Interactive Environment for Testing SfM Image Capture Configurations

Ivan Nikolov and Claus Madsen
Department of Architecture, Design and Media Technology, Aalborg University, Rendsburggade 14, Aalborg, Denmark

Keywords: Structure from Motion (SfM), 3D Reconstruction, Evaluation, Unity.

Abstract: In recent years 3D reconstruction has become an important part of the manufacturing industry, product design, digital cultural heritage preservation, etc. Structure from Motion (SfM) is widely adopted, since it does not require specialized hardware and easily scales with the size of the scanned object. However, one of the drawbacks of SfM is the initial time and resource investment required for setting up a proper scanning environment and equipment, such as proper lighting and camera, number of images, the need of green screen, etc, as well as to determine if an object can be scanned successfully. This is why we propose a simple solution for approximating the whole capturing process. This way users can test fast and effortlessly different capturing setups. We introduce a visual indicator on how much of the scanned object is captured with each image in our environment, giving users a better idea of how many images would be needed. We compare the 3D reconstruction created from images from our solution, with ones created from rendered images using Autodesk Maya and V-Ray. We demonstrate that we provide comparable reconstruction accuracy at a fraction of the time.

1 INTRODUCTION

Capturing 3D models of objects has become an important part of the entertainment (Statham, 2018), medical (Péntek et al., 2018) and manufacturing industries (Galantucci et al., 2018). Having not only 2D representations of the objects through images, but a whole 3D model can give more information about the object's appearance, form and scale. When a high level of accuracy is needed in the captured 3D model, the go to technology has been laser scanners (Pritchard et al., 2017) and structured light scanners (Eiríksson et al., 2016), as well as structure from motion (Özyeşil et al., 2017). This paper focuses on SfM.

SfM works by first taking images all around the desired object, covering its whole surface. Features points are extracted from each image and matched between images. By triangulating these matched 2D points on the images, the 3D world coordinates of each camera position, as well as a sparse point cloud of the scanned object can be calculated. The camera positions and the sparse point cloud are then adjusted and interpolated to create a much denser point cloud, which captures a lot of the details of the object. Currently there exist multiple commercial (Bentley, 2016), (Agisoft, 2010)and open-source (Schönberger and Frahm, 2016), (Sweeney et al., 2015) solutions for SfM reconstruction.

Here comes one of the biggest drawbacks of SfM - the reliance on the quality of the input images. If problems like lack of enough images, blurriness, over/underexposure or noise are present in the input images, they will result in lower quality or complete failure of the reconstruction. Further problems can arise if the captured object has a specular surface, transparent parts or lacks a detailed surface. Testing different configurations of the capturing environment, camera settings, capturing conditions and objects can take a lot of time and can easily become costly if equipment needs to be changed or if the captured object needs to be processed to make its surface more diffuse. Additionally, different SfM solutions have varying degrees of robustness to these problems, making it crucial to know what is the best setup for the task at hand. A lot of research (Schöning and Heidemann, 2015), (Knapitsch et al., 2017) has gone into looking into how all these factors contribute to the output of SfM.

2 OUR PROPOSED SOLUTION

The normal way to test out different capturing conditions and setups is by rendering out images from a 3D model in programs such as Autodesk Maya (Autodesk, 1998). This way the user can be in control and change lighting conditions, camera positions, change

Nikolov, I. and Madsen, C.
Interactive Environment for Testing SfM Image Capture Configurations.
DOI: 10.5220/0007566703170322
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 317-322
ISBN: 978-989-758-354-4

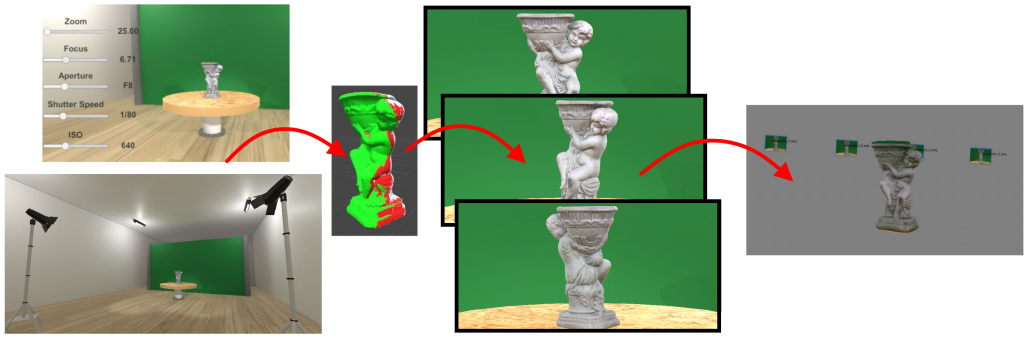


Figure 1: Overview of our proposed solution. A Unity interactive testing environment, an approximated DSLR camera with changeable settings and a visualization of how much of the object's surface has been scanned. The captured images can then be used for SfM reconstruction testing purposes.

