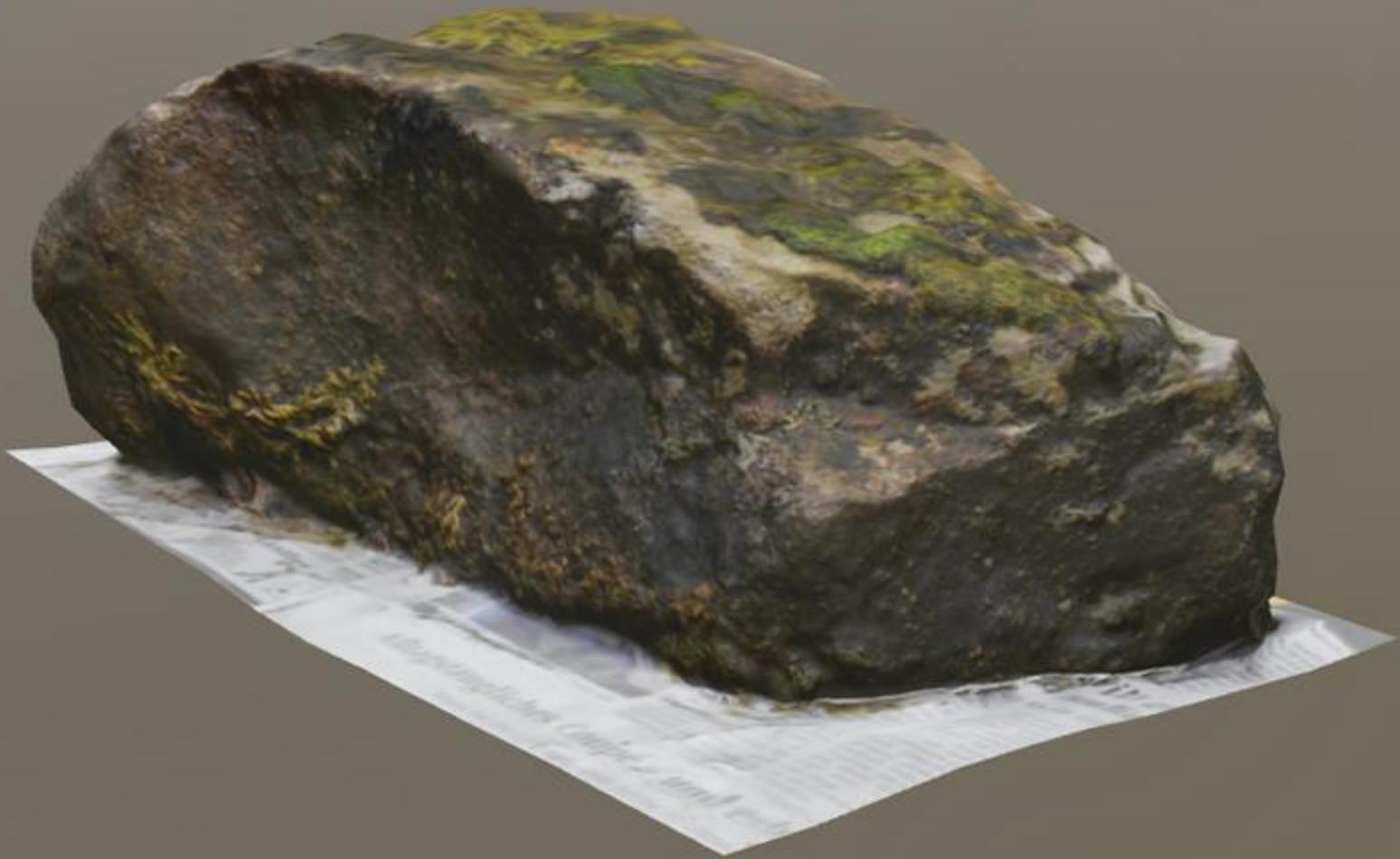


2016



## [PHOTOGRAMMETRY]

*What are beneficial use-cases for photogrammetry in a 3D game?*

*Martin Niehoff, 22.08.2016*

*For the Game Creation and Producing at the Saxion University of Applied Science*

# Photogrammetry

*What are beneficial use-cases for photogrammetry in a 3D game?*

Written by

**Martin Niehoff**

**info@martin-niehoff.de**

As specialization topic in the Game Creation and Producing Course at the  
Saxion University of Applied Science

Supervising Teachers

**Bram den Hond & Taco van Loon**

22.08.2016

## Summary

Photogrammetry is a technique, which is already around for quite some time, but it's also some kind of trend that gets used more and more in games recently. One reason for this might be the greater availability of VRAM for bigger textures, but certainly also the trend of creation immersive environments for Virtual Reality.

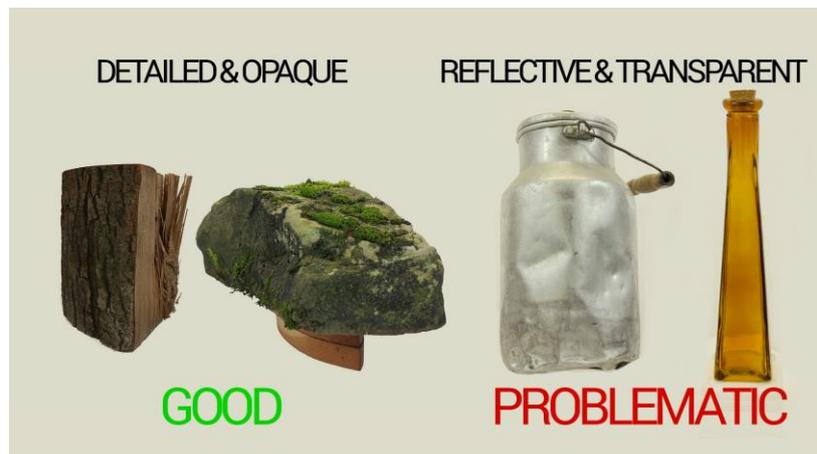
The term Photogrammetry describes a technique to reconstruct three dimensional scenes, or objects, from a series of overlapping photographs. As the accurate recreation of real world objects for games is a quiet time consuming process with a regular modeling & texturing workflow, it's interesting to see,

### *What are beneficial use-cases for photogrammetry in a 3D game?*

Therefore, this paper is examining the differences between a regular modeling & texturing workflow and photogrammetry as alternative, or addition. To examine its influences on the production speed & quality. Furthermore, this paper shows the challenges, which arise while applying photogrammetry, as well as its advantages. It's not meant as an exhaustive, step-by-step guide to photogrammetric scanning, but rather an introduction and collected set of hints and tips.

