission II, WG II/3

KEY WORDS: 3D scanning, smartphone, real-time, depth map, on-the-fly-reconstruction, ARCore.

ABSTRACT:

Real-world three-dimensional reconstruction is a project of long-standing interest in global computer vision. Many tools have emerged these past years to accurately perceive the surrounding world either through active sensors or through passive algorithmic methods. With the advent and popularization of augmented reality on smartphones, new visualization issues have emerged concerning the virtual experience. Especially a 3D model seems to be essential to provide more realistic AR effects including consistency of occlusion, shadow mapping or even collision between virtual objects and real environment. However, due to the huge computation of most of current approaches, most of these algorithms are working on a computer desktop or high-end smartphones. Indeed, the reconstruction scale is rapidly limited by the complexity of both computation and memory. Therefore, our study aims to find a relevant method to process real time reconstruction of close-range outdoor scenes such as cultural heritage or underground infrastructures in real time locally on a smartphone.

1 INTRODUCTION

Real-world three-dimensional reconstruction is a subject of long-standing interest in computer vision. It has become even more predominant with the advent and popularization of autonomous cars or augmented reality. Especially a 3D model seems to be essential to provide more realistic AR effects including consistency of occlusion, shadow mapping or even collision between virtual objects and real environment. This opened up the need for real time scanning at scale with continuous integration of the accumulated 3D data.

Initially, reconstructions were created by fusing depth measurements from specific active sensors such as structured light, time of flight (Izadi et al., 2011) or LiDAR (*Lightning Detection and Ranging*) into 3D models. Although these sensors provide accurate results, their expansive aspect and the need for adequate equipment make them less attractive. Therefore, multiple approach have emerged to reconstruct a scene based on monocular (Yang et al., 2011), binocular or multi-view stereo methods that predict depth maps according to RGB images only. A depth map represents an image where pixels does not contain colour information, but a distance to objects in the scene from a given viewpoint (Figure 1). This data describes a pseudo-reconstruction of space that does not directly represent a surface, but rather a sample of discrete values that allows understanding the geometry of the environment captured from a specific camera position.

Most of the existing solutions are relying on massive data exchange and cloud computing because the smartphone alone cannot provide sufficient computational performance. Therefore, the challenge of our research is to imagine a new approach of producing outdoor 3D models from a smartphone-based acquisition and a full processing performed on the device for the SYSLOR Company. We aim to reach a productivity in real-time in order to provide a continuous overview of the 3D reconstruction from the recording. The interest of this visualisation is to assist continuously the user during the survey by providing him security and confidence regarding both the quality of the acquisition and the exhaustiveness of the data produced.

The algorithmic procedure has thus induced a succession of constraints to be taken into account during the developments:

- Ensuring a computational reliability of the process efficiently supported by the performances offered by the current smartphones,
- Enabling optimization of the computational process to achieve real-time productivity,
- Ensuring accessibility of the acquisition for everyone.

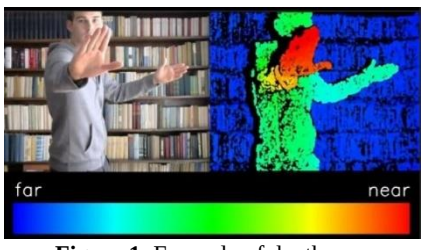


Figure 1. Example of depth map
Source: (Wojciech Mo, 2016)

2 STATE OF THE ART OF EXISTING SOLUTIONS

Before presenting how reconstruction can be processed, we need to take a closer look at scene representation. As depicted by (Seitz et al., 2006), it is possible to consider a differentiation of the

* Corresponding author

reconstruction approaches in terms of scene representation, since the geometry of an object can easily be described according to:

- A voxel grid (3D equivalent of pixel), used for its simplicity and its ability to approximate any surface,
- A polygonal mesh (usually triangular), very popular for simplicity of storage and convenience of data visualization,
- A depth map, which avoids the need to resample the 3D geometry.

Then, more generally, the reconstruction approaches can be subdivided into four ways according to how they are undertaken:

1. Extraction and mapping of feature points used to fit a surface to the generated features surface,
2. Calculation of a cost function associated with a 3D volume, then extraction of a surface from it,
3. Recurrent evolution of a surface to minimize a cost function,
4. Computation of a set of depth maps.

2.1 2.5 reconstruction – depth maps

In the absence of specific depth sensors, depth data can come from the study of a colorimetric similarity, also called photo-consistency criterion, between the pixels of a stereoscopic pair. A first well-known approach is the planesweep algorithm (Collins, 1996). The idea of this algorithm is to project backwards the whole image on successive virtual planes, perpendicular to the center of projection, indented according to a certain sampling step in 3D space. According to each virtual plane established, each pixel will be assigned a depth value derived from a similarity calculated between a reference view and the adjacent viewing position (Figure 2). Especially (Muratov et al., 2016) have reused and improved this method to generate a high-resolution mesh within minutes on a smartphone. The nature of the photo-consistency indicator determining the depth value has a significant impact on the relevance of the results in the case of changes in illumination or edge effects as indicated by (Mari Molas, 2017).

