TU WIEN Informatics

Gaussians on a budget

Untersuchung subjektiver und objektiver Indikatoren für Gaussian Splatting Heuristiken

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Paul Erpenstein, B.Sc.
Matrikelnummer 12107369

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assistant Prof. Doctor Pedro Hermosilla Casajus

Wien, 28. Februar 2025

Paul Erpenstein	Pedro Hermosilla Casajus

TU WIEN Informatics

Gaussians on a budget

Surveying subjective and objective indicators for Gaussian Splatting heuristics

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Paul Erpenstein, B.Sc.
Registration Number 12107369

to the Faculty of Informatics

at the TU Wien

Advisor: Assistant Prof. Doctor Pedro Hermosilla Casajus

Vienna, February 28, 2025

Paul Erpenstein	Pedro Hermosilla Casajus

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Paul Erpenstein, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 28. Februar 2025

Paul Erpenstein

Danksagung

Mein besonderer Dank gilt dem Team des Vienna Scientific Cluster für ihren kompetenten technischen Support und ihren Einsatz, hochperformante Rechenressourcen für die Wissenschaft bereitzustellen.

Ebenso danke ich meinen Freunden Hendrik Spitzenpfeil und Jan Ölhoff für die inspirierenden User-Study-Sessions, die mich ab der Mitte meines Projekts sehr motivierten.

Ein herzliches Dankeschön geht auch an die Mitglieder der Fachschaft Wirtschaftsinformatik für das Bereitstellen großartiger Räumlichkeiten und die offene, unterstützende Gemeinschaft.

Eine Evaluation ist nichts ohne tatkräftige Teilnehmerinnen und Teilnehmer. Mein Dank gilt also jedem der Mitwirkenden - eure Hilfe war essenziell für den Erfolg dieser Arbeit.

Ich danke meinen Eltern für ihr Vertrauen und ihre Unterstützung dieses Studium zu ermöglichen.

Letztlich möchte ich mich bei meiner Partnerin Lidia bedanken, welche mir sowohl bei der Organisation der User-Study als auch durch ihre Geduld und Unterstützung während der gesamten Zeit geholfen hat.

Kurzfassung

Novel View Synthesis (NVS) ist ein reges Forschungsfeld mit dem Ziel, ungesehene Bilder einer Szene aus einer begrenzten Anzahl von Ursprungsbildern zu generieren. Dazu gibt es verschiedene Ansätze. Die relativ neue Methode des 3D Gaussian Splatting (GS) ist ein besonders vielversprechender Lösungsansatz, der visuelle Qualität mit schnellem Rendering kombiniert. Ein Problem besteht darin, dass die Heuristik, welche GS-Modelle erstellt, stark von der Modellgröße abhängig ist, um die geometrische Beschaffenheit der Szene originalgetreu einzufangen. Diese Arbeit untersucht GS-Optimierungsverfahren, um kompaktere Modelle zu erstellen. Es werden sechs bestehende Optimierungsverfahren erweitert, indem eine Begrenzungsstrategie eingeführt wird, die strikte Größenbeschränkungen erlaubt. Eine Evaluation der sechs Methoden sowie des ursprünglichen Verfahrens wurde durchgeführt. Subjektive und objektive Qualitätsmetriken sowie Verhaltensindikatoren während des Trainings werden analysiert. Drei der sieben getesteten Methoden führen zu signifikant verbesserten Ergebnissen im Vergleich zum Ursprungsverfahren. Quantitative und qualitative Indikatoren zeigen, dass 3D Gaussian Splatting as Markov Chain Monte Carlo (3DGS MCMC) von Kheradmand et al. Modelle mit minimalen Fehlern und hohem Detailgrad erstellt, selbst unter starker Beschränkung der Modellgröße. Eine gemeinsame Analyse subjektiver und objektiver Qualitätsmetriken zeigt zudem, dass objektive Metriken nur dann mit der wahrgenommenen Qualität übereinstimmen, wenn die Kamerapfade des Datensatzes mit den Kameraperspektiven der Studienteilnehmenden übereinstimmen.