The gathered data shows that photogrammetry can offer a delightful addition to a regular asset pipeline, if the project requires a certain type of assets, which is ideal for photogrammetry. It offers a valid alternative in situations, where objects with a lot of surface details, like wear & tear, or weathering effects are needed. These type of objects are ideal, as the captured photographs allow an exact recreation of these little influences in a very detailed and realistic manner.

However, when objects are not ideal for capturing, the entire capturing and processing gets way more time intensive for an artist. As its time consuming to compensate these problematic material types. This makes photogrammetry less usable for projects, which require a lot of objects with reflective, or transparent parts, due to the increased effort that the creation of certain object types requires.



Therefore, photogrammetry offers a nice addition to the regular modeling workflow, to increase the quality of certain assets. Tough, it's not advisable to fully replace the manual asset creation, due to the increased effort that the creation of certain object types requires. It's ideal use-case are projects, which require old objects with a lot of wear & tear and history on their surfaces. These additional surface details are not only looking most impressive with photogrammetry, also they simplify the scanning process.

## Preface

First off I want to thank my teacher who convinced me to actually write this report about photogrammetry itself, as this allowed me to focus a bit more in detail on this awesome technology.

Furthermore, I want to thank the great team behind the Playcanvas engine, as they are building a great tool, without that I probably couldn't make all the interactive examples for this paper in the extend it is now. Of course not to forget, my colleagues who gave me throughout the creation of this report a lot of useful input.

# Table of Content

Introduction	1
Reason	2
Preliminary Problem statement	2
Theoretical framework	3
Definitions	3
General Model Requirements	4
Technique Comparison	5
Background Information	7
Definition of the problem	10
Main Question	10
Sub Questions	10
Scope	10
Research and design	11
Research results	12
1. Wooden log	12
2. Owl Decoration	13
3. Milk Can	14
4. Mossy Stone	15
5. Woodwaste	16
Conclusion and discussion	17
Recommendations	18
Graduation Products	19
References	20
Additional Resources:	20

## Introduction

The gaming industry is quite rapidly evolving industry, that is always pushing for new technologies. One thing that is already talked about for years, is the so called “Photo realistic” game, or environment. Although we are closer to it than ever before, with all the stunning cinematic game trailers nowadays and the recent trend in the industry to switch to a “Physically Based Rendering” (Russell, 2016) method, which caused a big step forward in realistic real-time rendering, we are still quite a bit away from a fully photo realistic game.

Nevertheless, there is already a technique, that produces some “real” photo realism in games, admittedly in a different meaning, the so called “Photogrammetry”. Photogrammetry is already around for quite some time, but it’s some kind of trend that gets used more and more in games recently. Whether it’s to capture accurate facial likeness of athletes in FIFA (DIMENSIONAL IMAGING LTD, 2016), or to visualize real world locations as accurate, as possible, like in the VR applications “Realities” (realities.io inc, 2016). Photogrammetry is a technique, that allows to convert a bunch of photos of an object, into a textured 3D model. This model includes all the details of the real world object, like wear and tear and all the influences the world had over time on the object.

This is precisely, that makes photogrammetry so interesting and useful. It’s not only making a virtual copy of an object in form of a mesh, but furthermore it also captures the history of the object, in form of scratches, dirt and all that stuff, which texture artists need a lot of time and experience for, to accurately recreate it.



For this reason, this report is examining what exactly photogrammetry is, which boundaries and benefits it has and in which use-cases it offers a good alternative to the standard way of manually modeling and texturing objects from scratch.

*What are beneficial use-cases for photogrammetry in a 3D game?*

As usual, there is no perfect, or only one way to look at a technique like this, as its usability always depends on the given use case and sometimes, personal preference. But this report is trying to give you a detailed inside into Photogrammetry, to allow you, to build your own opinion about it.

## Reason

Photogrammetry is something I am interested in already for quite a while, as I found it always delightful to be able to put real world objects into a virtual environment.

Since I knew a bit more about the technique and the workflow behind it, it's something, that I wanted to try out a bit more in depth on my own, but never had the time for.

Therefore, this "free" time of my study to write this report offers the perfect opportunity for me to look a bit more in detail at Photogrammetry. As Photogrammetry seems to be used more and more in the industry recently and the company, where I was doing my internship is also planning to look into photogrammetry, the topic is more relevant than ever.

The basic requirements to start with photogrammetry are quite simple, you only need a camera and a software to convert your captured images into a 3d model and its texture. Tough there are always a couple more things to be aware of to get ideal results.

Therefore, I would like to have a look at the general boundaries and benefits of photogrammetry and look at its usability in a few different use cases to be able to draw a conclusion, for which type of project its suitable. So that I am able to recommend at the end of my research, how and when photogrammetry can be beneficial to be used in a project.

### **Preliminary Problem statement**

Therefore the following preliminary problem statement was built:

*What are efficient use-cases for photogrammetry?*

# Theoretical framework

The theoretical framework is used to build a foundation for the further research done throughout this report. Furthermore, it defines the used terminologies and technologies for the reader to make sure they can follow the further explanations throughout this report.

## Definitions

This report is going to look a bit more into detail at Photogrammetry, to conclude for which kind of projects, or use cases, it would be a beneficial alternative to the regular modeling & texturing workflow. This assumes of course, what both terms are actually standing for in the context of this report. Therefore, before getting started, lets clarify these definitions, to avoid misinterpretation.

