ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN 3301
http://www.wscg.eu
WSCG 2023 Proceedings

Photogrammetry workflow for obtaining low-polygon 3D models using free software.

Ricardo Pardo Romero
Institute of New Imaging Technologies
Universitat Jaume I
Spain, Castello de la Plana
ripardo@uji.es

Inmaculada Remolar Quintana
Institute of New Imaging Technologies
Universitat Jaume I
Spain, Castello de la Plana
remolar@uji.es

ABSTRACT

This paper proposes a workflow for inexperienced designers to create low-poly 3D models using free software. It addresses the problem of the complexity generated by photogrammetry. The solution aims to enable independent developers to create realistic assets at cheap cost. It eliminates the need for experienced 3D artists or expensive commercial solutions.

Keywords

Photogrammetry, Low-poly objects, free-software, geometric simplification, realism

1 INTRODUCTION

Highly realistic 3D objects often have a high polygon count. Simplification methods require manual work or expensive tools, creating challenges for independent video game studios. Interactive technologies like extended reality are used in Industry 4.0 and education, but end-users lack expertise in adapting virtual environments. Photogrammetry [2] is an Image Based modelling (IBM) technique that uses photographs as the fundamental medium to extract accurate measurements and information about the physical properties of objects and their surroundings. It offers a cost-effective solution, allowing non-experts to scan real objects and obtain digital representations quickly. LiDAR, another remote sensing technology, produces precise 3D maps but is affected by weather conditions. According to [4], while both of these methods capture locations, photogrammetry requires less expertise to provides photorealistic results. The article proposes a solution for adapting photogrammetric objects using free software, benefiting independent studios and inexperienced creators.

Summarizing, the main contributions made in this paper are the following:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

- A workflow is given to adapt point clouds obtained by photogrammetry to assets suitable for video games, with high realism but low geometric complexity.
- A method is given to optimise the scanned geometric mesh, independently of the applied photogrammetry method, using only free software.
- A simple pipeline is offered to perform the whole process of complexity reduction without losing realism, so that anyone, even without experience as a 3D artist, can follow it.
- To evaluate the level of performance, visual quality and automation, the results obtained with the proposed pipeline will be compared with other methods.

The remainder of the paper is organised as follows. Section 2 briefly reviews important methods explored in recent years. Section 3 outlines the design of the presented pipeline. In Section 4, the results obtained in the work are compared and discussed. Finally, Section 5 presents the conclusions and future work.

2 RECENT SOLUTIONS

Photogrammetry is a popular and user-friendly Image-Based Modeling (IBM) method for creating 3D models [6]. It involves capturing images from different perspectives and using software to recreate the object. However, real-time visualization is challenging due to the high polygon count. Structure from motion (SFM) [5] is one of the most popular photogrammetry methods. Laser scanning, particularly LiDAR, provides accurate depth information and is suitable for larger objects and spaces. LiDAR is commonly used for mapping terrain and architectural purposes. While laser scanning is expensive, it is starting to be incorporated into mobile devices. Photogrammetry has gained traction in the gaming industry, reducing production costs and enhancing asset creation.Companies like. Smaller studios may face challenges due to limited resources, but free software and phone cameras can still yield impressive results. Despite the availability of free software, there is a lack of specific papers presenting a pipeline workflow.This article aims to fill this gap by providing steps to obtain accurate 3D representations of real-life objects using free software.

3 PROPOSED SOLUTION

The proposed solution aims to provide an easy workflow for obtaining a low-poly 3D model of a real object, specifically targeting inexperienced designers and indie game studios. The workflow utilizes free software to promote accessibility and affordability. The object chosen for this work is a complex piece of dry trunk, as shown in Figure 1, which exhibits non-homogeneous shapes and indentations.

Figure 1: Photograph of the real object to be digitally represented.

The workflow is summarized in Figure 2. It is primarily based on photogrammetry, starting with capturing pictures of the object using any device, in our case, the iPad Pro 2022. These images are then imported into *Meshroom* to generate a polygonal mesh. However, the resulting mesh is highly complex and needs simplification to be suitable for various applications. *Blender* and *Instant meshes* are used for this simplification process. To maintain realism in the simplified mesh, texture maps are created using *Blender*. Additional maps are generated using *Materialize* to enhance the realism and create a Physically Based Rendering (PBR) material.

3.1 Capturing the object

To ensure a faithful 3D model from photogrammetry, consider the following factors: Objects should be opaque, avoiding transparent or translucent materials [3]. Optimal lighting conditions are crucial for capturing accurate textures and proportions. Cloudy weather is recommended for outdoor photography, while flat lighting is preferred indoors. Capture a minimum of 20 photos from various angles around the object, with at least 50% overlap and neighboring images [7].In our experiment (Figure 3), 42 surrounding photos were taken. More photos improve the accuracy of the result.

