Application of Photogrammetry to Gaussian Splatting for mesh and texture reconstruction

Ken J. Kiisa[1]
Supervisor(s): Dr. Xucong Zhang[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2024

Name of the student: Ken J. Kiisa
Final project course: CSE3000 Research Project
Thesis committee: Dr. Xucong Zhang, Dr. Michael Weinmann

Abstract

Gaussian Splatting is a successful recent method for generating novel views of a scene based on photographs taken from that scene [1]. It uses rasterization in order to render the scenes it generates, which consist of 3D Gaussians. However, modern hardware and tools are designed and optimized around rendering polygonal and texture based models [2]. This paper proposes a method of extracting both a 3D model and texture file from a Gaussian Splatting scene by using renders of that scene in Photogrammetry. It shows that this can be a viable method for generating a traditional 3D model from Gaussian Splatting scene, and can for certain cases generate a model of comparable quality while lowering the number of initial images required by up to three times.

1. Introduction

Gaussian Splatting (GS) is one of the most successful recent developments in novel-view synthesis [3]. In it, images are taken and the scene of the images is recreated using a point cloud [1]. Since the publishing of this method in August 2023, many further developments and refinements of the method have been developed and published [3].

Most available tools and hardware are designed and optimized for rendering and manipulation of polygon based 3D models, which makes it desirable to extract a traditional mesh from the GS point cloud [2]. One of those recent developments is the introduction of SuGaR, by which a polygonal 3D mesh can be extracted from a Gaussian scene [2]. This method, however, has the downside of not offering a method to also extract the texture data from the point cloud, only the mesh. In order to make use of the full set of available tools, both a mesh and texture data would be ideal.

One of the methods for constructing a 3D model of a real world object is Photogrammetry (PG), in which several photographs are taken of said object from different angles and then applied to a program that produces a mesh and textures [4]. The method, however, is limited by the degree of overlap between images, necessitating a large number of input images for best results [5].

The overall question of this research paper is: "Is PG a viable method for extracting the polygonal 3D mesh and texture from a GS scene?" This is done both to evaluate the method as a way to extract a full 3D model from a Gaussian scene, as well as whether the amount of photographs needed for PG can be reduced via data augmentation with GS.

The research that will be conducted in this paper is done with three sub-questions, used to evaluate and answer the overall research question:

1. "Is there a significant difference in the quality measurements for PG output, when using renders of a GS scene for source images, in comparison to using photographs of the original object?" [6][7]
2. "Is there a significant difference in the quality measurements for PG output, when using renders from a GS scene that was trained from less images than another scene of the same object?"
3. "What visual artefacts form from the use of novel views synthesized by GS as input images for PG, in comparison to using photographs of the original model?"

This paper begins with a background on GS and the methods that have already been developed for model reconstruction. It will also discuss PG and some of its limitations. These questions were then answered via a controlled experiment, with both a qualitative and a qualitative analysis of the experiment's results.

2. Background

This section discusses the outline of what GS is, what methods have been developed to extract a Mesh from it, as well as PG and its limitations.

2.1. Gaussian Splatting

Gaussian Splatting is a recent successful development in novel-view synthesis methods [1][3]. It involves the creation of a 3D representation of a scene from photographs by creating a point cloud of 3D anisotropic (parameter changing based on viewing angle) Gaussians [1]. These Gaussians are then trained using gradient descent, via rerendering of the scene and comparison to the source images, to manipulate their scales, orientation and spherical harmonics used for colours, in order to accurately recreate the scene [1].

This method was successful as it not only boosted the training and rendering times, but it also produced comparable (and at times higher) novel view synthesis quality as previous developments in the field, such as NeRF and Plenoxels [1][8][9]. Most notable of these previous methods was NeRF, as this was the previous state of the art in the field which made use of a neural network for learning and reconstruction of a scene [10][1][3][9][11].

2.2. Existing Mesh & Texture Reconstruction Methods

GS's use of 3D Gaussians is what allows it to be rendered and trained as quickly as it is, however it comes with its limiations [1]. Most available hardware and tools for working with 3D visualizations are designed and optimized for working with 3D models built with polygons and textures, not Gaussians, incentivizing a method for transforming the Gaussian representation into a traditional polygonal 3D model [2][3][12][13].

Among these developments, a major one that extracts a mesh from a Gaussian scene is SuGaR, which applies a reg-

ularization term to the learning in order to incentivize the Gaussians to follow the surface of the object, and then applies Poisson reconstruction to extract the polygonal mesh with the shape of the object[3][2]. Another recent development is Texture-GS, which works to separate the texture data from the Gaussians for easier manipulation, the first of its kind to do so as far as the literature review for this paper found [13][3]. There have also been works such as the 3D Gaussian Model, which aims to circumvent the issue by proposing a rendering pipeline for Gaussians that make them as easily malleable as traditional rendering methods [12] [3].

Critically, however, there has been no published method by which to extract both the mesh and texture data from a Gaussian scene [3]. SuGaR only extracts a colorless mesh which it textures by flattening the associated Gaussians over it, and the texture data from Texture-GS is not generated in a way that can be readily accurately mapped to a mesh generated by a method like SuGaR [2][13]. From my research, a method for extracting a full model with mesh and texture data from Gaussian Splatting is not currently available [3][13].

2.3. Photogrammetry

A widely used method for extraction of a mesh and texture data from images is Photogrammetry (PG) [4]. The process involves the capture of several images of an object from different angles in order to produce an accurate representation of it in the form of a polygonal 3D model [4].

The mathematics and principles behind the method have been around for over a hundred years, and it has been put to use decades ago with the rise of computers to help create mathematically accurate representations [4]. As such, there are several mature software packages available that can be used for PG, such as RealityCapture [14].

However, the method comes with several limitations, namely that the generation of a high quality model requires a high degree of overlap between input images, necessitating a very large number of photographs to be taken with minimal differences [5]. In contrast, GS is able to generate models from comparably low image counts, most notably with the GaussianObject paper that managed to create viable scenes from only 4 input images [15].

As GS is designed for the synthesis of accurate novel views of an object, this begs the question of if these novel views can be used in PG to generate a full 3D model, and whether the minimal number of actual photographs of an object can be lowered by use of GS.

3. Controlled Experiment

A controlled experiment was performed to answer the first two research sub-questions. As both of the research questions cover a similar topic, that requires similar set up in order to validate, they will both be explored within the same experiment.

3.1. Hypotheses

The hypothesis that is associated with the first sub-question is: "The quality measures will be the same for models generated with PG using renders from GS and renders of the original model."

The hypothesis that is associated with the second sub-question is: "Using less source images for the training of a GS scene will negatively affect the quality measurements of its PG generated model."

3.2. Independent Variables

Within this experiment there are a number of independent variables, that are manipulated in order to measure the impact on the dependent variables:

- The dataset and the models selected from that dataset for use. A variety of models will be used to evaluate the efficacy of the method for different subjects.
- Number of images taken for training a GS scene. This will be varied to measure the impact of training image counts on final model quality.
- The perspectives that training photographs are taken from for each object [5]. Renders need to be taken that together provide a full view of the model in order to make sure that the model can be recreated fully and accurately.
- Number of images taken for use in PG. [5]. PG requires a high degree of overlap between input images for best results.
- The perspectives that renders are taken from for use in PG [5]. The same applies as for GS, full visibility of the model is needed for accurate recreation.