### What does the term Photogrammetry mean?

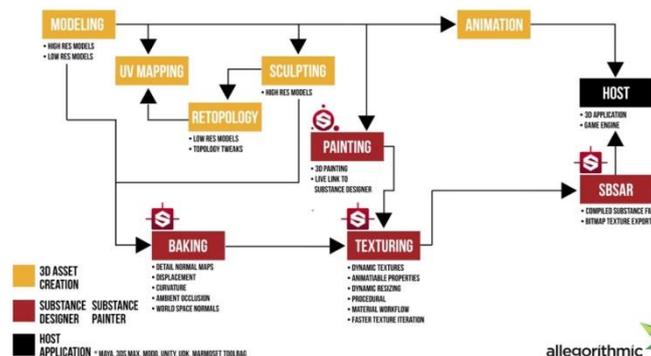
The term "Photogrammetry" stands for the technology to reconstruct three dimensional scenes or objects, from a series of overlapping photographs. It's used throughout this report as reference to the technique of generating 3D models & textures for games from a series of photographs. This includes the process of taking the images of the object, as well as its processing in a photogrammetry software and its optimization for the use in a game engine.



(FBFX Digital - What we do, 2016)

### What means a regular modeling & texturing workflow?

The term "regular modeling & texturing workflow" used in this report stands for the workflow of creating three dimensional models in a 3D modeling application like 3Ds Max, or Maya (Polycount, PropsModeling - Polycount, 2016) and to texture them afterwards in Photoshop, or Substance (Polycount, PropsTexturing - Polycount, 2015) based on some reference images, or concept art.



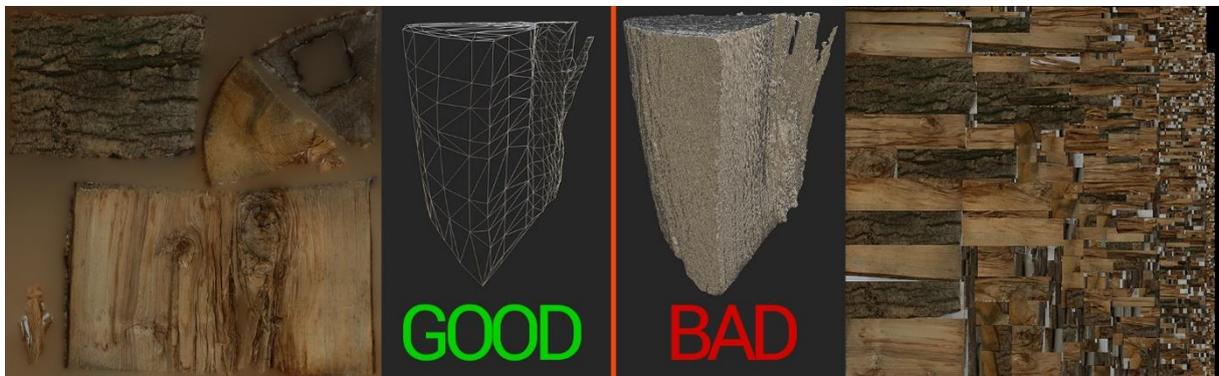
(Allegorithmic, 2016)

## General Model Requirements

The number of different model & objects types is probably as high as the number of projects you can think of. It makes of course a difference if you are going to model a small object like a bucket, or try to capture an entire scene like a cathedral. Sometimes your objects will require a proper texture unwrap, but sometimes tile-able textures or shaders will be more beneficial for you.

To handle all this different use cases would of course blow up the scope of this report, therefore we are only going to compare a single type of objects to give a general overview about the boundaries and benefits of photogrammetry. Though I will try to explain the different aspects of photogrammetry, as clear as possible, so it should be transferable to different use-cases and projects as well.

Therefore, this report is mainly focusing on the creation of smaller assets, to dress up a scene for a game. The kind of assets, that 3D artists have to create in masses to detail their game environments. These assets should have a game ready topology and a proper unwrap with PBR ready textures.



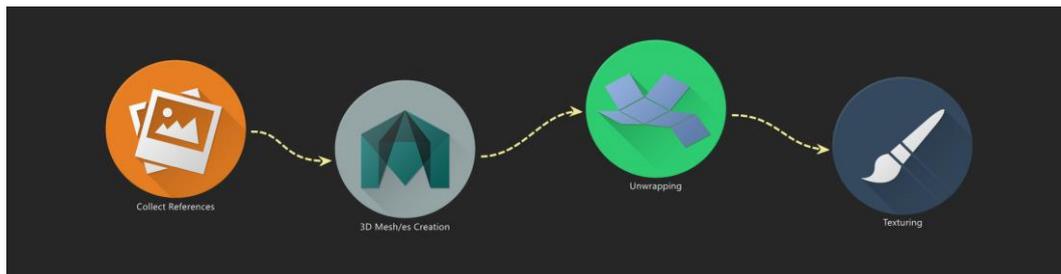
Game-ready means, that their polygon count stays in a reasonable and shape supporting extent, to be still render-able as part of complex environments in a real-time application, like a game. Furthermore, the UV-layout should be clean enough to avoid additional draw-calls.

## Technique Comparison

Before we are able to compare both techniques, let's first have a look at some example workflows for both techniques and what general advantages & disadvantages they have.

### Regular Modeling & Texturing

Most 3D models for games are created manually (or with the help of some scripts & tools) in a specialized 3D modeling software like 3Ds Max, or Maya. Afterwards they get textured again by hand in a texturing suite like Substance, or Photoshop. This requires of course a lot of manual work for an artist and also a lot of experience, if the objects should have some realistic aging and wear & tear.