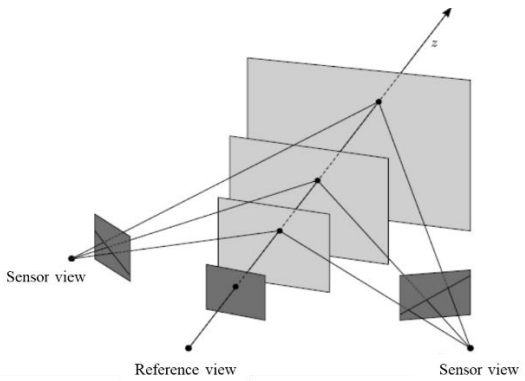


Figure 2. Planesweep method
Source: (Graber, 2012)

More recently, depth maps find a relevant use in augmented reality application, especially to offer management of occlusion effects between the real world and the virtual contents (Valentin et al., 2019). In the Depth API provided by ARCore, depth maps are generated from the hybrid optimization of two algorithms: PatchMatch (Bleyer et al., 2011) and HashMatch (Fanello et al., 2017). On the basis that an image is made up of regions of constant depth, the idea is to alternate generating an isolated depth value and propagating it to neighbouring pixels (Figure 3).

Finally, the popularization of machine learning has led to the emergence of new solutions for creating depth maps, notably by integrating convolutional neural networks (CNN) such as MVSNet (Yao et al., 2018) or DORN (Fu et al., 2018).

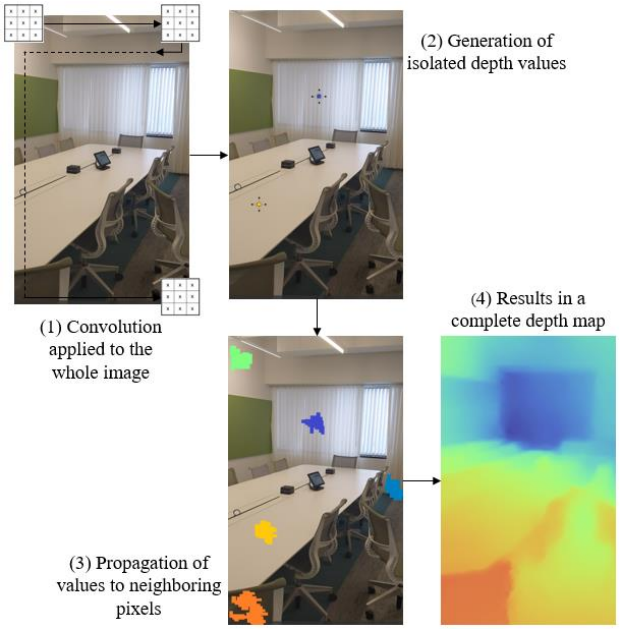


Figure 3. Process for creating depth maps within ARCore

2.2 Depth maps fusion

Usually, to fully reconstruct a 3D model, accumulated depth maps need to be gathered and combined. The integration of all these new data allows to continuously update the model and can be approached from different perspectives.

A first one is to represent the model as a surfel (*surface element*) cloud. Adopted by (Kolev et al., 2014), this representation seems more adequate for interactive applications running in real time in contrast to triangular meshes (Piazza et al., 2018). Indeed, the unstructured set of surfels can easily be kept consistent. A very interesting alternative to the one proposed above is based on Truncated Signed Distance Functions (TSDF). Inspired by (Zach, 2008), the idea is to convert the depth maps into a Signed Distance Field (SDF) where pixels are projected as voxel (3D point) with a value between [-1;1]. This value describes the distance of the voxel to the true surface of the object. This volumetric fusion of maps, taken up by (Izadi et al., 2011) in KinectFusion and (Graber, 2012), is interesting since it offers a relatively robust management of outliers while guaranteeing an efficient update of the model.