the environment, etc. This can produce photorealistic results, but has the shortcoming that it requires in-depth knowledge of the used software solution, there is no easy way to observe the capturing progress. Rendering each image, also takes a long time, making the whole process cumbersome.

This is why we introduce a solution, which aims to address all the above shortcomings and deliver results which are comparable. We propose the creation of a testing environment in Unity, which can approximate the physical properties of a real world capturing setup for taking synthetic images. This environment can be used for initial testing with different setup variations, camera settings, objects, number of images, so a deeper insight can be achieved in the problem, before spending too much time and resources. In addition we propose a intuitive visualization of how much of the scanned object is captured with each image, as well as how much of an overlap there is between the images. An overview image of our proposed solution can be seen in Figure 1.

We compare our solution's results to both reconstruction results from real life images, as well as to results from synthetic images produced with Maya and V-Ray (Group, 1997), by using a ground truth created with a structured white light scanner. We demonstrate that our method produces comparable results to the offline rendering approach in a fraction of the time and captures the overall shape and detail present in the real life image reconstruction.

In addition we give some use cases for SfM capturing, where our proposed solution can come in handy and introduce some quality of life functions, which will make the normally tedious and long process easier.

3 METHODOLOGY

To create the testing environment each of the parts of a capturing setup needs to be modeled - the camera, the environment and the scanned object. The implementation of each of these is explained in detail in the subsections below. As DSLR cameras are the most widely used type of cameras for SfM 3D reconstruction, the testing environment's cameras are model after the typical DSLR parameters.

3.1 Camera Approximation

Our proposed solution does not aim to simulate how the physics of a real camera work. As mentioned before this has been implemented to a much greater extend in V-Ray for Autodesk Maya and will be extremely challenging to implement in Unity and even more to make it work in real time. This is why we choose to simply model how the different parameters of the camera can change the output image in appearance. The image itself is a "screenshot" of what a Unity camera renders of the environment and the parameters of the camera change this "screenshot" by introducing standard Unity shader effects, to mimic the real world changes.

A number of camera parameters and functions are modeled for approximating the results from changing them on the final image - aperture, shutter speed and ISO, as well as focal length and depth of field. The design consideration while implementing each are given in the sections below.

3.1.1 Focal Length and Depth of Field

To approximate the change of the field of view and zoom level, when adjusting the focal length, Equation 1 is used. In the equation h is the sensor height and f is the current focal length. The modeled camera's

sensor size is given as an input and can be changed depending on the modeled camera and is given in *mm*. The calculated field of view can be clamped to specific values to suit the needs of the testing scenario and to best approximate the effect of using a specific lens with the camera.

$$fov = 2arctan(\frac{h}{2f}) \quad (1)$$

To mimic more closely an output image from a real camera, the radial barrel and pincushion distortions are also implemented. Both distortions are implemented by modifing the fisheye standard asset shader. As an extension of this, future work is planned to use camera calibration algorithms on a number of cameras and lenses to estimate and better model the intrinsic camera parameters and distortions.

The changing of the depth of field when focusing is done using the depth of field shader that comes with the standard assets. Changing the focus of the approximated camera, changes the calculated shader distance from the camera to the Unity environment, which in turn determines the far plane, beyond which the shader applies a disc shaped blur filter.

3.1.2 Aperture, Shutter Speed, ISO

Each of the camera settings, change the intensity of the effects that they introduce to the final rendered image. The steps are taken from (ISO12232, 2006), which are used in most of the state of the art DSLR cameras. The aperture is in the interval between $[f/1.2; f/64]$, the shutter speed is in $[20; 1/8000]$, while the ISO is in the interval $[100; 51200]$.