### Example workflow:

An example workflow for this technique would be that;

1. Somebody has to create, or search reference photos of the object, which should be created.
2. An artist builds the 3D model in his modeling application of choice, based on the reference images.
  - a. Depending on the type of model and quality requirements, a high- and a low-poly model has to be created, to bake down additional details onto the game-ready low-poly model.
3. The model has to be unwrapped by the artist.
4. Afterwards the model has to be textured in a separate application again, based on the provided reference images.

That's a rather simplified overview about the different creation steps without consideration of additional requirements as collision models etc. just to give an overview about the steps, which are interesting for our comparison.

### Advantages & Disadvantages

#### Pros:

- ❖ Most 3D Artists know this technique.
- ❖ It does not necessarily require additional hardware, or space to capture images of the objects, as enough images of most objects can be found on the internet as well.
- ❖ It works with all kind of objects.
- ❖ Fictive objects can be created based on concept art.

#### Cons:

- ❖ Requires a lot of manual work.
- ❖ Requires a lot of skill for the texture creation to not look too artificial.

## Photogrammetry

For this technique a series of photos will be taken from different angles of a real world object. These photos will then be transformed into a point cloud by a special software, to be able to generate 3D models and textures from the images.



### Example workflow:

An example workflow for this technique would be that;

1. An artist captures a series of photographs of the object.
  - a. If necessary, the images will be cleaned up.
2. The photographs are taken into a photogrammetry software.
3. The output model of the Photogrammetry software gets re-topologized by an 3D Artist.
4. The re-topologized model has to be unwrapped by the artist.
5. Afterwards texture maps from the Photogrammetry Software can be baked down onto the re-topologized model.
6. Additional texture maps, like roughness and metalness have to be generated by a texture artist, based on the diffuse texture, with a software like Bitmap2Material.

On the first look, photogrammetry requires more steps in the creation process of a game ready model. However, some of these steps are pretty much automated, so that artists don't have to do too much manual work in the end. *(For more details see Research and Design)*

### Advantages & Disadvantages

#### Pros:

- ❖ Can be automated to a certain extend.
- ❖ Generates accurate and realistic looking models with real world aging and wear & tear.

#### Cons:

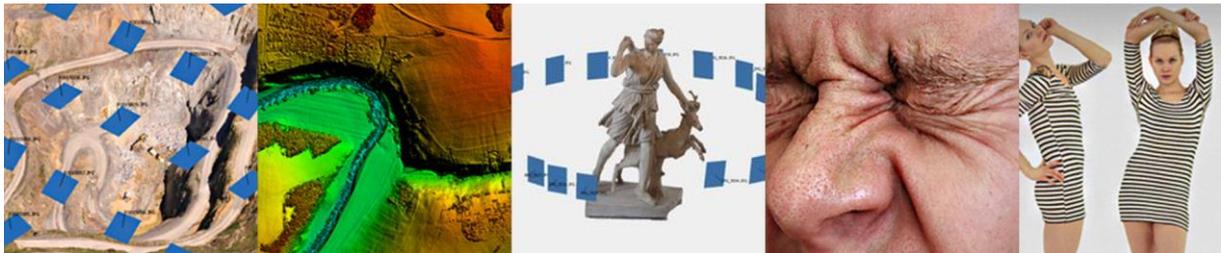
- ❖ Not every 3D artist has experience with this technique.
- ❖ It requires additional equipment like Cameras & sufficient Capture space.
- ❖ Not every object is scan able, like transparent or strong reflecting objects.
- ❖ You need the real world object at hand.

## Background Information

### Overview of Software

There are a couple of applications out there (wikipedia, 2016) to convert photographs into 3D Models, as usual, each with its own advantages and disadvantages.

These applications are reaching from specialized software to convert images from drones (UAVs) into meshes of entire landscapes, to software that aims at capturing facial expressions of real world characters as detailed as possible.



*Composition of images from: (AgiSoft LLC, 2016)*

As this report is only explaining photogrammetry in general and not its use in a specific software I won't talk in detail about all the different software solutions, which are out there, but I would like to give you some insights from my research, so you can decide what's worth looking into for you.

The probably most used Photogrammetry Software out there for Game Development is [AgiSoft Photoscan](#), which was used in games like "The Vanishing of Ethan Carter" (Poznanski, 2014) who did a phenomenal job. Photoscan offers an all in one solution to convert your images into 3D Meshes and textures in one go (except the re-topologizing ofc.), its relatively easy to use and offers an affordable small version, with enough features to start your photogrammetry project.

Another commonly used software is [123Catch](#) from Autodesk, which performs all the heavy calculation in the cloud. Its available in a free version as app for your phone and pc, therefore it's nice to gain some first insights into photogrammetry (Autodesk, 2016). Grab your Mobile phone and an object and fiddle a bit around with it. Although it's nice to play around with the principle of photogrammetry it's not really useable for a game production, due to the limitations Autodesk gives you on the amount and resolution of used photographs. So if this report raised your interest in photogrammetry, give it a try, but don't expect perfect results.

A bit more complicated but more adaptable and free software is [VisualSFM](#). With it alone you won't come far, as its just handling one part of the pipeline (Wu, 2016). In combination to this you will need a dense reconstruction tool like PMVS/CMVS and a software like "[MeshLab](#)" to process their output mesh in the end (you still need to re-topologize and bake this output to get a proper model). A good tutorial for using this bundle can be found [here](#).

Although this setup might sound way more complex, than the use of AgiSoft (in fact it is), it also offers some very nice advantages. Its relatively modular, as things like the dense reconstruction tools it uses, can be simply swapped out and in addition to that it also offers a command line option as well as versions for Windows, Linux and Mac OSX.

That was actually the main reason why I looked more in depth into this software “bundle”. I like to automate as much repetitive work as possible, with closed source software that can be tricky sometimes. Furthermore, photogrammetry is quite computation extensive, therefore my work machine is unusable for a few minutes every time I start to calculate some meshes.

The command line options and multi-OS support of VisualSFM allows to build a custom render farm, by simply splitting parts of the process across different machines in the office.