Then, the geometry of the model can be extracted using known methods such as raycasting (Graber, 2012) (Ondruska et al., 2015) or marching cubes (Lorensen and Cline, 1987). The raycasting approach is generally preferred over the marching cubes method because the latter relies on a binary decision to determine the surface position. However, as a result, the geometry of the 3D model may contain false positives, *i.e.* non-existent artefacts, as well as false negatives, *i.e.* poorly extracted

	Real time productivity	Computation optimization	Reconstruction quality	Reference
3D-R2N2	*	*	*	(Choy et al., 2016)
Depth API ARCore	***	***	***	(Valentin et al., 2019)
DORN	*	*	*	(Fu et al., 2018)
GEOMetrics	*	*	***	(Smith et al., 2019)
KinectFusion	***	**	*	(Izadi et al., 2011)
Marching cubes	**	**	***	(Lorensen and Cline, 1987)
Mesh R-CNN	*	*	***	(Gkioxari et al., 2019)
MobileFusion	*	*	***	(Ondruska et al., 2015)
MonoFusion	*	*	***	(Pradeep et al., 2013)
PatchMatch	***	**	*	(Bleyer et al., 2011)
Pixel2Mesh	*	*	***	(Wang et al., 2018)
Planesweep	***	***	**	(Collins, 1996) (Muratov et al., 2017)
MVSNet	*	*	***	(Yao et al., 2018)
Raycasting	***	***	**	(Graber, 2012)
RoutedFusion	**	*	**	(Weder et al., 2020)
SurfelMeshing	*	**	***	(Schöps et al., 2019)
Delaunay Tétraédization	*	**	*	(Piazza et al., 2018)
TGV	**	**	**	(Graber, 2012)
Visual hull constraints	***	***	**	(Prisacariu et al., 2015)

* : Poor consistency, ** : Acceptable consistency, *** : particular adequacy to the criteria

Table 1. Conclusion of our study concerning reconstruction methods

features. In addition, the marching cubes method is more favourable because it offers directly a mesh model against a point cloud for raycasting.

2.3 3D volume creation

Rather than going through an intermediate representation of the data, it is possible to build a volumetric geometry directly. From CNNs constructed as graphs, Pixel2Mesh (Wang et al., 2018) with and GEOMetrics (Smith et al., 2019) are extracting spatial deformations that will shape a predefined mesh model to converge to the captured geometry. Pixel2Mesh uses dissociation layers to refine the facets of the mesh by uniform oversampling. In GeoMetrics, the mesh is redefined only in target areas, which allows a better adaptation to the local complexity of an object. Nevertheless, these two methods are built from an initial mesh model, either a sphere or an ellipse, which directly induce topological biases.

To bypass this specific constraint, Gkioxari et al. (2019) developed Mesh R-CNN. The 3D mesh structure is obtained indirectly by passing through an intermediate description, namely a voxel occupation grid. This coarse description of the object will be transformed into a triangular mesh using the operation called cubify. Based on the voxel occupancy probabilities within the grid, each voxel is substituted by a triangular cubic mesh with 8 vertices, 18 edges and 12 faces. Then, the vertices and edges common to adjacent entities are merged while shared interior faces are eliminated.

2.4 Comparison of reconstruction methods

The specifications defined within our problematic enabled us to classify the various methods. Some of them have already been implemented on old generation smartphones (Ondruska et al., 2015; Muratov et al., 2016) or latest generation smartphone (Valentin et al., 2019). This observation confirms that 3D processing can be potentially embedded on this type of device. Although most of the presented methods based on deep learning represent the concrete future of real-time reconstruction for computer vision, they are still too limited for direct implementation on a smartphone.

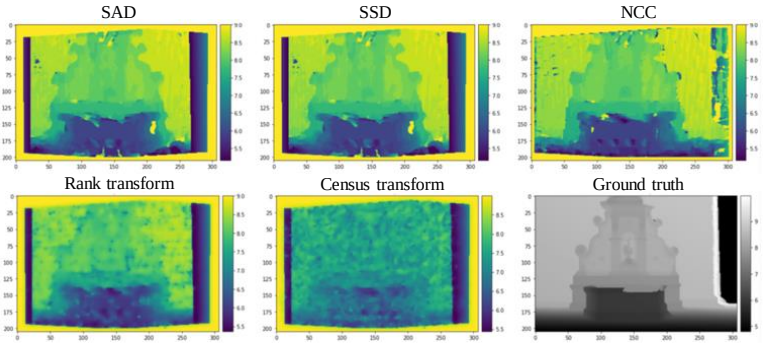
We have thus confronted the identified methods with our specifications (Table 1). The results provided are the outcome of tests undertaken, but for many an investigation carried out in relation to the remarks reported and correlated between the various authors cited. During the study, we have especially noticed that depth maps are particularly relevant data to bypass the triangulation steps related to collinearity equations while being an interesting trade-off between computation complexity and effective results. Especially, we highlighted the relevancy of the planesweep method implemented on smartphone by (Muratov et al., 2017) and the ARCore Depth API (Valentin et al, 2019) to generate accurate depth maps. Therefore, these methods seem to be very interesting for the development of our solution. Concerning data fusion, we prefer the method using surfels (Schöps et al., 2017), as it seems to be quite relevant for an interactive application while being within reach of our objectives from an algorithmic point of view.