3.3. Confounding Variables

There are also a number of confounding variables that cannot be directly manipulated, but that affect the dependent variables, namely:

- Camera lens distortions, including lens quality, focal length, image texture [5].
- Camera resolution [5].
- Blurred and distorted images.
- The lighting of the objects when taking training photographs.

All of these limitations are controlled for by use of renders of a 3D model in a dedicated rendering software, which eliminates lens and camera distortions and other such defects unless manually enabled.

3.4. Dependent Variables & Measures

The dependent variables that will be affected and measured via changes to the independent variables are the 3D model

generated via PG, and the quality of that 3D model. The quality of the 3D models generated by PG will be measured with 5 heuristics:

- Average Track Length (ATL), denoting on average how many images a generated point appears in, which is used for alignment of images [16].
- Median Reprojection Error in Pixels (MRE), is the median error by which points may be offset from one image to the next [16].
- Percentage of Registered Cameras (%-RC), denoting the percentage of the original number of input images that the PG software actually used to generate a model [16].
- Peak Signal-to-Noise Ratio (PSNR) of renders of the final generated model, used to measure the degree of noise in an image compared to the original [7]. This is used for renders of both a lit, textured version of the final models, as well as one that displays the surface normals of the models, and compares them to such renders of the original input model. This is used to quantize how accurately the model visually captures the original. The lit renders measure how accurate it is to the original model while textured. The face-normal renders measure how accurately it recreates the geometry of the original model.

3.5. Procedure

The selected objects will be placed in a scene with a black background. Lighting will be set parallel with the camera's point of view, to simulate a ring-light around the camera. Each object will have 120 renders taken of it. These renders are taken in a sphere around the object, each on 3 horizontal layers on that sphere, 40 renders taken on each layer. The exact spherical coordinates of these camera's relative to an object at origin are written in a table in appendix C.1.

For each object, the GS algorithm will be trained three times: once with 120 renders, once with 60 renders, and once with 30. The scenes trained from these renders will be labelled 120-GS, 60-GS and 30-GS respectively.

From each GS scene, 182 renders will be taken, spherically as before, but in 4 layers of 45 images, and with two renders vertically above and below the object. A table with the spherical coordinates of these camera angles can be seen in appendix C.2. Each object will also have its original model rendered from these angles. These renders will be used as input images for PG. Thus, each object will have 4 model categories generated from it, including from renders of a gaussian scene trained with:

- 120 source renders: the '120-GS' model,
- 60 source renders: the '60-GS' model,
- 30 source renders: the '30-GS' model,

as well as from renders of the original object: the 'No-GS' model. From these, the generated 3D models will have the first 3 measures (ATL, MRE and %-RC) associated with it shown from the software used to create it.

These generated models will then be placed back in the original rendering scene, and renders will be taken from those same 182 camera angles. Renders will be taken with a lit version of the model, and a version that colorizes the model's polygon faces to display their surface normal direction. These two versions will have their PSNR measures evaluated separately. Each set of 182 PSNR results will be aggregated into a single median value for analysis.

3.6. Analysis Plan

For the analysis of the results, a series of statistical tests will be performed to verify any difference in the data. These will be compared to the null-hypotheses that relate to the two hypotheses. These are tested across all quality measures, and a significant difference in one denotes a significant difference in the overall category.

The null-hypothesis associated with the first hypothesis is: "There is no significant difference between the quality measurements resulting from the PG model generated from GS renders from 120 training images, compared to from renders of the original model." This also has two other null-hypotheses that are otherwise the same, but for Gaussian splatting scenes from 60 and 30 training images.

The null-hypothesis associated with the second hypothesis is: "There is no significant difference between the quality measurements resulting from the PG model generated from GS renders trained from 120 and 60 images." The null hypothesis is also repeated for comparison between 120 and 30 training images, and between 60 and 30 training images.

First, all of the values from the different objects will be pooled together per category and per measure. Meaning f.e. 120-GS/MRE will have 30 results containing the MRE of the models based on renders of GS with 120 training images. These 30 results are taken from each of the 30 original input models used in this experiment.

For each measure, pairs of model categories will be compared with a significant difference test. For each of these measures, a Shapiro-Wilk Test will be applied as a means to test whether the values in the pool are normally distributed [17][18]. If both pools are normally distributed according to the Shapiro-Wilk Test, then a T-Test will be used to evaluate a difference [19]. If one or both of the pools are non-parametric, then a Mann-Whitney U-Test will be performed instead, [20].

If a difference is found between two measures, then the means of the numbers will be compared to determine the superior model. For most measures, the higher mean will be chosen as superior, except for MRE, where a lower mean is better.

3.7. Qualitative Analysis

In order to answer the third research question, a qualitative analysis will also be performed on the resulting models gen-

erated by PG.

As finding a good set of quantitative quality measurements for PG is still an active field of research, the measurements listed before are not fully indicative of the quality of the model [6]. A qualitative analysis will be used to point out any visual artefacts, errors and abnormalities that the quantitative results can not show.

The qualitative analysis will be performed via visual inspection of each of the generated 3D models, both lit and textured, and unlit and colorized to show the normal direction of the faces. This way both the general look of the model, as well as its geometry can be easily inspected.

4. Experimental Setup

During the experiment described in the methodology, specific tools and set ups were used to perform data generation and analysis. These will be discussed in this section.

4.1. OmniObject3D Dataset

The dataset that was used for the controlled experiment was the OmniObject3D dataset [21]. This is a dataset consisting of several photographic scans of real life objects. It consists of both photographs and photogrammetrized 3D models generated from the photographs. The models used were found under the 'raw-scans' folder.

For this research, the first three 3D models from each category of item was used. For example: from anise.tar.gz, the first three folders, anise-001, anise-002, and anise-003, were taken. In the case that the categories didn't start with the numbers 1 through 3, the numerically sorted first three folders were taken.

The number of models from each category was limited to 3 due to time concerns with qualitatively analyzing each model's PG scan output.

4.2. Blender

Blender was used for the purposes of camera placement and rendering of the 3D objects taken from OmniObject3D's dataset [21] [22]. Within Blender, a script was prepared to take renders of each object from the dataset. This script was created largely from the blender rendering script in the NerF research paper repository [11]. Cameras are located on a 1 meter radius sphere centered on the origin. All of these cameras are pointed towards the origin of the scene. At each of these camera positions, a sunlight would also have its direction changed to be parallel to the camera's viewing angle. The exact script for rendering can be found under appendix D.

In order to make sure the different camera positions could properly view the objects, each model from the dataset was manually rescaled to fit within the viewing range of the cameras. The materials were also edited so that all objects have a roughness value of 1, in order to avoid artefacts and Gaussian floaters caused by light reflections.

4.3. Gaussian Splatting

For the generation of GS scenes, the original GS repository was used [23]. These scenes were trained on a Laptop RTX 4080 GPU. The resolution of the images used for training was 1920 by 1080 pixels.