### Hardware Requirements

Photogrammetry has some requirements on the used hardware, although they are greatly influenced by the quality level you want to achieve. In theory you can capture almost everything as good with your mobile phones camera as with an expensive DSLR Camera, but obviously the results will differ in the end.



Most of the previously mentioned Photogrammetry Software’s have own guides mentioning the required hardware to run their software, as well as the required capturing equipment. So that’s stuff you should check out, before you start shooting.

A rough general guideline is, that you should use a camera with at least 5 megapixels, to get useable images, while around 12 are ideal. It is reasonable to remember that the resolution of the input images influences the quality of the processing results, but also the calculation time. Therefore, don’t go for a much higher resolution, if you don’t miss information in your photographs, or you calculate for days on every single object. More important than the pure resolution, is the size of the camera’s sensor and the option to lock its ISO, since photogrammetry works on identifying small details across photos, you want shots to be as sharp and as free from sensor noise as possible.

In addition to the camera, it’s also worth investing in a proper tripod or single pod, as blurred images are useless. A cheap alternative for a professional single stick is to simply mount a camera holder onto a telescopic bar from some gardening equipment.



### Limitations of Object/Material Types

Due to the fact how Photogrammetry works, identifying small details across multiple photos, there are some limitations regarding the objects which are capture-able. While things like cliffs, or stone walls capture perfectly well, due to their natural roughness, it's hard, or impossible to capture shiny objects like cars, or transparent objects like glasses. This is caused by the fact, that the algorithm calculating the three dimensional models in the software is trying to match the pixels between the different images which are provided. If it finds enough matching pixels between the images, it will be able to generate a point cloud from them, but specular highlights and reflections will horribly confuse it.

Therefore, try to capture matte or opaque objects only, but if you really need to capture shiny or transparent objects, you can try to cover all specular highlights and reflections. Some ways to do this is painting the object, or to sprinkle calk dust over it, but keep in mind that you won't get any texture information from your objects this way.

### Influence of Lights & Shadows

Another important aspect of photogrammetry to keep in mind, is the way you light your object while capturing it.

For objects, like props, that you want to place in different environments in your game, it's imperative to capture them without any direct light pointing at them. As this light would be baked into the texture in the end, which results in odd lighting, if not all objects in the scene are captured with the same light influence.



On the other hand, if you capture an entire scene, like a small room, this direct light influence might be exactly what you want. As this baked light will look very realistic (because it is real ;D) and does not require any additional light setup in your game, if you don't need any dynamic light sources.

Another thing to keep in mind, that applies to both mentioned use-cases, is to avoid strong shadows, or highlights caused by a wrong light setup. As each photograph needs comparable pixels with its neighbors in the image sequence, its self-speaking, that strong shadows, or highlights would confuse the comparing algorithm. The same part of the object would have different color values, or totally white or black pixels, that end up in artefacts, or skipped points in the point cloud.

Therefore, it's important to capture objects in as diffuse light as possible, to avoid any unwanted shadows, or strong highlights in the image sequence. If you want to capture smaller objects and have some equipment like a light diffuser to hand, it's advisable to light your object equally from all sides with them, to get rid of the shadows. If you don't have access to this hardware just try to avoid any direct light on your object, or capture you're your scene on a cloudy day, if the sun isn't shining too strong.

## Definition of the problem

The previous research shows some limitations of photogrammetry regarding the type of objects and materials, as well as the surrounding in form of light and reachability by the photographer. Furthermore, it is visible that with photogrammetry created 3D models can offer beautiful natural and irregular textures, which are hard to create for artists. In addition, the creation of certain meshes can be quicker with photogrammetry, compared to the regular modelling & texturing workflow depending on their characteristics and the available resources.

### Main Question

Therefore, the following main question can be formed:

*What are beneficial use-cases for photogrammetry in a 3D game?*

### Sub Questions

Now after our main research question is formulated, we have to find out, what is needed to answer it, while staying in the scope of this report. Therefore, we have to clarify a few things up front:

- ❖ *How useful is photogrammetry for scanning objects that are used in a 3D game?*
- ❖ *What kind of objects are suitable for photogrammetry?*
- ❖ *What challenges are there when applying photogrammetry?*

With answers to these sub questions it's possible to conclude which technique can be preferred on a given use case. Although as the previous research showed, that both techniques have their own limitations and ideal use cases, this paper is not going to give a fixed rule, but rather a recommendation for certain use cases, in which one technique could be preferred over the other.

## Scope

The scope of this report is bound to the research, that is required to recommend one of the listed asset creation techniques, for certain use cases, or project requirements.

Factors which are influencing this recommendation are the technical limitations and advantages of both techniques, as well as their usability for the artist.

Points that are out of the scope of this report, but could have influenced the recommendation of this report are, although briefly mentioned, partly automated asset pipelines with self-written scripts, or additional software packages and additional hardware like automated turntables.

## Research and design

The main research method, which is used to gather the data required for the recommendation of this paper is to validate the gathered background data with some praxis oriented test cases.

These test cases are built to answer the list of sub questions, which raised while collecting the background information needed for this research. While the collection of these background information helped to figure out the more and less important aspects of the compared asset creation workflows, they give no clear answer to the main question. Therefore, the test cases are used to verify the collected theoretical data and also to give a more in depth answer to the given main question. In addition, they offer some interesting content to visualize the more theoretical parts of the investigated techniques.

*To be able to find out the pros and cons of photogrammetry I selected some best and worst case scenarios based on my previous research and looked for suitable objects to represent each of these scenarios. These scans should help to verify the previous made research, or to find additional point of interest.*

I ended up with the following objects for the different scan scenarios:

1. **Wooden log** – Represents an ideal use case for photogrammetry due to the detailed organic shape. In addition to that its easy moveable, thus simplifying the scanning process.
2. **Owl Decoration** – Represents a more problematic, but still doable use case for photogrammetry due to the transparent parts of the object.
3. **Milk Can** – Represents a bad, but still doable use case for photogrammetry due to the strong reflection and smaller parts of the surface.
4. **Mossy Stone** – A comparison of the effect, that the used camera resolution has on the output model and its texture.
5. **Woodwaste** – Represents additional challenges, which arise if scanning a not moveable objects outside.