3.2 Generating the point cloud and mesh

Once the images have been taken, it is time to generate the geometric mesh from them. For this, *MeshRoom* is the free option that has been selected for its simplicity. This 3D reconstruction software [1] is easy to use and allows the entire photogrammetric pipeline to be executed. Designers simply input the images obtained (seen in Figure 3) and the software generates a 3D model and textured mesh, using a node-based workflow.

At the end of this process, in the project folder *MeshRoom*, an .fbx 3D model will appear, with the geometric mesh information and the texture distribution unwrapped (Figure 5). The 3D model obtained usually has a large number of polygons. In this example there are 1,392,431 polygons, (Figure 4). This amount has to be reduced with the proposed pipeline to obtain the low-poly representation.

3.3 Optimizing the mesh

The next step involves decimating the geometry to obtain different levels of detail. Before this process, it is recommended to clean up unnecessary parts of the mesh captured during the photogrammetry process. *Blender* can be used to delete these parts. The modifiers in *Blender* are not suitable for this process due to their limitations. Instead, a third-party software called Instant Meshes is recommended for geometric simplification. It allows setting the desired number of polygons and predefined edge flow. In this case, the target polygon count is 6,892 seen in Figure 6.

Once the simplified mesh is obtained, a high-poly version of it is generated in *Blender* for texture baking, using the "Multi-Resolution" and the "ShrinkWrap" modifiers to project the vertices of the decimated mesh onto the SFM model's surface. Now we have two levels of detail.

Finally, *Blender*'s "Face orientation" option can be used to identify inverted normals and fix them, either flipping or recalculating them.

Mapping coordinates

The SFM mesh already have textures, but we want to put them on our new mesh. So new mapping coordinates need to be obtained for the simplified mesh. *Blender* offers the "UnWrap" modifier, and selecting

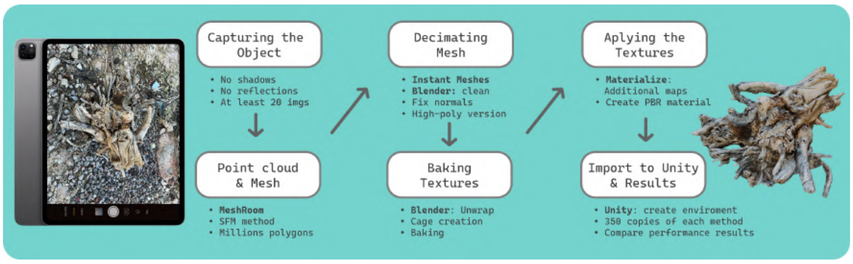

Figure 2: Overview of the presented work.

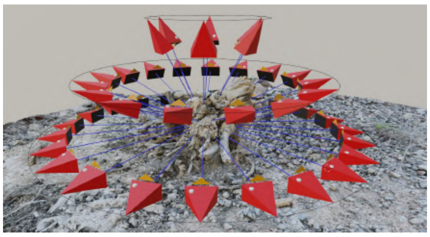

Figure 3: The cameras surrounding the object, represented by red pyramids.

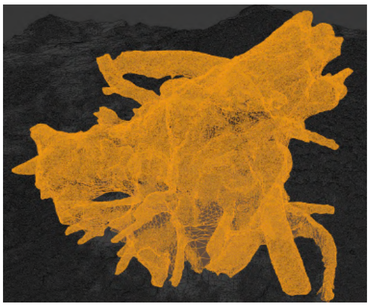

Figure 4: Geometric mesh extracted using the SFM method (*MeshRoom*). It is composed by 1,392,250 polygons.

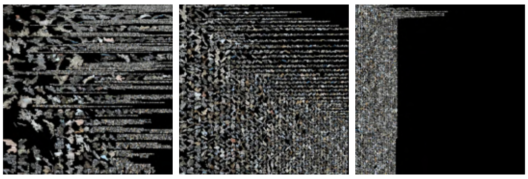

Figure 5: Unwrap distribution extracted using the SFM method (*MeshRoom*): 3 images are generated that contain all the unwrapped texture.

the "Smart UV projection" option can be a suitable solution on most cases, but depending on the mesh's complexity it is recommended to manually place marker seams to ensure accurate mapping. This process generates a 2D image where the texture of the simplified mesh is unwrapped.