3 PRODUCTION OF DEPTH DATA

3.1 Planesweep algorithm

The planesweep method presented by (Collins, 1996), operates locally to estimate the photo-consistency between two adjacent views. Its interest is to promote a result relatively quickly by ensuring robustness to incongruous changes in brightness, but also in the event of large bases. The indicator used to measure photo consistency directly influences the calculation of colorimetric homogeneity and therefore affects the overall quality of the depth maps.

We tested several indicators very well described by (Mari Molas, 2017) in order to determine the relevance of the method as a whole. Among those studied (SAD *Sum of Absolute Differences*, SSD *Sum of Squared Differences*, NCC *Normalized Cross Correlation*, Census transform, Rank transform), the one that seems to be the most relevant because of a favourable ratio accurate results / computation time is the SAD indicator (see Figure 4). Indeed, it allows describing the photographed environment more efficiently with a minimum error of 13.1 cm. This result was obtained after comparing computed depth map to



	SAD	SSD	NCC	Rank transform	Census transform
RMS [cm]	13.1	13.1	13.8	15.1	66.9
Time computation [min]	3.19	3.29	7.26	5.37	10.12

Figure 4. Impact study of photo-consistency measurements on depth maps quality

a ground truth map obtained from a famous benchmark: *fountain-P11* (Strecha et al., 2008).

After testing the influence of the parameters composing the planesweep, it appears that this method is quite legitimate for generalized multi-view reconstruction applications. However, given that it is based on a purely sequential computational procedure, it results in a latency of the calculations. Therefore, we have discarded this solution and turned to a more localized approach of stereo-correspondence, which uses a simpler procedure to implement, namely the API integrated in Google's ARCore application.

3.2 Depth API ARCore

The development of mobile technologies, has allowed the emergence and democratization of augmented reality applications. Google services have especially developed an API integrating the measurements of depth maps generated from a monocular camera (Valentin et al., 2019) at a rate of 30 Hz. In order to understand how these depth maps are generated, it is necessary to look at the overall working principle of ARCore. This application is based on three fundamental concepts: motion tracking, environment understanding and light estimation. Using an approach similar to conventional SLAM, feature points are detected along the movement to locate the smartphone in space. Using a methodology similar to RANSAC, these points are used to determine average planes ensuring an accurate understanding of the surrounding environment.

To use the Depth API from ARCore wisely, it is important to take into account the following factors that can influence the quality of a recording:

- Illumination conditions,
- Potential phenomena of reflection or scattering specific to the materiel,
- Poor textured scenes.

Figure 5 shows the analysis of the data produced by ARCore. The first case was performed in a relatively poor textured environment. The differences in depth observed are of the order of ten centimeters. This is interesting but it highlights the limitations of the solution, which relies solely on visual colorimetric information to calculate the different distances. The second case represents a more realistic scene with different depths. We can see right away that the results are better than the first case. In particular, the difference in depth is estimated to be about 8 cm. It is important to understand that these results depend on the quality of the movement performed, so a redundancy of views will provide more accurate results than a single view as showed in Figure 5.