Abstract

Novel View Synthesis (NVS) is an active field of research with the aim of generating unseen views of a scene given a limited set of initial images. There are a number of different approaches for tackling this problem. The relatively novel approach of 3D Gaussian Splatting (GS) has garnered immense interest as of late. Its popularity stems from the fact that it combines visual fidelity with great real-time rendering performance. The problem is that the heuristic solution that creates the GS models relies on very large models in order to accurately approximate the scene geometry. The aim of this work is to identify and compare approaches, which focus on creating compact GS models. Six existing methods were extended with a capping strategy that allows for improvements even under strict size constraints. A comparative evaluation of the six optimization techniques and the original procedure was conducted. The evaluation combines subjective user-study results with objective quality metrics and an analysis of model behavior during training. Three of the seven tested approaches significantly outperform the baseline under constrained conditions. Based on quantitative and qualitative indicators 3D Gaussian Splatting as Markov Chain Monte Carlo (3DGS MCMC) by Kheradmand et al. [KRS+24] yields models with the least errors and best detail resolution under strict size constraints. A joint analysis of subjective and objective measures was conducted. It indicates that objective measures tend to correspond to perceived quality only when dataset camera paths are aligned with views experienced by study participants.

Contents

CHAPTER 1

Introduction

Early NVS approaches date back to the 1990s [Fau94, SVDV96, SGHS98, LH96, GGSC96]. The goal is to generate unseen images of a scene from new viewpoints based on a limited set of *reference views*. These approaches achieve 3D visualization in an unconventional way. Instead of relying on geometric primitives in order to model the scene content, they use the structure of captured images to derive new views.

Early work includes the approach by Laveau and Faugeras, where new views are generated by interpolating between existing ones [Fau94]. Not much later, Levoy and Hanrahan use a four-dimensional function that describes the light passing through a scene. They call the process of using this function to create new images *Light Field Rendering* [LH96]. Recently, an astonishingly similar method has breathed new life into this area of research. Neural Radiance Fields (NeRF) [MST+20] uses a 5-dimensional function to describe the radiance in a scene. As the name suggests, the method uses a neural network. NeRF leverages the strength of machine learning in order to optimize the network based on the reference images. The technique represents a possible departure from traditional 3D rendering approaches. The problem is that, in comparison to conventional rasterization methods, the rendering is extremely slow and expensive. So what if we could combine the strengths of rasterization with those of machine learning? This is where Gaussian Splatting (GS) [KKLD23] comes in. GS models the objects in a scene using a set of geometric primitives. The primitives are often referred to as Splats or Gaussians. Gaussians are placed in the scene by using an iterative machine learning scheme, very similar to that of NeRF. This yields visually impressive results while maintaining acceptable performance on modern systems [KKLD23].

In the original release of GS, Kerbl et al. [KKLD23] propose a heuristic solution in order to fit the 3D representation based on the reference images. They call this solution *Densification*. As the name suggests, Densification increases the number of primitives as the iterations progress. They find that this strategy enables fine-grained detail and also reduces visual artifacts.

More primitives also increase the size of the GS model. Large 3D models are problematic, especially when there is an inverse relationship between model compactness and visual fidelity. The research community is actively addressing this problem, as evidenced by the numerous papers published on the subject. These solutions are diverse in their approaches. Some revise the optimization procedure [HBZN24, KRS+24, BPK24], others focus on pruning unimportant primitives [FW24, GGS23], and some employ smart compression schemes in order to reduce the model size [CWL+24, NSW24, YYX+24].

The abundance of research regarding GS shows that there are plenty of different ways in which the method can be modified and extended. This work will focus on the refinement of the optimization procedure. Gaussians can be placed in the scene arbitrarily. New densification techniques result in very different models [YCH+23, KRS+24, YSG24, LLD+24, CLY+24, FW24]. But which of the proposed approaches handles the placement the best? Previous work often focuses on models that are fully converged and have maximal detail [YCH+23, YSG24, LLD+24, CLY+24, ZZX+24]. It is rare to see a comparison based on constraied model size [KRS+24]. This clearly biases previous results towards optimization techniques that benefit from a large model size. But what if that doesn't tell us the whole story? What if some techniques only shine in scenarios in which compactness matters?