For the generation of input images for PG, the SIBR-based renderer provided by the repository was used. The images were rendered by loading a pregenerated path and saving the images via the 'Play (No Interp)' button. This path is available under appendix D.

4.4. Photogrammetry

For the generation of 3D models from images, the RealityCapture program was used [14]. All of the default settings were unchanged during the experiment.

The resolution of all of the input images, from both Gaussian Splatting renders and images of the original model, was 2000 by 2000 pixels. For alignment of the scenes, the very first model, anise-1, was used. Control points were placed on renders of that model until the number of components was reduced to 1, at which point the metadata of the camera positions was exported and reused for all other objects. This process used 5 control points, which created camera alignment metadata for 180 of 182 input images. The two input images that were not able to have metadata generated for them, were the top and bottom views of the model.

During the generation of the models, after loading the images, the 'Align', 'Calculate Model (Normal Quality)' and 'Texture' steps of the process were used. The 'Simplify' step was skipped for the sake of geometry analysis, as it allows to see what the geometry of the model was before the program began to cull vertices. The 'Colorize' step was skipped, as the purpose of this research is to extract texture data, while colorizing only applies color data to vertices.

5. Results

In this section, the results of the experiment are handled. Do note, that out of the 30 models that were taken from the dataset for use in the experiment, asparagus-1, asparagus-2 and asparagus-3 were not recorded in the results. The asparagus models were not included, because, except for asparagus-3/120-GS, Gaussian splatting failed to create a representation of any of the models and instead produced blank scenes for each of them for all numbers of training images. A possible reason for the failure, was that the original input model is too thin, and there was insufficient overlapping area on the model between images.

In order to avoid biases caused by differing numbers of input values per category, models where at least one of the

Measure	No-GS	120-GS	60-GS	30-GS
ATL	3.766	3.630	3.538	3.102
MRE	0.319	0.366	0.386	0.411
%-RC	0.991	0.990	0.989	0.989
PSNR-Lit	40.136	39.841	39.747	39.452
PSNR-Normal	41.191	40.352	40.126	39.418

Table 1. The mean of each measure of each category across the 27 models included in the final results. Generally, the results worsen across all measures when comparing from No-GS, to 30-GS, to 60-GS, to 120-GS.

categories failed to generate a 3D model were excluded from the overall results. The results for the models that were able to be generated are recorded under appendix A.1. The results that were excluded from the analysis were also recorded under appendix A.2.

The aggregated means of the quantitative results can be found in table 1. The %-RC does not appear to change much between categories. From 120-GS to 60-GS there's only a 0.021% difference, between 60-GS and 30-GS there's only a 0.041% difference. No-GS boasts a slightly larger improvement of 1.64% over 120-GS.

From a viewing of the results, a trend can be seen where the results become worse from 120-GS to 60-GS to 30-GS. The difference between 120-GS and 60-GS is generally slight, on average an absolute difference of 2.2%, excluding %-RC. The difference between 60-GS and 30-GS is generally stronger, on average an absolute difference of 5.3%, excluding %-RC.

The scores when comparing 120-GS to No-GS appear to show an improvement with No-GS over 120-GS in all categories, with No-GS having an average absolute difference of 4.9%, excluding %-RC.

6. Responsible Research

As this research paper involves an experimental verification of a method, a number of risks are involved that need mitigation. The topics covered in this section are related to possible moral, reproducibility, data transparency & availability, as well as ecological concerns.

6.1. Moral

As this research paper involves the evaluation of a 3D model generation method, it does not handle any subjects or perform any experiments that could pose potential risks to individuals. The models used in the experiment were sourced from an open access dataset, mitigating any risks posed by potential breaches of privacy or consent when using real life objects. Blender and the SIBR renderer were accessed as open source renderers for 3D models and Gaussian Splats respectively [22][23]. RealityCapture was used as PG software made freely available for students [14].

6.2. Reproducibility

In order to make sure the experiment is as reproducible as possible, the methodology was described in as much detail as possible. These descriptions also include any errors technically issues or human errors that were made during the procedure of the experiment.

Further references to the appendix were also added for additional information on reproducing the experiment. Appendix C includes the full list of spherical coordinates denoting the camera positions used for gathering renders of models. Appendix D includes a link to a database containing the python scripts used to render the models in blender and perform the quantitative analysis. That document also includes the recorded camera positions used in the SIBR renderer for gathering images for use in PG, as well as RealityCapture camera alignment metadata files.

One aspect of the methodology that was not able to be fully detailed was the description of the qualitative analysis. As that form of analysis used visual inspection of the models as its basis, it is not possible to fully describe the method by which noteworthy elements of models are found and noted. The method suffers from the inherent bias of what aspects of a model does the viewer find noteworthy. This problem in itself cannot be mitigated, but the harmful effects of it were mitigated by making the models used during visual inspection available publicly in the database referenced in Appendix D. This way, each person that views this paper is free to take the models and inspect them for themselves.

6.3. Data Transparency & Availability

One risk is the potential misanalysis of results due to omission of other results that could conflict with the conclusions drawn. This was mitigated by including the full quantitative results in appendix A, with both the used and unused results recorded per input model.

In order to ensure the ability to contest the findings of the qualitative analysis, the final models produced by PG were made available as part of the database in Appendix D. These models are free to be used and placed into rendering software, such that each person reading this paper has the ability to visually inspect the final models for themselves.

6.4. Ecological

The final risk factor, is the eco-friendliness of a method such as the one presented here, in the case that it were to be widely adopted. The process of first performing GS, and then using those renders in PG, reduces the number of initial images required, but does so at a cost of computing power and electricity. This cost is especially present in scenarios where GS training can last upwards of half an hour to an

Measure	120-GS	60-GS	30-GS
ATL	0.333	0.105	$\mathbf{2.351e-6}$
MRE	$\mathbf{0.015}$	$\mathbf{4.194e-3}$	$\mathbf{3.831e-5}$
%-RC	0.077	$\mathbf{0.032}$	$\mathbf{2.588e-3}$
PSNR-Lit	0.522	0.406	0.161
PSNR-Normal	0.185	$\mathbf{0.048}$	$\mathbf{4.551e-3}$

Table 2. For each category, for each measure, p-value for difference to 'No-GS' model measures. Significant differences ($p < 0.05$) are marked with bold.

Measure	120 vs. 60	120 vs. 30	60 vs. 30
ATL	0.491	$\mathbf{4.965e-5}$	$\mathbf{5.710e-4}$
MRE	0.228	$\mathbf{8.993e-3}$	0.096
%-RC	0.654	0.081	0.161
PSNR-Lit	0.775	0.324	0.467
PSNR-Normal	0.494	0.069	0.172

Table 3. P-values for difference tests between Gaussian Splatting based models. Significant differences ($p < 0.05$) are marked with bold.

hour, and where PG can potentially take long as well. It is generally less environmentally taxing for more initial photographs to be taken, rather than to generate them, given that the process of image gathering is done by a person. In the future, this issue can hopefully be lessened by future developments in GS leading to further reductions in training time, or developments in hardware required to perform these tasks.

7. Discussion

7.1. Quantitative Analysis

The results of the difference test analyses comparing to No-GS can be found in table 2. Significant differences ($p < 0.05$) are marked with bold.