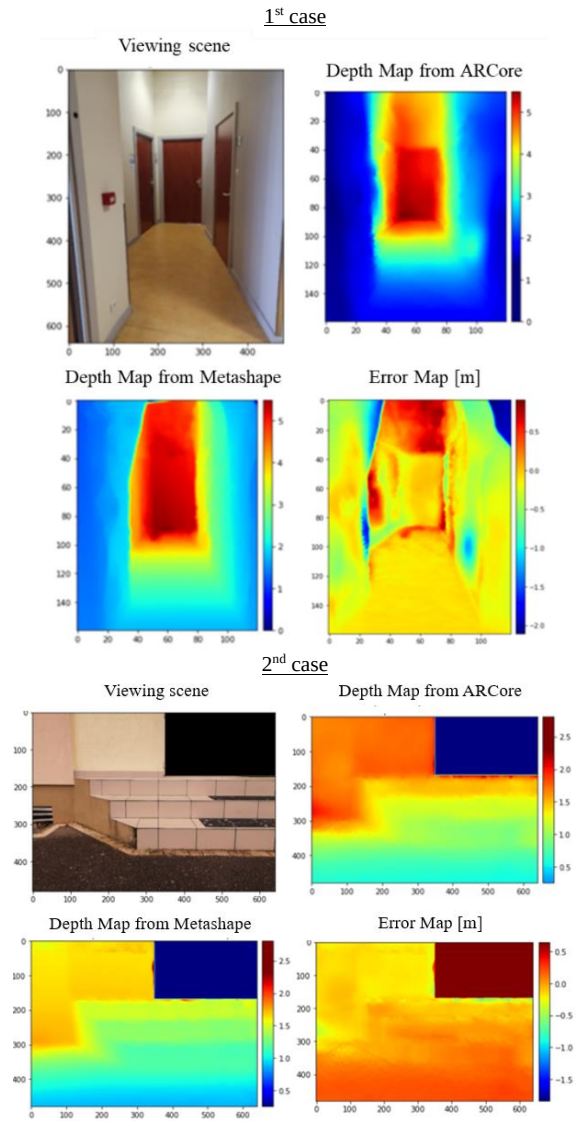


Figure 5. Two cases of analysis of depth results provided by ARCore

ARCore applies a smoothing process to its depth maps in order to express a value for all the pixels of an image. This means that the depth gradient is quite small; implying that around the edges of the objects discontinuities will appear causing a slight difference in accuracy within the comparison. To get rid of these distorted values, we set up a post-processing of the data consisting in the elaboration of a discontinuity mask based on a Canny detection (Canny, 1986) followed by mathematical morphology operators to remove a maximum of uninteresting pixels. The size of the kernel is chosen so that only the dominant contours are kept in the image.

Finally, the API provided by ARCore meets our expectations more favourably in terms of both the quality of the results and the rate of data production. This method has therefore been selected in our developments.

4 3D RECONSTRUCTION METHODOLOGY

ARCore can been used in the reconstruction process. Nevertheless, (Manni et al., 2021) decided to bypass the classical reconstruction process by using an object classification method to determine the most similar synthetic object from a database.

The contribution of depth maps alongside image data, resulting in so-called RGB-D data, opens the way to the creation of 3D models. Thanks to all the research carried out, we were able to design an optimal processing chain that converts RGB-D data to point clouds based on the intrinsic parameters of the camera. Figure 6 illustrates some reconstructed point clouds created with our method.

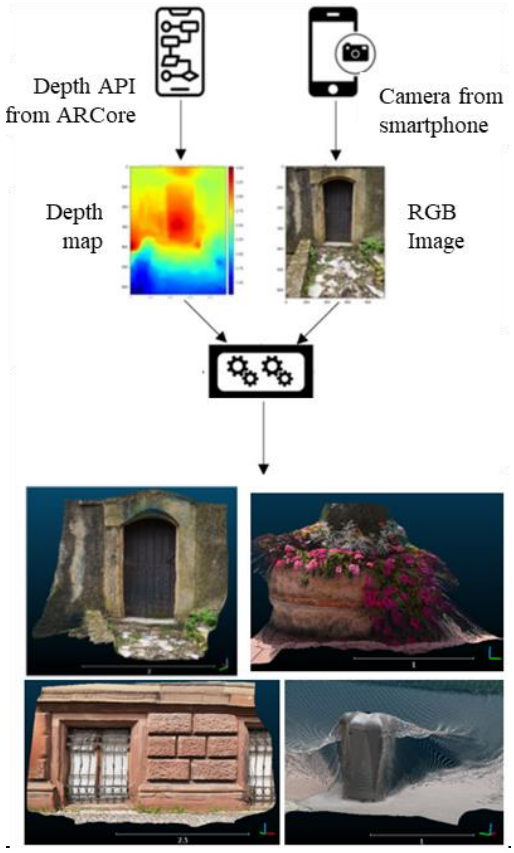


Figure 6. Example of reconstructions provided by our solution

4.1 Characteristics of point clouds

The characteristics of the produced point clouds are directly influenced by the resolution of the RGB-D images. Initially, the resolution of the depth maps produced by ARCore is 160x120 pixels to ensure a high productivity. Given that each pixel in the image produces a single 3D point, this will lead to a sparse point cloud of 19200 points with a spacing of about 2 cm (Figure 7a). Therefore, we have increased the size of depth maps to 320x240 and 640x480 pixels using linear interpolation to oversample the data. This operation intends to increase the amount of data from 19200 to 76800 and 307200 points respectively while providing a better density of points with a spacing reduced to 1 cm and 0.5 cm respectively. Even if more data to be processed implies more calculations to be carried out, a simple parallelization of the projection will be enough to guarantee a real time productivity. Moreover, visually, a higher resolution makes sense when we wish to have a finer detail of the recorded scene (Figure 7b-7c).