Baking

To create texture maps for a 3D model, texture baking is a crucial process. It allows the recovery of lost details from the decimation process with minimal effort. Each baked map should be manually saved to avoid overwriting previous versions. To start, assign a default material

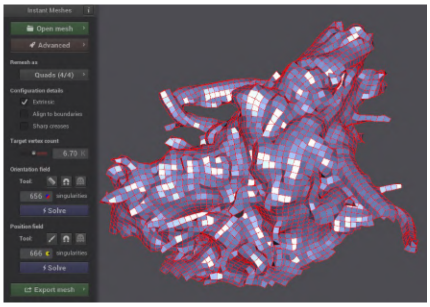

Figure 6: Configuration to obtain a decimated mesh in Instant Meshes.

with a 2D image map to the polygonal mesh. The content of this map is unimportant as it will be replaced. Configure the Render Properties to use the "Cycles" engine and access the "Bake" tab to begin the texture baking process. To extract the maps from a SFM 3D model in Blender, enable the "Selected to active" option and utilize the cage feature to determine texture projection points on the simplified mesh. Different texture maps are generated during the baking process. The *Diffuse* map contains color information, the *Normal* map analyzes high-poly mesh details, the *Ambient Occlusion* map adds depth through darkening areas with limited light exposure, and the *Height* map stores height information for vertex displacement. For enhanced realism, additional maps like *Roughness*, *Metallic*, and *Edge* can be generated using software like Materialize. To create a cage for baking and extracting textures, follow these steps. Duplicate the low-poly mesh and inflate it using the "Sculpt" section's "Inflate" tool. Ensure that the inflated cage completely covers the original mesh. Add the inflated cage in the "Bake" section and adjust the "Max Ray Distance" property for desired results. With the settings prepared, proceed with the baking process. For the *Diffuse* map, select only the "Color Contribution" and perform the bake. The *Height* map can only be generated using the "Multires" option and may not be applicable to all objects. Bake the remaining maps without changing any options. After baking in Blender, apply the generated maps to the textured object. The *Roughness* map obtained from Materialize will improve the visuals. However, the *Metallic* and *Edge* maps are unnecessary for this particular object.

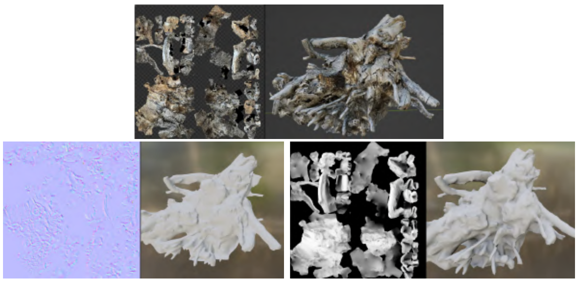

Figure 7: The *Diffuse*, *Normal* and *Ambient Occlusion* maps.

Once the maps are generated, link them to the applied material and evaluate the results. Additional adjustments can be made using Materialize to refine the *Roughness* map. For a better visual representation, consider adding the ground to the scene. This will enhance the overall presentation of the model.

Figure 8: Comparison between the original picture (left) and the render made in Blender 3D (right).

4 EXPERIMENTAL RESULTS AND DISCUSSION

Figure 9 shows a comparison of the visual results, highlighting the realism of each method. LiDAR does not provide good visual results for complex shapes. SFM representation has a very good visual quality. Our Low-Poly method has 99.5% fewer polygons than the SFM model, yet the visual quality remains comparable.

The LiDAR pipeline utilized an iPad app called Scanner 3D App, which produced a textured model. Considering the results, users may prefer one method over another based on their specific needs. LiDAR performs better with simpler objects and larger environments. SFM method has a lot of polygons but can be utilized to improve performance and maintain visual quality when combined with the presented pipeline.

5 CONCLUSION

With the pipeline proposed, obtaining photorealistic low-poly models is accessible to almost any developer. The contributions enable cost-effective creation of assets for interactive apps/games, with future plans to develop *Blender* plug-ins to automate part of the process.

Figure 9: LiDAR (First), Imaged Based SFM (Second), High-Poly (Third), Low-Poly (Fourth).

Acknowledgements

Research supported by the e-DIPLOMA, project number 101061424, funded by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Funded by
the European Union

6 REFERENCES

[1] Photogrammetric computer vision framework. https://alicevision.org/. Last accessed January 2023.

[2] J. S. Aber, I. Marzolff, and J. B. Ries. *Small-Format Aerial Photography*. Elsevier Science, 2010.

[3] Sébastien Lachambre, Sébastien Lagarde, Cyril Jover, et al. Photogrammetry workflow. *Rapport Technique, Unity*, 2017.

[4] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5), 2020.

[5] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] Nataska Statham. Use of photogrammetry in video games: A historical overview. *Games and Culture*, 15(3):289–307, 2020.

[7] Krzysztof Woloszyk, Pawel Michal Bielski, Yordan Garbatov, and Tomasz Mikulski. Photogrammetry image-based approach for imperfect structure modelling and fe analysis. *Ocean Engineering*, 223:108665, 2021.