From the results, it appears that all GS based models significantly differ from No-GS in at least one measure. The MRE measures of all categories is significantly different from No-GS, with 120-GS having that as its only significant difference. The only measure that does not have any significant differences for any category is the Lit scene PSNR score.

In 60-GS, the %-RC and Face-normal rendered PSNR scores significantly differ from No-GS, with the MRE difference being very significant ($p < 0.01$). With 30-GS, every value except PSNR-Lit differs very significantly from No-GS. This rejects the null-hypothesis that there is no quantitative difference between the No-GS and GS based models, even with 120-GS.

In the comparisons between the different levels of Gaussian training, there appears to be no significant difference in quality between 120-GS and 60-GS. Comparing 60-GS to 30-GS a very significant difference appears for the ATL. Comparing 120-GS and 30-GS, very significant differences appear for the ATL and MRE scores. The %-RC and PSNR scores do not significantly differ between any of the GS based models.

This disproves the null-hypotheses, that there is no significant difference between 30-GS and higher training image counts. This indicates that the second hypothesis is correct. Reducing the number of training images down to 30 significantly deteriorates the quality measures of the generated 3D models. However, reducing the training image counts from 120 to 60 does not.

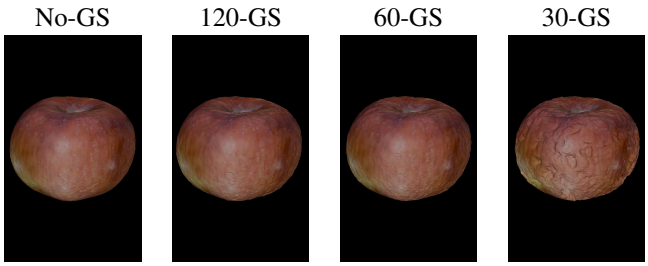

Figure 1. Visual comparison of textured model of '120-GS', '60-GS', '30-GS', and 'No-GS' models respectively for the apple-1 input model.

When going from PG with renders of the original model, to renders of a GS based representation of that model, there appears to be a sudden drop in quality in at least one aspect. However, when reducing the number of initial images down from 120 to 60, the quality of the model does not appear to significantly worsen any further. This would indicate that the method presented in this paper comes with a loss in quality, but that to a minimum of 60 initial images, that loss in quality does not significantly worsen when reducing images.

7.2. Qualitative Analysis

On visual inspection of the models, most appear to fairly accurately recreate the original models they were referenced from, although there are some exceptions. In most cases, the difference between 120-GS models and No-GS is visually negligible if not at times non-existent. In most of the models, the same also applies for the 60-GS models. With 30-GS is where visual decay begins to emerge, with a more noticeable loss in geometric detail, and the growing number and size of holes in the model's surface. The textures themselves of the models do not appear to have any artefacts in lit scenes, most of the visual decay is caused by the noisiness of the geometry. An example of this is in figure 1 and 2.

One feature where this is most noticeable is the rough-

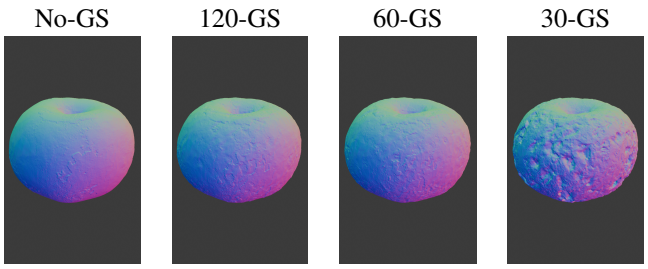

Figure 2. Visual comparison of geometry of '120-GS', '60-GS', '30-GS', and 'No-GS' models respectively for the apple-1 input model.

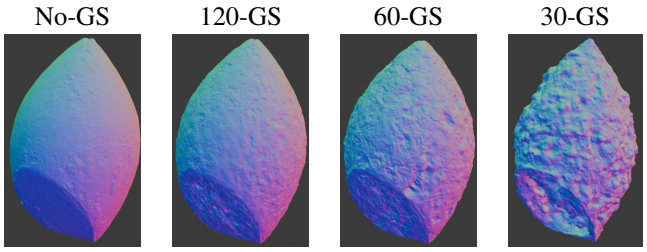

Figure 3. Visual comparison of geometry of 'No-GS', '120-GS', '60-GS' and '30-GS' models respectively for the beauty-blender-1 input model.

ness of smooth featureless surfaces when reducing the number of training images in GS. It is most easily demonstrated in figure 3. In this case, the roughness becomes so pronounced that it begins to affect the outline of the model's otherwise smooth curved silhouette. It is also a case where a small difference is present between the No-GS and 120-GS models. It does not appear to be caused by a lack of lighting information, as the spherical harmonics of Gaussian Splatting are able to successfully recreate the terminator gradient around the object's curved surface leading into shadow. This can be seen in figure 4.

A possible cause is the presence of floaters in the GS scene in angles where the training images were not able to have a clear view. Along with that, in the case of 30-GS models, the input images used for PG also showed parts of the scene object that appeared transparent and where the inner Gaussians were visible. These two problems being more prominent in lower training image counts in GS would explain why the difference in quality between 60-GS and 30-GS is so drastic. The floaters can be removed by reducing the GS scene rendering frame down to only the object. However, the presence of gaps in the object's Gaussian representation would need changes to how Gaussians are trained in order to fix, if possible.

Generally, featureless surfaces were struggled to be recreated across all the categories of models, an example of which being in figure 5, where many of the details of the

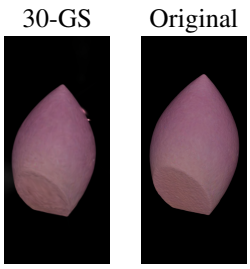

Figure 4. Comparison of gaussian splatting input image (left) to original 3D model render (right) used for generating the 30-GS and No-GS category models for beauty-blender-1 respectively.

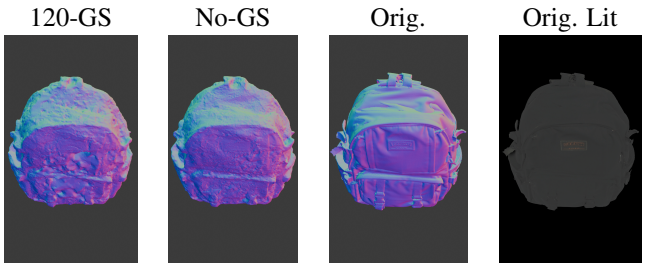

Figure 5. For backpack-1, renders of the geometry of the 120-GS, No-GS, and original model, as well as a lit render of the original model.

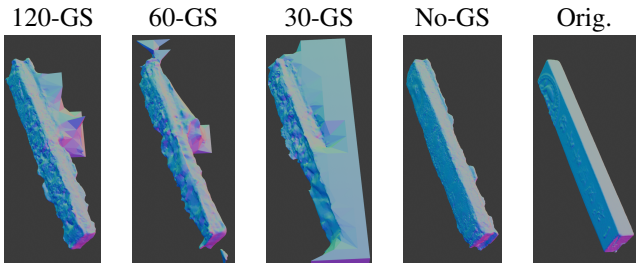

Figure 6. Visual comparison of geometry of '120-GS', '60-GS', '30-GS', 'No-GS', and original input models for antique-2.

backpack are lost in the noisy geometry. However, these effects were generally much more drastic in the Gaussian Splatting based models, as opposed to No-GS. In cases such as figure 6, all GS based models failed to accurately recreate the basic shape of the input model, with the effects worsening when reducing training image count.