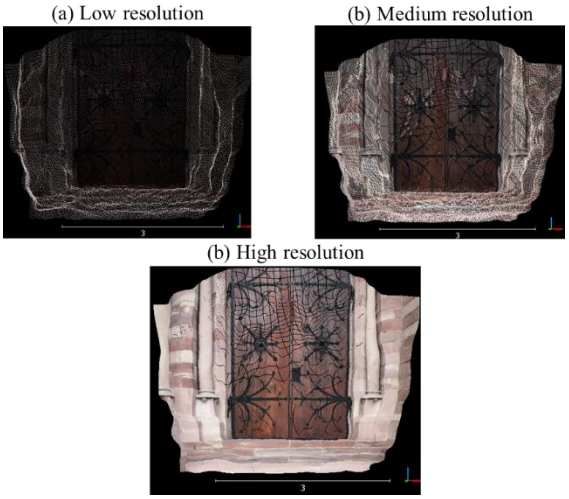


Figure 7. RGB-D resolution impact on point clouds

4.2 Analysis of results obtained

We built a proof of concept that can be generalized to several environments. Our solution has been designed to digitize all elements within a maximum range of 4 m. Actually, beyond a certain range, depth data provided by ARCore are going to be less accurate. Indeed, as one moves away from the scene or the object, the depth values will become more and more inaccurate. In particular, the ARCore documentation informs us that the data estimation error increases quadratically with the distance.

Therefore, to quantify the value of the models resulting from our method, we conducted comparisons with a ground truth built from Agisoft Metashape. The results are quite interesting and encouraging since we obtain an average deviation of 6-7 cm (Figure 8). The accuracy of our point clouds is directly linked to factors influencing depth estimation without forgetting the acquisition procedure. Indeed, to provide good estimation, a solid motion tracking at an adapted speed is required so that the algorithms can correctly understand the surrounding environment. Moreover, since the depth data results from colorimetric similarity between the pixels of a stereoscopic pair, deformations can be visible on the facade or the ground. For a first sketch of the result, the latter is quite promising, especially since the generation of the models is done almost in real time.

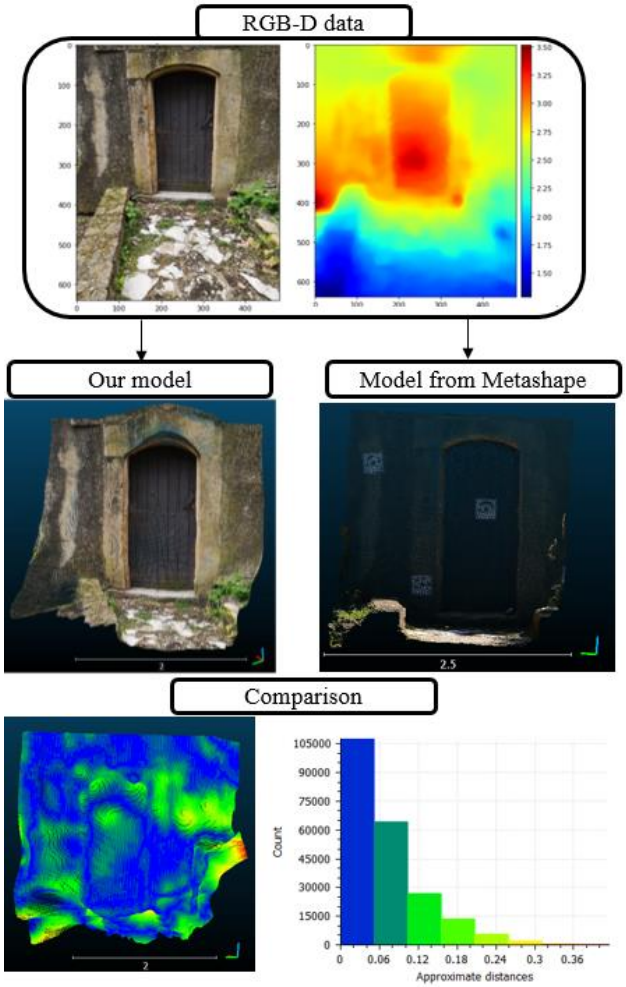


Figure 8. Qualitative analysis of a resulting point cloud

4.3 Computational bottleneck

Producing 3D model through cloud processing induces data exchange that can be time-consuming. Therefore, after conducting the acquisition, users have often to wait a certain amount of time to have a feedback. By relying on simple but effective methodology, we designed a proof of concept to ensure a permanent monitoring of the reconstruction with reasonable computation time.

At the time we present these results, our solution has not yet been fully integrated into the smartphone due to code conversion issues. Therefore, time consumptions are given according to computer computation using the minimum amount of CPU (*Central Processing Unit*) and by doing without the GPU (*Graphical Processing Unit*). In Table 2, we provide an overview of the average computation time of creating 3D model from a single view according to different input data resolutions. Results are very promising because we are able to produce very efficiently complex data consisting of several hundred thousand points all in a fraction of a second. In comparison, with a basic photogrammetry process, it can took several minutes to provide the corresponding model. Even if progress in automation has been made, allowing reducing the production time, the results of our method are still produced more quickly and requires less computing power.