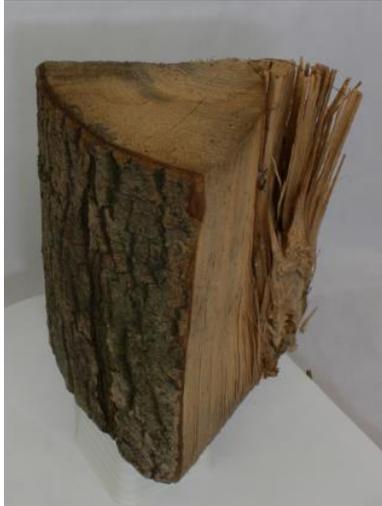
*Note: Additional use cases that I haven't covered here, but I recommend people to look into, if they are interested in photogrammetry. The scan of fully transparent objects like glasses, or the scan of room with interiors, or building facades.*

A positive side effect of building the test projects is that their creation allows a basic comparison of the usability of both workflows in a production environment. As its generally not feasible to use a technique, even if it offers slightly better results, if it's too hard to use, or too time consuming. Therefore, this is another important aspect, that should be taken into account for a recommendation.

# Research results

## 1. Wooden log

A wooden log is a relatively easy object to scan, so it is a quite good starting point to get used to the process of photogrammetry, while learning the dos and don'ts in an easy way.



A split block of wood as the one shown on the side offers quite a few interesting things for photogrammetry.

- ❖ It is relatively small and lightweight, what makes it easier to find a good place to scan it. (You don't need a big room, or think about the weather conditions)
- ❖ It has an interesting organic surface with a lot of details, which would require quite some time to sculpt by hand. In addition, this is beneficial for the photogrammetry software, as it's easy to find matching points.
- ❖ The splintered parts on the side show nicely some problematic shapes for photogrammetry, as they can produce some floating parts in the geometry, if they get too small.

*Note: I made two scans from of the log, one upside down to be able to fully capture all sides of the log. Their output was merged afterwards to have a fully captured model from all angles. Both scans container 3 image rows, top, mid and bottom of 24 images per row.*

### 3D-Viewer:

<https://playcanv.as/p/FLpNI0EL/?object=wooden-log>

### Conclusion

A Split piece of wood offers a perfect use case for photogrammetry, because the scan process is a quick and easy task, due to the handiness of the object. Furthermore, the object offers an interesting and detailed surface, that would require a lot of time to manually create it. The only thing to watch out for are small splintered parts, as they can result in floating points and scan errors.



## 2. Owl Decoration

A small decoration object like the solar owl lamp on the side shares the handiness and surface detail (the feathers add some nice surface detail) of the above shown wooden log. Additionally, it has a particularity, that makes it a bit harder to properly capture it.

- ❖ The eyes and the solar panel on the object are transparent/semi-transparent, which produces artifacts and noise areas on the output mesh.

- ❖ A simple solution for object like this is to cover the transparent/semi-transparent parts of the object with something opaque, that you can easily remove later on. This way it's possible to capture at least the surface of these objects as well.

- ❖ Planar transparent parts like the ones on this owl, can be easily textured manually afterwards by the use of an additional photographed texture, which can be projected onto the mesh again in an application like Substance Painter.



*Note: This mesh is made with a single scan of 3 images rows (top, mid, bottom), with the transparent parts covered by some duct tape. These parts are afterwards manually textured with the help of additional images of the affected parts.*

### 3D-Viewer:

<https://playcanv.as/p/FLpNI0EL/?object=owl-decoration>

### Conclusion

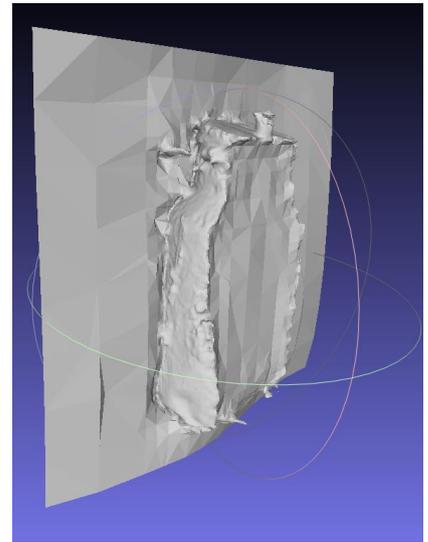
The transparency of objects is a bit problematic for photogrammetry, as it causes artifacts and unclear surface definitions, but with a few tricks it's also possible to capture these types of objects. Although if it's worth the effort, depends on the complexity of the object. This owl allowed a relatively quick and easy manual texturing of the transparent objects, because all transparent parts are planar. For more complex shapes this manual texturing might not be as easy anymore and therefore might reduce the advantage of scanning the object.



### 3. Milk Can

An old dented milk can like the one on the side might look like a nice object for photogrammetry, as it's quite handy and shows some nice wear and tear, which shows some nice history. Although the shiny material makes it a bit harder to capture, than it might look like.

- ❖ Due to the reflectivity of the surface the light in the capture area has to be reduced, to avoid strong highlights and reflections.
  - ❖ The reduced light intensity requires adjustment of the camera to still get bright enough images.
  - ❖ Small parts like the wire of the handle, offer similar problems as the splintered parts of the log shown above.
- ❖ As the surface of the can is beside the dents quite detail less, it's not offering a lot of matching points, which the photogrammetry software could work with. Therefore, it's hard for the software to recreate the proper surface into a mesh, this makes the post processing more complex, as the point cloud tends to produce output meshes like the one on the right.
  - ❖ If it's the goal to only capture the surface, without its texture detail, the capture process of this type of objects can be greatly simplified with covering up the surface, like shown on the owl above.