However, a number of models have also shown that the there is little negligible difference between 120-GS, 60-GS and No-GS, such as in figure 7. For this model, very few apparent visual differences exist between the No-GS and 120-GS model. Even with the reduction to 60-GS, the model is still mostly identical to No-GS, with some noise in the crevices.

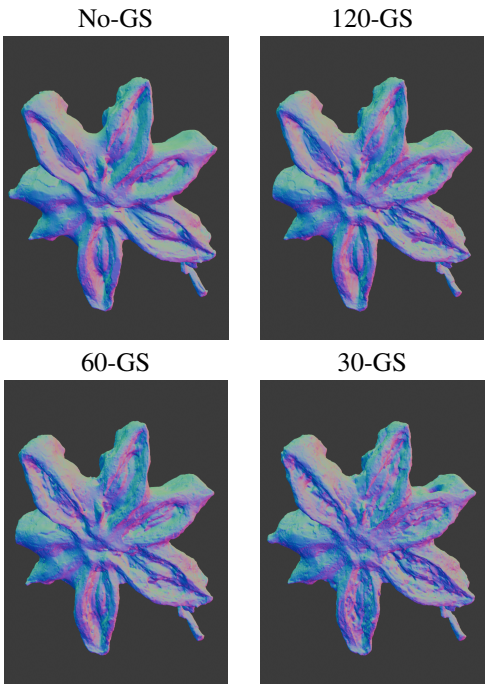

Figure 7. Visual comparison of geometry of 'No-GS', '120-GS', '60-GS', '30-GS', and models respectively for the anise-2 input model.

8. Conclusions

Through the experiment, it has been shown that it is possible to generate 3D models from a GS scene by using PG. It also shows that it is possible to augment the number of images used in PG by using them in GS first to train a scene and take more images from.

In the comparison of the quality measures for GS based models with No-GS based models, significant differences occurred for all levels of GS training, with more differences arising as the number of training images decreased. When comparing GS based models with each other, no significant differences occurred when reducing the training image count from 120 to 60. However, reducing the number of training images to 30 showed significant drops in quality compared to 60 and 120 images.

This method does come with some visual errors, namely that the geometry of a model becomes less detailed the less original images are used to train the GS model. Reducing the number of training images to 30 causes holes to appear in the models surface. The visual errors are also much more pronounced on models that consist of mostly smooth surfaces, and especially with featureless surfaces.

Despite these errors, the method is able to successfully generate a viable 3D model from Gaussian Splatting that is sometimes on par with only using original images, while reducing the amount of images required from the original object by up to 3 times.

9. Future Improvements

There are aspects of this research that could be improved or built upon in future papers. Something worth evaluating is if there is a tangible difference in the quality of models that are, for example, made with hundreds of renders taken of a Gaussian Splatting scene trained with only a few photographs, versus only using those few photographs as inputs for Photogrammetry.

A future paper could also use better, more finely tuned quality measures or more input models for a more robust basis for its conclusions. The practicality of this method can also be tested by making use of real photographs and scenes as sources of images for the experiment. This way its applicability for real life scenarios is evaluated.

It is also worth to instead explore a possible combination of works such as Texture-GS and SuGaR for full textured model extraction directly from a GS scene's Gaussians, versus using renders as inputs in already existing methods [2][13]. If such a line of research would also at some point include techniques from GaussianObject to further reduce the number of input photographs needed, then it could potentially introduce a novel method for PG to generate accurate 3D models while using far fewer images than is needed with current techniques [15]. In the future, a method such as that could be much more practical than by continuing to use PG for model extraction from GS.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering, August 2023. URL http://arxiv.org/abs/2308.04079. arXiv:2308.04079 [cs]. 1

[2] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering, December 2023. URL http://arxiv.org/abs/2311.12775. arXiv:2311.12775 [cs]. 1, 2, 8

[3] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent Advances in 3D Gaussian Splatting, April 2024. URL http://arxiv.org/abs/2403.11134. arXiv:2403.11134 [cs]. 1, 2

[4] T Schenk. Introduction to Photogrammetry. 1, 2

[5] Haiyan Sally Xie, Ioannis Brilakis, and Eduard Loscos. Reality Capture: Photography, Videos, Laser Scanning and Drones. In Marzia Bolpagni, Rui Gavina, and Diogo Ribeiro, editors, *Industry 4.0 for the Built Environment: Methodologies, Technologies and Skills*, pages 443–469. Springer International Publishing, Cham, 2022. ISBN 978-3-030-82430-3. doi: 10.1007/978-3-030-82430-3_19. URL https://doi.org/10.1007/978-3-030-82430-3_19. 1, 2

[6] Evan L. Denmark. *A technical analysis of photogrammetry with reality capture*. Thesis, Massachusetts Institute of Technology, 2020. URL https://dspace.mit.edu/handle/1721.1/129202. Accepted: 2021-01-06T18:34:23Z. 1, 4

[7] Alexander Tanchenko. Visual-PSNR measure of image quality. *Journal of Visual Communication and Image Representation*, 25(5):874–878, July 2014. ISSN 1047-3203. doi: 10.1016/j.jvcir.2014.01.008. URL https://www.sciencedirect.com/science/article/pii/S1047320314000091. 1, 3

[8] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks, December 2021. URL http://arxiv.org/abs/2112.05131. arXiv:2112.05131 [cs]. 1

[9] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based Neural Radiance Fields, March 2023. URL http://arxiv.org/abs/2201.08845. arXiv:2201.08845 [cs]. 1

[10] A. Basso, F. Condorelli, A. Giordano, S. Morena, and M. Perticarini. EVOLUTION OF RENDERING BASED ON RADIANCE FIELDS. THE PALERMO CASE STUDY FOR A COMPARISON BETWEEN NERF AND GAUSSIAN SPLATTING. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2-W4-2024:57–64, February 2024. ISSN 1682-1750, 2194-9034. doi: 10.5194/isprs-archives-XLVIII-2-W4-2024-57-2024. URL https://isprs-archives.copernicus.org/articles/XLVIII-2-W4-2024/57/2024/isprs-archives-XLVIII-2-W4-2024-57-2024.pdf. Publisher: Copernicus Publications. 1

[11] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 1, 4

[12] Xiangzhi Eric Wang and Zackary P. T. Sin. 3D Gaussian Model for Animation and Texturing, February 2024. URL http://arxiv.org/abs/2402.19441. arXiv:2402.19441 [cs]. 1, 2

[13] Tian-Xing Xu, Wenbo Hu, Yu-Kun Lai, Ying Shan, and Song-Hai Zhang. Texture-GS: Disentangling the Geometry and Texture for 3D Gaussian Splatting Editing, March 2024. URL http://arxiv.org/abs/2403.10050. arXiv:2403.10050 [cs]. 1, 2, 8