During our tests, we also compared computational time while including post processing of the depth maps provided by ARCore (Table 2). As we already mentioned, these depth maps are smoothed to provide sufficient data all over the scene, but this induces flying points when we compute the corresponding 3D model (Figure 9).These particular points are outliers that damage the visual representation of the scene. The process described to remove these points seems to lead to an increase in calculation time of about 20%. Therefore, further research have to be conducted to find some alternatives that suppress these outliers without increasing that much the computational time. However, in the case of more accurate depth data, we can easily imagine doing without this process.

	RGB-D data resolution		
	Low	Medium	High
w/o depth map post-processing	0.004s	0.02s	0.08s
w/ depth map post-processing	0.02s	0.09s	0.4s

Table 2. Computational time of our method for a single view

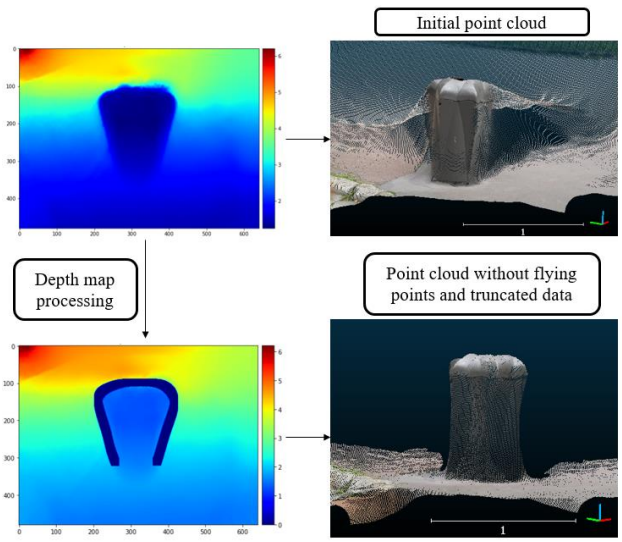


Figure 9. Flying points removal

5 CONCLUSION

Designing a proof of concept is not an easy thing to do, especially when significant material constraints are imposed. It implies having to find a computational approach simple enough while being as effective as possible compared to a reference method. For our study, we had three major constraints on which our developments had to rely:

- Ensuring a computational reliability of the process efficiently supported by the performances offered by the current smartphones,
- Enabling optimization of the computational process to achieve real-time productivity,
- Ensuring accessibility of the acquisition for everyone.

Thanks to all the researches and tests carried out, we ended up with a first processing chain, which works quite well. We are able to create dense point clouds within an accuracy of 7 cm for a single view, which can further be refined by the addition of more surrounding views. Even if the depth maps generated by ARCore can be noisy in the context of scattering surfaces or around edges,

we still managed to develop a very satisfactory method that provides quite encouraging results of about 8 cm. The initial objectives of the project led by the SYSLOR Company have been well outlined since we are able to provide results very quickly with limited computing power compared to usual method such as photogrammetry, which provide relevant information about the completeness and conformity of the acquisition. Although real time has not yet been concretely reached since the implementation of the algorithm has not been completely finished on smartphone, the provided initial results are very promising.

6 FUTURE WORKS

Our ambitions go beyond a simple ephemeral visualization as found in most AR applications integrating reconstruction methodologies. We want to establish a 3D model used to concretely map the environment around us, and more specifically underground infrastructures. For this, we wish to integrate extrinsic elements to our methodology, in particular GNSS data from a low cost antenna developed by (Quentin et al., 2019) within the company SYSLOR for permanent geolocalized restitution.

7 REFERENCE

Bleyer, M., Rhemann, C., and Rother, C., 2011: PatchMatch Stereo – Stereo Matching with Slanted Support Windows. In Procedings of the British Machine Vision Conference 2011, pages 14.1–14.11, Dundee. British Machine Vision Association. doi : 10.5244/C.25.14

Canny, J., 1986: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6): 679–698. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S., 2016: 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, Computer Vision – ECCV 2016, Lecture Notes in Computer Science, pages 628–644, Cham. Springer International Publishing.

Collins, R., 1996: A space-sweep approach to true multi-image matching. In Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 358–363, San Francisco, CA, USA. IEEE. doi: 10.1109/CVPR.1996.517097.

Fanello, S. R., Valentin, J., Kowdle, A., Rhemann, C., Tankovich, V., Ciliberto, C., Davidson, P., and Izadi, S., 2017: Low Compute and Fully Parallel Computer Vision with HashMatch. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 3894–3903, Venice. IEEE. doi: 10.1109/ICCV.2017.418

Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D., 2018: Deep Ordinal Regression Network for Monocular Depth Estimation. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2002–2011, Salt Lake City, UT. IEEE.

Gkioxari, G., Johnson, J., and Malik, J., 2019: Mesh R-CNN. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 9784–9794, Seoul, Korea (South), Korea (South). IEEE. ISSN: 2380-7504.