*Note: This mesh is made with a single scan of 3 images rows (top, mid, bottom), but it required a few test runs to find the right mix of light intensity and camera settings. Furthermore, the image post processing required a lot more time than on the other models shown above.*

#### 3D-Viewer:

<https://playcanv.as/p/FLpNI0EL/?object=milk-can>

#### Conclusion

To capture reflective or shiny objects like this milk can is still possible with photogrammetry, as long as they offer enough surface detail and don't reflect too much of their environment. So that the software can find enough matching points, to properly reconstruct its surface. Although the capturing process of these objects tends to be way more complex, than for less reflective objects. It requires way more thinking about the proper light setup and image correction, to take good images for the use in a photogrammetry software. This makes the overall process more complex and less useful for simpler objects compared to a manual modeling and texturing workflow.



#### 4. Mossy Stone

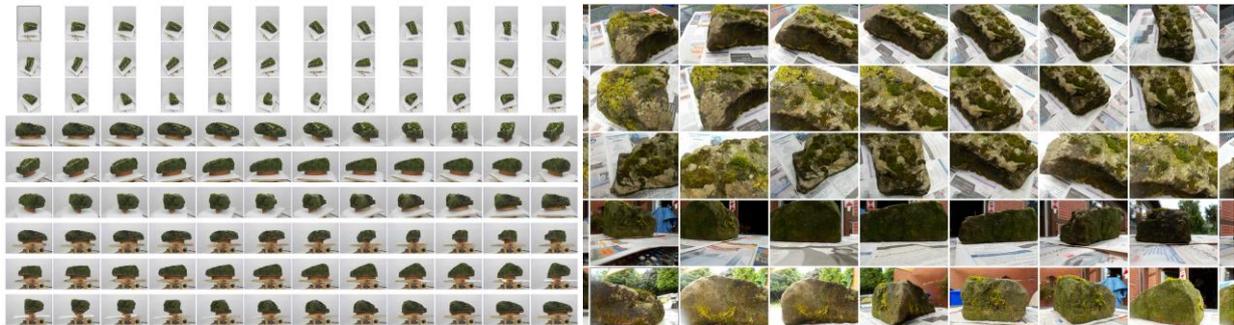
A mossy stone like the one shown on the side offers a quite ideal use case for photogrammetry, as the moss and weather influences offer cool and unique surface details with a lot of match-able points.

That's in fact one of the reason, why it was one of the first objects, which I tried to scan with an old compact camera I still had lying around. After a while I switched to a newer DSLR camera and asked myself, what influence the camera actually has on the output quality, so I recaptured the

stone with the DSLR and made a comparison of both output meshes.

- ❖ The overall shape is perfectly matching with both cameras, though the higher resolution of the DSLR adds a bit more micro definition to the mesh.
- ❖ The higher resolution of the DSLR allows the creation of a texture with a slightly higher end resolution, which is beneficial for close up, or hero models, but it also increases the computation time.

*Note: The two meshes were not captured with a fully identical setup, as I captured both scans with different cameras and some temporal distance between the shots. In addition, the capturing setup differs a bit, since I made the first scan on our garden table, while the second one was made in a lid studio. If somebody is interested in actually comparing the influences of different aspects of used cameras, like their resolution, more in depth, I suggest to do some more comparable scans with identical environment setups.*



#### 3D-Viewer:

<https://playcanv.as/p/FLpNI0EL/?object=mossy-stone>

#### Conclusion

Although the two scans of the above shown stone are not fully identical, due to different camera models and environment conditions, they show some advantages of using a bit more advanced camera for photogrammetry. But they certainly also show that it is not necessary required, to use an expensive DSLR camera, if you don't plan to go too close up to the scanned meshes. Tough a DSLR certainly offers more options to the user while shooting, which might be helpful while capturing larger amounts of images.

## 5. Woodwaste

After I captured quite a few smaller objects, which I could easily move around and light up correctly, I was a bit more curious about the additional challenges which arise, while capturing bigger objects outside. So I went outside and started to look for something interesting to capture. I found an old pile of rotten wood, which was ideal, as its surrounding was relatively open, so I could easily move around it.



A few things I have noticed, which should be considered while capturing objects outside:

- ❖ Watch the weather forecast, as its really annoying to do the same capture twice, just because it starts to rain 5minutes before you got all your images. The sun can get really annoying as well, if it appears every second image behind a cloud.
- ❖ A single pod to mount your camera makes capturing way easier and quicker.
- ❖ For outside shooting it's even more important to make a few more images, than you think you need, as it's hard to get equal lighting conditions again. So it can happen, that you have to reshoot everything, even if you only need one additional image.

### 3D-Viewer:

<https://playcanv.as/p/FLpNI0EL/?object=woodwaste>

### Conclusion

Capturing objects outside can make quite some fun and offers a unique opportunity to add objects with a lot of weathering effects to your virtual environments, but it can also be a bit tricky and frustrating. A lot more factors have to be thought of, which you cannot necessary control, like the weather at the capture scene.

## Conclusion and discussion

The use of photogrammetry offers a valid alternative for the regular modeling workflow in situations, where objects with a lot of surface details and/or wear & tear, or weathering effects with a realistic appearance are needed. These types of objects are the strength of photogrammetry, as it shows all these little influences in a very detailed and realistic manner. In addition, the capturing process of these object types is relatively quick and easy.



However, when objects don't have a lot of surface definition, or problematic material types like reflective, or transparent parts, the capturing and processing gets way more time intensive. Furthermore, the results of these objects tend to be less impressive, as stuff like glass has to be faked with shaders. Moreover, actual surface detail can be lost, because the photogrammetry software cannot find enough matching points on these material types, to reconstruct its surface details.

While this shows that the use of photogrammetry offers a nice addition to the regular modeling workflow, to increase the quality of certain assets. It also shows that it cannot fully replace the manual creation of assets from scratch, due to the increased effort that the creation of certain object types requires. So that the usability of photogrammetry remains a question of the project requirements, rather than limitations of hard- or software, as the starting barrier with the variety of paid and free software with little hardware requirements is quite low.