[14] RealityCapture - 3D Models from Photos and/or Laser Scans. URL https://www.capturingreality.com/. 2, 4, 5

[15] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. GaussianObject: Just Taking Four Images to Get A High-Quality 3D Object with Gaussian Splatting, February 2024. URL http://arxiv.org/abs/2402.10259. arXiv:2402.10259 [cs]. 2, 8

[16] RealityCapture Help. URL https://rchelp.capturingreality.com/en-US/tools/alignquality.htm. 3

[17] Aurora Monter-Pozos and Elizabeth González-Estrada. On testing the skew normal distribution by using shapiro–wilk test. *Journal of Computational and Applied Mathematics*, 440:115649, 2024. doi: 10.1016/j.cam.2023.115649. 3

[18] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591, 1965. doi: 10.2307/2333709. 3

[19] Tae Kyun Kim. T test as a parametric statistic. *Korean Journal of Anesthesiology*, 68(6):540, 2015. doi: 10.4097/kjae.2015.68.6.540. 3

[20] Thomas W. MacFarland and Jan M. Yates. *Mann–Whitney U Test*, pages 103–132. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30634-6. doi: 10.1007/978-3-319-30634-6_4. URL https://doi.org/10.1007/978-3-319-30634-6_4. 3

[21] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation, 2023. 4, 11

[22] Blender Foundation. blender.org - Home of the Blender project - Free and Open 3D Creation Software. URL https://www.blender.org/. 4, 5

[23] graphdeco-inria/gaussian-splatting, April 2024. URL https://github.com/graphdeco-inria/gaussian-splatting. original-date: 2023-07-04T07:51:38Z. 4, 5

A. Full Results

A.1. Used Results

Name	ATL	MRE	RC	PSNR-L	PSNR-N
anise-1					
120-GS	3.548	0.41	0.989	39.796	38.504
60-GS	3.414	0.409	0.989	39.821	38.5
30-GS	3.135	0.431	0.989	39.719	38.358
No-GS	3.506	0.332	0.989	39.884	38.638
anise-2					
120-GS	3.228	0.414	0.989	38.873	37.446
60-GS	3.241	0.413	0.989	38.893	37.428
30-GS	2.985	0.456	0.989	38.76	37.174
No-GS	3.358	0.351	0.989	38.923	37.663
anise-3					
120-GS	3.083	0.379	0.989	41.192	39.913
60-GS	3.066	0.383	0.989	41.184	39.886
30-GS	2.874	0.415	0.989	41.105	39.738
No-GS	3.227	0.32	0.989	41.241	40.038
antique-1					
120-GS	3.493	0.501	0.989	37.401	36.622
60-GS	3.351	0.496	0.989	37.395	36.593
30-GS	2.829	0.568	0.989	36.786	35.832
No-GS	3.885	0.46	0.989	37.314	37.443
antique-2					
120-GS	3.053	0.45	0.989	39.648	39.399
60-GS	2.857	0.466	0.989	40.037	39.39
30-GS	2.788	0.537	0.989	36.47	36.424
No-GS	3.123	0.45	0.989	41.897	41.707
antique-3					
120-GS	4.134	0.372	0.995	39.6	39.882
60-GS	4.092	0.374	0.995	39.583	39.697
30-GS	3.621	0.413	0.989	39.225	38.682
No-GS	4.47	0.284	1.0	39.581	40.924
apple-1					
120-GS	4.166	0.392	0.989	39.812	41.165
60-GS	3.935	0.415	0.989	39.767	40.854
30-GS	2.985	0.454	0.989	38.988	39.151
No-GS	4.378	0.269	0.989	39.917	42.238
apple-2					
120-GS	4.368	0.389	0.989	39.174	41.342
60-GS	4.122	0.42	0.989	39.157	41.131
30-GS	3.271	0.457	0.989	38.632	39.637
No-GS	4.504	0.29	0.995	39.242	42.039
apple-3					
120-GS	4.357	0.219	0.995	39.226	42.203
60-GS	4.164	0.388	0.989	39.929	42.177
30-GS	3.335	0.428	0.989	39.501	40.638
No-GS	4.541	0.282	0.995	39.959	42.936
backpack-1					
120-GS	3.147	0.461	0.989	39.495	38.547
60-GS	3.123	0.474	0.989	39.47	38.477
30-GS	2.939	0.448	0.989	39.489	38.317
No-GS	3.14	0.432	0.989	39.416	39.111
backpack-2					
120-GS	2.787	0.464	0.989	38.876	37.677
60-GS	2.767	0.48	0.989	38.836	37.651
30-GS	2.645	0.468	0.989	38.815	37.549
No-GS	2.736	0.448	0.989	38.834	38.156

backpack-3					
120-GS	3.09	0.328	0.989	38.016	38.479
60-GS	3.03	0.329	0.989	38.002	38.421
30-GS	2.819	0.355	0.989	37.808	38.086
No-GS	3.2	0.288	0.989	38.146	38.67
ball-1					
120-GS	3.927	0.309	0.995	38.649	39.423
60-GS	3.844	0.319	0.995	38.596	39.225
30-GS	3.363	0.337	0.989	38.299	38.487
No-GS	3.813	0.266	0.995	38.866	40.342
ball-2					
120-GS	3.912	0.345	0.989	37.658	39.639
60-GS	3.845	0.363	0.989	37.62	39.472
30-GS	3.178	0.381	0.989	37.117	37.844
No-GS	3.98	0.304	1.0	37.736	40.017
ball-3					
120-GS	3.502	0.376	0.989	36.454	37.667
60-GS	3.345	0.412	0.989	36.336	37.212
30-GS	2.769	0.445	0.989	35.874	36.137
No-GS	3.509	0.255	0.989	36.538	39.11
bamboo-1					
120-GS	3.95	0.328	0.989	44.966	45.651
60-GS	3.841	0.346	0.989	44.966	45.655
30-GS	3.107	0.352	0.989	44.823	45.183
No-GS	4.234	0.361	0.989	44.977	45.833
bamboo-2					
120-GS	3.542	0.32	0.989	43.972	44.821
60-GS	3.409	0.329	0.989	43.982	44.761
30-GS	2.86	0.35	0.989	43.924	44.331
No-GS	3.695	0.323	0.989	44.006	44.943
bamboo-3					
120-GS	3.984	0.309	0.989	44.473	45.301
60-GS	3.844	0.319	0.989	44.477	45.279
30-GS	3.298	0.346	0.989	44.28	44.632
No-GS	3.965	0.296	0.989	44.534	45.442
banana-1					
120-GS	4.179	0.332	0.989	40.517	41.687
60-GS	4.173	0.341	0.989	40.488	41.586
30-GS	3.768	0.366	0.989	40.372	41.074
No-GS	4.268	0.28	0.995	40.586	42.173
banana-2					
120-GS	4.4	0.343	0.989	40.788	41.573
60-GS	4.436	0.35	0.989	40.788	41.573
30-GS	3.999	0.36	0.989	40.559	40.89
No-GS	4.409	0.264	0.995	40.974	42.436
banana-3					
120-GS	4.163	0.356	0.989	40.592	41.284
60-GS	4.14	0.361	0.989	40.547	41.115
30-GS	3.721	0.374	0.989	40.297	40.523
No-GS	4.535	0.309	1.0	40.744	42.139
battery-1					
120-GS	3.73	0.321	0.989	40.554	41.445
60-GS	3.674	0.335	0.989	40.5	41.286
30-GS	3.351	0.357	0.989	40.24	40.775
No-GS	3.692	0.294	0.989	40.619	41.656
battery-2					
120-GS	3.395	0.399	0.989	39.117	39.81
60-GS	3.428	0.419	0.989	39.128	39.789
30-GS	3.164	0.428	0.989	39.117	39.436
No-GS	3.462	0.392	0.989	39.132	39.885
battery-3					
120-GS	3.55	0.444	0.989	38.795	39.315
60-GS	3.542	0.434	0.989	38.768	39.249
30-GS	3.195	0.445	0.989	38.693	38.596
No-GS	3.665	0.434	0.989	38.794	39.908
beauty-blender-1					
120-GS	3.103	0.326	0.989	39.909	41.598
60-GS	2.865	0.396	0.989	39.602	40.508
30-GS	2.476	0.411	0.989	39.153	39.517
No-GS	3.402	0.211	0.989	40.174	43.531
beauty-blender-2					
120-GS	2.89	0.279	0.989	40.438	40.085
60-GS	2.858	0.325	0.989	37.617	37.739
30-GS	2.5	0.385	0.989	39.851	39.426
No-GS	3.015	0.171	0.989	42.973	43.363
beauty-blender-3					
120-GS	3.344	0.325	0.989	37.728	39.036
60-GS	3.133	0.323	0.989	37.681	38.753
30-GS	2.778	0.336	0.989	37.321	37.859
No-GS	3.961	0.24	0.989	38.679	41.831