To properly approximate how each of them affects the final exposure the Additive system of Photographic EXposure (APEX) standard is taken as a starting point (Kerr, 2007). It treats each of the parameters as an additive system, in which the increase or decrease of one, results in doubling or halving the exposure. Equation 2 shows the relation between the exposure value EV, the shutter speed value T_v and aperture value A_v, and the ISO sensitivity S_v and brightness value B_v.

$$E_v = A_v + T_v = B_v + S_v \quad (2)$$

The aperture, shutter speed and ISO components are given in Equation 3. In the aperture equation, the square of the aperture is taken, as the whole area is needed. For the ISO the equation contains N, which is the constant that gives the relation between the arithmetic sensitivity value and the value used by the APEX standard, while S_x is the arithmetic sensitivity value. Each of the equations takes the base-2 logarithm to make the equation behave linearly.

$$A_v = log_2(A^2),\ T_v = log_2(1/T),\ S_v = log_2(NS_x) \quad (3)$$

Finally, the brightness value B_v, is simplified for the Unity approximation, as it is calculated as a sum of the intensity values of each light source in the Unity scene. This is done to approximate the illuminance of the scene. To approximate the effect of changing exposure, the exposure/brightness shader is used from the standard shader package, with its value calculated from the APEX equation.

In addition to changing the perceived scene exposure, each of the three parameters gives other effects. With lowering the aperture size, the blur disc size is made larger, allowing more of the scene to come into focus and vice versa with increasing the aperture size. In addition, with lowering the aperture value, a blur effect is added to the final render to simulate the possible lens diffraction problem that can arise when the size of the aperture becomes smaller (Born and Wolf, 2013).

Changing the shutter speed changes the amount of blur present in the final rendered image, if the object or the camera are in motion when the image is taken. This is modeled by introducing the motion blur effect from the standard assets.

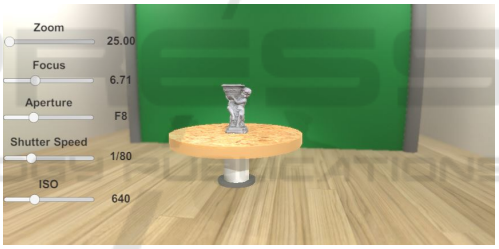


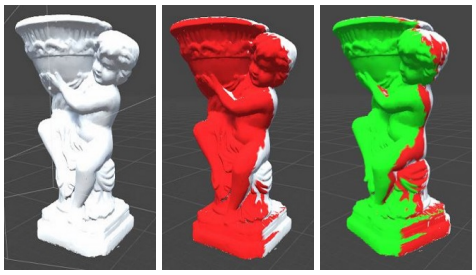
Figure 2: The camera parameter view. Each of the main camera parameters, plus the focal length and lens focus can be tweaked from this view, through the use of the sliders.

Changing the different camera parameters is done by switching to the designated parameter view and moving the specific sliders for each. Figure 2 shows the camera parameter view.

3.2 Environment Approximation

The most important parts of a SfM capturing setup for small scale objects are approximated - the lighting, the background and the way to capture different parts of the scanned object. Each of these parts is developed to be fast and easy to use.

The lighting is modeled as both an ambient lighting as well as directed lights. For ambient lighting a number of point lights are used all around the modeled studio. For a harder directed light, a number of directional lights can be setup. The number and position of both types of lights can be changed by the user



(a) Initial State (b) One image (c) Two images

Figure 3: The object coverage view. Initially (3(a)) the whole object's surface is white. After the first image, the seen surface is painted red 3(b). After the second image the parts that have been seen from two or more different camera views are colored green 3(c), indicating that there is overlap on the captured images.

as needed, as well as the intensity and warmth of the produced light. Soft shadows are rendered for all the objects in the room.

A turntable is implemented in the middle of the capturing room and the object for scanning is placed on it. Finally, a green screen is implemented for use whenever masking is necessary.

3.3 Object Approximation

The user can load the desired object into the environment, which is placed directly on the turntable. As the tested object only exists in the real world, there can be a number of solutions for substitutes used in the environment. A coarsely reconstructed version of the final object or a primitive object such as a sphere, cube, cylinder can be a substitute.

Each time a photograph image is rendered, the seen faces of the captured object from the camera are calculated using an matrix of raycasts from the camera. The object's material can be switched between normal textured view and a view of the seen faces. In the special view, initially the object is plane white. The faces that have been seen from one camera view are colored red, while faces that have been seen from more than one are colored green. Figure 3 shows the the object view and the coloring as more images are taken with enough overlap.