Graber, G., 2012: Real time 3D-Reconstruction. Master's thesis, Graz University of Technology - Institute for Computer Graphics and Vision, Austria.

Izadi, S., Davison, A., Fitzgibbon, A., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., and Freeman, D., 2011: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11, page 559, Santa Barbara, California, USA. ACM Press.

Kolev, K., Tanskanen, P., Speciale, P., and Pollefeys, M., 2014: Turning Mobile Phones into 3D Scanners. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 3946–3953, Columbus, OH, USA. IEEE.

Lorensen,W. E. and Cline, H. E., 1987: Marching cubes : A high resolution 3D surface construction algorithm. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques - *SIGGRAPH '87*, pages 163–169, Anaheim, California. ACM Press.

Manni, A., Oriti, D., Sanna, A., De Pace, F., Manuri, F., 2021: Snap2cad: 3D indoor environment reconstruction for AR/VR applications using a smartphone device. Computers & Graphics 100, 116–124. https://doi.org/10.1016/j.cag.2021.07.014

Mari Molas, R., 2017: Multi-view 3D Reconstruction via Depth Map Fusion for a Smartphone Application. BACHELOR THESIS UPF / YEAR 2017. Bachelor's Thesis, Pompeu Fabra University, Spain.

Muratov, O., Slynko, Y., Chernov, V., Lyubimtseva, M., Shamsuarov, A., and Bucha, V., 2016: 3DCapture: 3D Reconstruction for a Smartphone. In 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 893–900, Las Vegas, NV. IEEE.

Ondruska, P., Kohli, P., and Izadi, S., 2015: MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones. IEEE Transactions on Visualization and Computer Graphics, 21(11):1251–1258.

Piazza, E., Romanoni, A., and Matteucci, M., 2018: Real-Time CPU-Based Large-Scale Three-Dimensional Mesh Reconstruction. IEEE Robotics and Automation Letters, 3(3): 1584–1591. Conference Name: IEEE Robotics and Automation Letters.

Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M., and Bathiche, S., 2013: MonoFusion : Real-time 3D Reconstruction of Small Scenes with a Single Web Camera. In 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pages 83–88, Adelaide, SA, Australia. IEEE.

Prisacariu, V. A., Kahler, O., Murray, D. W., and Reid, I. D., 2015: Real-Time 3D Tracking and Reconstruction on Mobile Phones. IEEE Transactions on Visualization and Computer Graphics, 21(5): 557–570.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R., 2006: A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition -

Volume 1 (CVPR'06), pages 519–528, New York, NY, USA. IEEE. doi: 10.1109/CVPR.2006.19

Semler, Q., Mangin, L., Moussaoui, A., and Semin, E., 2019: Development of a low-cost centimetric GNSS Positioning solution for android applications. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, XLII-2/W17: 309–314.

Smith, E., Fujimoto, S., Romero, A., and Meger, D., 2019: GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects. In Chaudhuri, K. and Salakhutdinov, R., editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 5866–5876, Long Beach, California, USA. PMLR.

Strecha, C., von Hansen,W., Van Gool, L., Fua, P., and Thoennessen, U., 2008: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, Anchorage, AK, USA. IEEE.

Valentin, J., Kowdle, A., Barron, J. T., Wadhwa, N., Dzitsiuk, M., Schoenberg, M., Verma, V., Csaszar, A., Turner, E., Dryanovski, I., Afonso, J., Pascoal, J., Tsotsos, K., Leung, M., Schmidt, M., Guleryuz, O., Khamis, S., Tankovitch, V., Fanello, S., Izadi, S., and Rhemann, C, 2019: Depth from motion for smartphone AR. ACM Transactions on Graphics, 37(6): 1–19. doi: 10.1145/3272127.3275041

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G., 2018: Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In Proceedings of the European Conference on Computer Vision (ECCV), pages 52–67, Munich, Germany.

Weder, S., Schonberger, J., Pollefeys, M., and Oswald, M. R., 2020: Routed-Fusion: Learning Real-Time Depth Map Fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4887–4897, Seattle, WA, USA. IEEE.

Wojciech Mo, 2016: Stereo Vision - Depth Map. Available at: https://www.youtube.com/watch?v=bsA6RKUUA3M (Accessed: 31 March 2022).

Yang, X., Zhou, L., Jiang, H., Tang, Z., Wang, Y., Bao, H., and Zhang, G., 2020: Mobile3DRecon: Real-time Monocular 3D Reconstruction on a Mobile Phone. IEEE Transactions on Visualization and Computer Graphics, 26(12): 3446–3456.

Yao, Y., Luo, Z., Li, S., Fang, T., and Quan, L., 2018: MVSNet: Depth Inference for Unstructured Multi-view Stereo. In Proceedings of the European Conference on Computer Vision (ECCV), pages 767–783, Munich, Germany.