Table 4. Full Quantitative Results Per Category Per Measure Per Model

A.2. Unused Results

Name	ATL	MRE	RC	PSNR-L	PSNR-N
asparagus-1					
No-GS	3.330	0.386	180	47.577	46.950
asparagus-2					
No-GS	3.285	0.364	180	46.351	46.319
asparagus-3					
120-GS	3.527	0.384	180	45.950	45.621
No-GS	3.647	0.371	180	45.932	45.620

Table 5. Quantitative results from unused input models.

B. Model Names in OmniObject Dataset

OmniObject3D Dataset as it was accessed on 11-06-2024 [21].

Model Name in Paper	Model Name in Dataset
anise-1	anise_001
anise-2	anise_002
anise-3	anise_003
antique-1	antique_004
antique-2	antique_005
antique-3	antique_014
apple-1	apple_001
apple-2	apple_002
apple-3	apple_003
asparagus-1	asparagus_001
asparagus-2	asparagus_002
asparagus-3	asparagus_003
backpack-1	backpack_001
backpack-2	backpack_002
backpack-3	backpack_003
ball-1	ball_001
ball-2	ball_001
ball-3	ball_001
bamboo-1	bamboo_shoots_001
bamboo-2	bamboo_shoots_002
bamboo-3	bamboo_shoots_003
banana-1	banana_001
banana-2	banana_002
banana-3	banana_003
battery-1	battery_001
battery-2	battery_002
battery-3	battery_003
beauty-blender-1	beauty_blender_001
beauty-blender-2	beauty_blender_002
beauty-blender-3	beauty_blender_003

Table 6. Full list of names of models used from OmniObject3D, with their names as used in the paper and as listed in the dataset.

C. Polar Coordinates of Camera Positions

C.1. Gaussian Splatting Training Image Camera Coordinates

Camera Nr.	θ	φ
0	0.25pi	0.0pi
1	0.25pi	0.05pi
2	0.25pi	0.1pi
3	0.25pi	0.15pi
4	0.25pi	0.2pi
5	0.25pi	0.25pi
6	0.25pi	0.3pi
7	0.25pi	0.35pi
8	0.25pi	0.4pi
9	0.25pi	0.45pi
10	0.25pi	0.5pi
11	0.25pi	0.55pi
12	0.25pi	0.6pi
13	0.25pi	0.65pi
14	0.25pi	0.7pi
15	0.25pi	0.75pi
16	0.25pi	0.8pi
17	0.25pi	0.85pi
18	0.25pi	0.9pi
19	0.25pi	0.95pi
20	0.25pi	1.0pi
21	0.25pi	1.05pi
22	0.25pi	1.1pi
23	0.25pi	1.15pi
24	0.25pi	1.2pi
25	0.25pi	1.25pi
26	0.25pi	1.3pi
27	0.25pi	1.35pi
28	0.25pi	1.4pi
29	0.25pi	1.45pi
30	0.25pi	1.5pi
31	0.25pi	1.55pi
32	0.25pi	1.6pi
33	0.25pi	1.65pi
34	0.25pi	1.7pi
35	0.25pi	1.75pi
36	0.25pi	1.8pi
37	0.25pi	1.85pi
38	0.25pi	1.9pi
39	0.25pi	1.95pi

40	0.5pi	0.0pi
41	0.5pi	0.05pi
42	0.5pi	0.1pi
43	0.5pi	0.15pi
44	0.5pi	0.2pi
45	0.5pi	0.25pi
46	0.5pi	0.3pi
47	0.5pi	0.35pi
48	0.5pi	0.4pi
49	0.5pi	0.45pi
50	0.5pi	0.5pi
51	0.5pi	0.55pi
52	0.5pi	0.6pi
53	0.5pi	0.65pi
54	0.5pi	0.7pi
55	0.5pi	0.75pi
56	0.5pi	0.8pi
57	0.5pi	0.85pi
58	0.5pi	0.9pi
59	0.5pi	0.95pi
60	0.5pi	1.0pi
61	0.5pi	1.05pi
62	0.5pi	1.1pi
63	0.5pi	1.15pi
64	0.5pi	1.2pi
65	0.5pi	1.25pi
66	0.5pi	1.3pi
67	0.5pi	1.35pi
68	0.5pi	1.4pi
69	0.5pi	1.45pi
70	0.5pi	1.5pi
71	0.5pi	1.55pi
72	0.5pi	1.6pi
73	0.5pi	1.65pi
74	0.5pi	1.7pi
75	0.5pi	1.75pi
76	0.5pi	1.8pi
77	0.5pi	1.85pi
78	0.5pi	1.9pi
79	0.5pi	1.95pi
80	0.75pi	0.0pi
81	0.75pi	0.05pi
82	0.75pi	0.1pi
83	0.75pi	0.15pi
84	0.75pi	0.2pi
85	0.75pi	0.25pi
86	0.75pi	0.3pi
87	0.75pi	0.35pi
88	0.75pi	0.4pi
89	0.75pi	0.45pi
90	0.75pi	0.5pi
91	0.75pi	0.55pi
92	0.75pi	0.6pi
93	0.75pi	0.65pi
94	0.75pi	0.7pi
95	0.75pi	0.75pi
96	0.75pi	0.8pi
97	0.75pi	0.85pi
98	0.75pi	0.9pi
99	0.75pi	0.95pi
100	0.75pi	1.0pi
101	0.75pi	1.05pi
102	0.75pi	1.1pi
103	0.75pi	1.15pi
104	0.75pi	1.2pi
105	0.75pi	1.25pi
106	0.75pi	1.3pi
107	0.75pi	1.35pi
108	0.75pi	1.4pi
109	0.75pi	1.45pi
110	0.75pi	1.5pi
111	0.75pi	1.55pi
112	0.75pi	1.6pi
113	0.75pi	1.65pi
114	0.75pi	1.7pi
115	0.75pi	1.75pi
116	0.75pi	1.8pi
117	0.75pi	1.85pi
118	0.75pi	1.9pi
119	0.75pi	1.95pi

Table 7. θ and φ coordinate values for camera positions used to train GS scenes. r is always equal to 1.