4 SOLUTION TEST AND RESULTS

We choose to compare a SfM reconstruction produced by images from our proposed testing environment against ones rendered using one of the most widely used ways to simulate a physical camera and an image taking setup - Autodesk Maya and V-Ray. In addition a reconstruction is done using real life DSLR camera images as a base case.

For the test real life object a stone angel statue is selected, which can be seen in Figure 4. The three reconstructed meshes need to be compared to a ground truth model. A high accuracy ground truth is produced using a white light scanner.

The next step is to create a real life image capturing setup. A Canon 6D DSLR camera is used. The camera is a full-frame camera with a sensor size of $35.8mmx23.9mm$. The photos are taken with the maximum possible resolution of the camera - $5472x3648$ pixels. The camera is positioned on a tripod in front of a turntable with the captured object. Two Elinchrom D-Lite RX4 lights are setup on both sides behind the camera and are targeted towards the object. A green screen is set behind the object. A photo is taken and the turntable is rotated each time 20 degrees, until the whole object has been captured in 360 degrees, which gives a total of 18 images.

The same capturing setup is created both with our proposed solution and in Maya. In Maya, the physical camera in V-Ray is used for simulating the Canon 6D with the camera parameters saved from the real life capture. The same parameters are used in out environment. The resultant images from the real life setup, the Maya and V-Ray setup and our proposed solution can be seen in Figure 4.

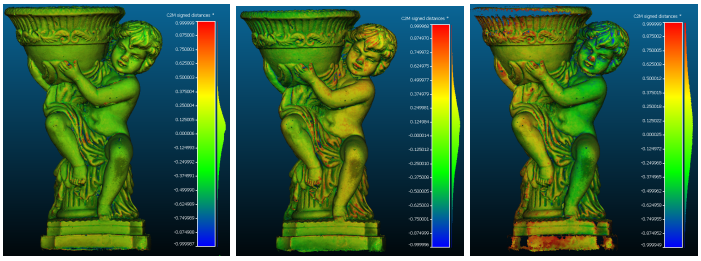
For each of the three sets of images, the reconstruction is done using Photoscan (Agisoft, 2010). The program is chosen as it is frequently used by researchers and provides robust and accurate results compared to other state of the art solutions (Schöning and Heidemann, 2015).

The three reconstructions are compared to the ground truth scan. The open source program CloudCompare (Girardeau-Montaut, 2003) is used for the comparison. Each of the reproduced meshes is scaled and registered to the ground truth object. The signed distances between the faces of the reconstructions and the ground truth are calculated. These distances are visualized as a heat map on Figure 5, where the blue color shows distances which are below the ground truth and the red color - distances above the ground truth, while green indicates that the two surfaces match. From these distances, the mean and standard deviation are calculated for each reconstruction. These are shown in Table 1.

The table shows that the difference between the reconstruction from the real DSLR images and the rendered V-Ray images is negligible, as expected. Only the standard deviation from rendered images is larger, mostly because the texture on the 3D model has lost

(a) Real DSLR (b) Maya and V-Ray (c) Our Solution

Figure 4: Images used for the reconstruction test. Figure 4(a), is the real life image taken from the 6D DSLR camera, 4(b) is the rendered image from Maya and V-Ray and 4(c) is the image from our proposed solution.



(a) Real DSLR (b) Maya and V-Ray (c) Our Solution

Figure 5: Heat map of the distances between the ground truth object and the reconstruction. Green indicates that the two coincide, while red and blue indicate larger positive and negative distances between the two.

Table 1: Mean in *mm* and standard deviation in *mm* of the distance metric for the three types of input data - images from a real DSLR Canon 6D camera, rendered images from Maya and V-Ray and our proposed solution.

Solution	Mean [*mm*]	Std. Dev. [*mm*]
Real DSLR	-0.38	2.41
Maya + V-Ray	-0.43	2.50
Our Solution	-0.49	3.51

some of its detail and has had some noise added. This is because the images from our solution lack the fidelity of the other images, as well as the smaller details that would come out from proper rendered lighting calculation. On the other hand, the Maya + V-Ray solution took almost 20 hours to render, making it less than ideal for fast testing and prototyping, compared to the 5 min it took to set the camera settings, find the proper camera position and take the images through our interactive environment.

5 USE CASES

Going from the work of (Nikolov and Madsen, 2016), where multiple SfM solutions have been tested under varying environmental and object conditions, it is apparent that a lot of time and work goes into creating a proper setup for a good 3D reconstruction. It is also seen that problems with the camera, the lighting or the capturing environment can drastically lower the quality of the produced model. This is why we introduce a lot of quality of life features in our proposed solution, which aim to make the capturing and testing process easier.