Therefore, beneficial use cases for photogrammetry are projects which include objects with mostly opaque and rough surfaces. The more surface details an object has, the easier the scanning gets. The strength of photogrammetry is the scan of old and worn down objects with a lot of wear & tear and history on their surface. As the virtual recreation of these influences is the easier and more impressive usage of photogrammetry.

As mentioned earlier, this article is not meant as an exhaustive, step-by-step guide to photogrammetric scanning, its rather an introduction and collected set of hints and tips. Therefore, the use of additional equipment or software could influence the quality of certain scans.

## Recommendations

The use of Photogrammetry offers a valid addition to a regular modeling workflow for projects, which:

- ❖ Require objects with a lot of surface details and/or wear & tear, or weathering effects with a realistic appearance.
- ❖ Don't contain a lot of objects with reflective, or transparent parts.

Furthermore, it's advisable to start with easier objects like the shown wooden log and some simple available equipment and then to gradually increase the difficulty of the objects to get a feel for the things, which might work, or don't work. Before spending a lot of money on additional equipment.

For rough scans of normal props, a simple camera with 5-9 megapixels should be okay for an up to date game, but if close-up, or hero models should be scanned in its useful to have a few megapixels more to get some additional surface details into the scans. Moreover, the more control you have over your camera, the better. As the ability to manual focus the target object and the ability to adjust the ISO and other options makes the capturing process way easier.

For scanning smaller moveable objects, a turntable is highly recommended, as it will greatly increase your capture speed. Furthermore, some studio lights will allow you to lit up your objects more evenly, so you can capture them regardless of the current lighting. In addition, a green screen will make your life a bit easier, as it allows you to cut off the background, which makes the post processing a lot easier.

What is important, however, while using a turntable is that you don't capture a lot of the static background. As the background would produce more matching point than your object on the turntable, which will result in broken scan. A common trick is to put a piece of newspaper under the object, so that bigger parts of your image are moving with the object. A slightly reflecting piece of paper works even better, as most photogrammetry applications cannot reproduce it, so that you don't have to cut it away afterwards. Another good trick is to lift your object a bit with some kind of pillar, to capture the detail on the bottom of the object. Furthermore, it helps to avoid ambient occlusion on the lower part of the object.

To get as much information into your scans as possible its recommended to do three rows of image sequences of your object. Once from a high angle, once directly from the front and once from a lower angle. Each row should contain a 360-degree sequence with images taken every 5 – 15 degrees depending on the complexity and desired amount of surface details.



For capturing outdoor scenes, or objects its quite important to plan in the weather and sun conditions on the planned scene, to avoid unnecessary breaks, or additional shooting days. Ideal is an overcast, or cloudy day with defuse light. Furthermore, it's advisable to shot always a couple more images, than you actually need, to have some angles in reserve for the case you miss some parts in the reconstruction days after the actual shooting. In addition, a single stick with an adjustable height is a great helper for capturing bigger objects from different heights, it also allows you to quicker capture sharp images, than with a tripod.

## *Graduation Products*

The prototype created for this paper can be found under the following link:

<https://playcanv.as/p/FLpNI0EL/>

## References

- Agisoft LLC. (2016, 08 15). *Agisoft Photoscan*. Retrieved from Agisoft: <http://www.agisoft.com/>
- Allegorithmic. (2016, 06 16). Retrieved from <https://www.allegorithmic.com/>:  
<https://www.allegorithmic.com/>
- Autodesk. (2016, 06 30). *123 Catch How To*. Retrieved from 123dapp.com:  
<http://www.123dapp.com/howto/catch>
- DIMENSIONAL IMAGING LTD. (2016, 06 14). *FIFA*. Retrieved from di4d:  
<http://www.di4d.com/work/entertainment/fifa/>
- FBFX Digital - What we do*. (2016, 08 16). Retrieved from FBFX Digital:  
<http://www.fbfxdigital.co.uk/#!what-we-do/c71j>
- Polycount. (2015, 01 22). *PropsTexturing - Polycount*. Retrieved from Polycount:  
<http://wiki.polycount.com/wiki/Category:PropsTexturing>
- Polycount. (2016, 05 27). *PropsModeling - Polycount*. Retrieved from Polycount:  
[http://wiki.polycount.com/wiki/Prop\\_Modeling](http://wiki.polycount.com/wiki/Prop_Modeling)
- Poznanski, n. (2014, 03 25). *VISUAL REVOLUTION OF THE VANISHING OF ETHAN CARTER*. Retrieved from theastronauts.com: <http://www.theastronauts.com/2014/03/visual-revolution-vanishing-ethan-carter/>
- realities.io inc. (2016, 06 14). *realities*. Retrieved from realities: <http://realities.io/>
- Russell, J. (2016, 06 14). *PBR Theory | Marmoset*. Retrieved from Marmoset:  
<https://www.marmoset.co/toolbag/learn/pbr-theory>
- wikipedia. (2016, 05 26). *Comparison of photogrammetry software | wikipedia*. Retrieved from wikipedia: [https://en.wikipedia.org/wiki/Comparison\\_of\\_photogrammetry\\_software](https://en.wikipedia.org/wiki/Comparison_of_photogrammetry_software)
- Wu, C. (2016, 06 30). *VisualSFM : A Visual Structure from Motion System*. Retrieved from ccwu.me:  
<http://ccwu.me/vsfm/doc.html#usage>

### Additional Resources:

<http://steamcommunity.com/games/250820/announcements/detail/117448248511524033>

<http://steamcommunity.com/games/250820/announcements/detail/117448248511523471>

<http://steamcommunity.com/games/250820/announcements/detail/117448248511522945>

<http://www.photogrammetry.com/>

<http://www.geodetic.com/v-stars/what-is-photogrammetry.aspx>

<http://www.zarria.net/nrmphoto/nrmphoto.html>

<http://www.gdcvault.com/play/1020162/Crafting-a-Next-Gen-Material>