C.2. Photogrammetry Input Image Camera Coordinates

Camera Nr.	θ	φ
0	0.2pi	0.0pi
1	0.2pi	0.044pi
2	0.2pi	0.089pi
3	0.2pi	0.133pi
4	0.2pi	0.178pi
5	0.2pi	0.222pi
6	0.2pi	0.267pi
7	0.2pi	0.311pi
8	0.2pi	0.356pi
9	0.2pi	0.4pi
10	0.2pi	0.444pi
11	0.2pi	0.489pi
12	0.2pi	0.533pi
13	0.2pi	0.578pi
14	0.2pi	0.622pi
15	0.2pi	0.667pi
16	0.2pi	0.711pi
17	0.2pi	0.756pi
18	0.2pi	0.8pi
19	0.2pi	0.844pi
20	0.2pi	0.889pi
21	0.2pi	0.933pi
22	0.2pi	0.978pi
23	0.2pi	1.022pi
24	0.2pi	1.067pi
25	0.2pi	1.111pi
26	0.2pi	1.156pi
27	0.2pi	1.2pi
28	0.2pi	1.244pi
29	0.2pi	1.289pi
30	0.2pi	1.333pi
31	0.2pi	1.378pi
32	0.2pi	1.422pi
33	0.2pi	1.467pi
34	0.2pi	1.511pi
35	0.2pi	1.556pi
36	0.2pi	1.6pi
37	0.2pi	1.644pi
38	0.2pi	1.689pi
39	0.2pi	1.733pi
40	0.2pi	1.778pi
41	0.2pi	1.822pi
42	0.2pi	1.867pi
43	0.2pi	1.911pi
44	0.2pi	1.956pi
45	0.4pi	0.0pi
46	0.4pi	0.044pi
47	0.4pi	0.089pi
48	0.4pi	0.133pi
49	0.4pi	0.178pi
50	0.4pi	0.222pi
51	0.4pi	0.267pi
52	0.4pi	0.311pi
53	0.4pi	0.356pi
54	0.4pi	0.4pi
55	0.4pi	0.444pi
56	0.4pi	0.489pi
57	0.4pi	0.533pi
58	0.4pi	0.578pi
59	0.4pi	0.622pi
60	0.4pi	0.667pi
61	0.4pi	0.711pi
62	0.4pi	0.756pi
63	0.4pi	0.8pi
64	0.4pi	0.844pi
65	0.4pi	0.889pi
66	0.4pi	0.933pi
67	0.4pi	0.978pi
68	0.4pi	1.022pi
69	0.4pi	1.067pi
70	0.4pi	1.111pi
71	0.4pi	1.156pi
72	0.4pi	1.2pi
73	0.4pi	1.244pi
74	0.4pi	1.289pi
75	0.4pi	1.333pi
76	0.4pi	1.378pi
77	0.4pi	1.422pi
78	0.4pi	1.467pi
79	0.4pi	1.511pi
80	0.4pi	1.556pi
81	0.4pi	1.6pi
82	0.4pi	1.644pi
83	0.4pi	1.689pi
84	0.4pi	1.733pi
85	0.4pi	1.778pi
86	0.4pi	1.822pi
87	0.4pi	1.867pi
88	0.4pi	1.911pi
89	0.4pi	1.956pi

90	0.6pi	0.0pi
91	0.6pi	0.044pi
92	0.6pi	0.089pi
93	0.6pi	0.133pi
94	0.6pi	0.178pi
95	0.6pi	0.222pi
96	0.6pi	0.267pi
97	0.6pi	0.311pi
98	0.6pi	0.356pi
99	0.6pi	0.4pi
100	0.6pi	0.444pi
101	0.6pi	0.489pi
102	0.6pi	0.533pi
103	0.6pi	0.578pi
104	0.6pi	0.622pi
105	0.6pi	0.667pi
106	0.6pi	0.711pi
107	0.6pi	0.756pi
108	0.6pi	0.8pi
109	0.6pi	0.844pi
110	0.6pi	0.889pi
111	0.6pi	0.933pi
112	0.6pi	0.978pi
113	0.6pi	1.022pi
114	0.6pi	1.067pi
115	0.6pi	1.111pi
116	0.6pi	1.156pi
117	0.6pi	1.2pi
118	0.6pi	1.244pi
119	0.6pi	1.289pi
120	0.6pi	1.333pi
121	0.6pi	1.378pi
122	0.6pi	1.422pi
123	0.6pi	1.467pi
124	0.6pi	1.511pi
125	0.6pi	1.556pi
126	0.6pi	1.6pi
127	0.6pi	1.644pi
128	0.6pi	1.689pi
129	0.6pi	1.733pi
130	0.6pi	1.778pi
131	0.6pi	1.822pi
132	0.6pi	1.867pi
133	0.6pi	1.911pi
134	0.6pi	1.956pi
135	0.8pi	0.0pi
136	0.8pi	0.044pi
137	0.8pi	0.089pi
138	0.8pi	0.133pi
139	0.8pi	0.178pi
140	0.8pi	0.222pi
141	0.8pi	0.267pi
142	0.8pi	0.311pi
143	0.8pi	0.356pi
144	0.8pi	0.4pi
145	0.8pi	0.444pi
146	0.8pi	0.489pi
147	0.8pi	0.533pi
148	0.8pi	0.578pi
149	0.8pi	0.622pi
150	0.8pi	0.667pi
151	0.8pi	0.711pi
152	0.8pi	0.756pi
153	0.8pi	0.8pi
154	0.8pi	0.844pi
155	0.8pi	0.889pi
156	0.8pi	0.933pi
157	0.8pi	0.978pi
158	0.8pi	1.022pi
159	0.8pi	1.067pi
160	0.8pi	1.111pi
161	0.8pi	1.156pi
162	0.8pi	1.2pi
163	0.8pi	1.244pi
164	0.8pi	1.289pi
165	0.8pi	1.333pi
166	0.8pi	1.378pi
167	0.8pi	1.422pi
168	0.8pi	1.467pi
169	0.8pi	1.511pi
170	0.8pi	1.556pi
171	0.8pi	1.6pi
172	0.8pi	1.644pi
173	0.8pi	1.689pi
174	0.8pi	1.733pi
175	0.8pi	1.778pi
176	0.8pi	1.822pi
177	0.8pi	1.867pi
178	0.8pi	1.911pi
179	0.8pi	1.956pi

Table 8. θ and φ coordinate values for camera positions used to generate a 3D model with PG. r is always equal to 1.

D. Repository

https://doi.org/10.4121/d8970e29-b0da-406e-ac69-c16df992b322