The camera can be made stationary with the turntable rotating a specified number of degrees, until the whole object has been scanned. The second way is by keeping the object stationary, both the standard First Person Shooter (FPS) controller or the flight controller can be attached to the approximated camera, so the user can move manually to the desired positions and rotations.

The implemented green screen can be toggled on and off and its color changed to better contract the color of the object being captured. The lighting can be moved and the intensity of the light can be regulated both for the directional and point lights, to test different possible illumination setups.

To make it easier for users to judge how much of the object's surface has been captured from each image an additional visualization mode is implemented. An important part of performing a successful SfM 3D reconstruction, is providing enough images, covering the whole surface of the object and having enough overlap between them.

6 CONCLUSION AND FUTURE WORK

In this paper we proposed a interactive testing environment for capturing images for SfM reconstruction. Our solution provides an approximation to the way images of a real life DSLR camera will look like and how the final image changes depending on the camera settings, focal length and focus. Together with the approximated camera we introduce a capturing environment, which can be interactively changed by the user, to accommodate different testing scenarios. Finally we added the possibility for the user to visualize how much of the scanned object's surface has been captured with each photo and is there overlap between different photos.

We tested our solution's output against an offline rendering output produced by Autodesk Maya and V-Ray and demonstrated that we achieve similar results at a fraction of the time.

For future work we would like to remake the interactive testing environment in another engine like Unreal, which has the possibility to use physical cameras and a better lighting model, as well model more of the DSLR intrinsic parameters.

ACKNOWLEDGEMENTS

This work is funded by the LER project no. EUDP 2015-I under the Danish national EUDP programme. This funding is gratefully acknowledged.

REFERENCES

Agisoft (2010). Agisoft: Photoscan. http://www.agisoft.com/. Accessed: 2018-11-10.

Autodesk (1998). Maya. https://www.autodesk.eu/products/maya/overview. Accessed: 2018-11-10.

Bentley (2016). Bentley: Contextcapture. https://www.bentley.com/. Accessed: 2018-11-10.

Born, M. and Wolf, E. (2013). *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier.

Eiríksson, E. R., Wilm, J., Pedersen, D. B., and Aanæs, H. (2016). Precision and accuracy parameters in structured light 3-d scanning. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40:7.

Galantucci, L. M., Guerra, M. G., and Lavecchia, F. (2018). Photogrammetry applied to small and micro scaled objects: A review. In *International Conference on the Industry 4.0 model for Advanced Manufacturing*, pages 57–77. Springer.

Girardeau-Montaut, D. (2003). Cloudcompare. http://www.cloudcompare.org/. Accessed: 2018-11-10.

Group, C. (1997). V-ray. https://www.chaosgroup.com/. Accessed: 2018-11-10.

ISO12232 (2006). Photography - digital still cameras - determination of exposure index, iso speed ratings, standard output sensitivity, and recommended exposure index. https://www.iso.org/standard/37777.html. Accessed: 2018-11-05.

Kerr, D. A. (2007). Apex-additive system of photographic exposure. *Issue*, 7(2007.08):04.

Knapitsch, A., Park, J., Zhou, Q.-Y., and Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78.

Nikolov, I. and Madsen, C. (2016). Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions. In *Euro-Mediterranean Conference*, pages 15–26. Springer.

Özyeşil, O., Voroninski, V., Basri, R., and Singer, A. (2017). A survey of structure from motion*. *Acta Numerica*, 26:305–364.

Péntek, Q., Hein, S., Miernik, A., and Reiterer, A. (2018). Image-based 3d surface approximation of the bladder using structure-from-motion for enhanced cystoscopy based on phantom data. *Biomedical Engineering/Biomedizinische Technik*, 63(4):461–466.

Pritchard, D., Sperner, J., Hoepner, S., and Tenschert, R. (2017). Terrestrial laser scanning for heritage conservation: the cologne cathedral documentation project. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Schönberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schöning, J. and Heidemann, G. (2015). Evaluation of multi-view 3d reconstruction software. In *International Conference on Computer Analysis of Images and Patterns*, pages 450–461. Springer.

Statham, N. (2018). Use of photogrammetry in video games: a historical overview. *Games and Culture*, page 1555412018786415.

Sweeney, C., Hollerer, T., and Turk, M. (2015). Theia: A fast and scalable structure-from-motion library. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 693–696. ACM.