The goal of this work is to identify and compare techniques that are likely to perform well, when the number of Gaussians is limited. To this end, a review of the literature, followed by a comparative evaluation, is conducted. Rather than comparing static renders or videos, users will have the ability to see, interact with, and compare the models. This means that the evaluation will yield more insight into what the actual outcome of an optimization technique looks like. The aim of the evaluation is to answer the research question *"Which optimization procedure delivers the best visual fidelity when the number of splats is constrained?"*. Training multiple models based on different sets of reference images also results in objective indicators of visual quality. These are indicators that can be used to determine model quality without having to involve real users. In order to understand the relationship between the objective indicators and user perception, the quality measures will be analyzed jointly. This will answer the question *Do objective and subjective image quality assessment measures align across different techniques and model size restrictions?*

In order to make informed comparisons, model size budgets are established. Multiple diverse reference image sets, also referred to as scenes, are chosen as training data. Based on the size budgets, scenes, and optimization techniques, models are trained. To display the models, a custom evaluation tool is implemented. The tool is designed based on the evaluation approach prevalent in the literature and is available over the web. The visual quality of the presented models is computed and interpreted in a comprehensive data analysis. Subjective and objective indicators are contrasted to determine how well the guiding mathematical principles actually align with perceived visual fidelity. Based on the insights, a number of recommendations for future techniques as well as evaluation procedures are given.

1.1 Contributions

This research will shine a light on the potential of different GS optimization techniques. The contributions to the current state of the research are as follows:

- A ranking of the chosen optimization techniques.
- Concise insights about the behavior, strengths, and weaknesses of the optimization techniques in different circumstances. The insights are presented to contextualize the overall rankings.
- A correlation analysis between the user judgments and the objective quality measures. This is presented to showcase the difference between the perception of interactive 3D models and objective metrics.
- Multiple extensions to state-of-the-art optimization procedures to constrain model size while enabling progressive optimization.

1.2 Structure of the Thesis

In this section, the structure of the thesis is explained. Chapter 2 details the relevant literature and theoretical background. Firstly, the topic of NVS is expanded upon. The goal is to give a sense of the scientific context in which GS is situated. Afterwards, different NVS approaches are explained. These include Structure-from-Motion (SfM), Point-based Splatting, and NeRF. Finally, the section will move on to an in-depth explanation of GS and its accompanying literature. The purpose of the section will be to give a general sense of the capabilities of GS, but also inform the reader about research regarding GS model compactness. Lastly, the science of Image Quality Assessment (IQA) will be outlined as it relates to the development and evaluation of NVS techniques.

In Chapter 3, the methodology used to answer the research question will be laid out. The first section justifies the approach based on other work in the field. Next, the framework of this study is outlined. Then the densification methods, which were chosen as the focus of this evaluation, are presented. Reasons for their inclusion, as well as exclusion criteria, are given. The last two sections describe the training setup and the implementation details of the software components.

Chapter 4 presents the results in detail. The chapter is broken up into the different data sources that serve as indicators for model behavior. Firstly, the subjective user study results are laid out. Afterwards, the objective quality metrics are presented. Then a joint analysis of the two different quality indicators is conducted to come to an overall conclusion. The chapter is concluded with a discussion of the model behavior during training.

Chapter 5 deals with discussing and interpreting the results. In order to understand the abstract statistics presented in Chapter 4, actual renders of the models will be

discussed. This gives context as to what structures are actually experienced by study participants and what informed their ratings. Afterwards, the limitations of the study will be presented.

Chapter 6 summarizes the results. It provides reflection about the work and the success of the project. Recommendations and possible future research directions are given.

CHAPTER 2

Related Work

This chapter provides an overview of work relevant to this study. It starts with the evolution of NVS and its connection to Multi-View Stereo (MVS). Afterwards, different scene representation techniques are examined, leading to an in-depth discussion of GS, its extensions, and methods for creating more compact GS models. Then IQA techniques and their role in evaluating NVS models are reviewed. Finally, some studies similar to this one are presented.

2.1 The Evolution of View Synthesis

The research problem of NVS deals with generating images of a scene from previously unseen viewpoints [Sch99]. There are a number of different approaches to tackle this problem. This section will explore some of these techniques and show how GS and its competitors evolved. A brief review of the emergence of these approaches will be given, to explain the basic ideas and mechanics of NVS solutions.

2.1.1 Early Approaches

Before the 2000s, the field of Image-based Rendering (IBR) received much attention [Kan98]. Scharstein [Sch99] describes IBR as a group of techniques that leverage information contained in images to render novel views. They require camera parameters to be known ahead of time. He states that IBR is closely linked to *Stereo Vision*, which deals with inferring the scene content, particularly the depth of each pixel, from images. Depth information is used by many IBR techniques to inform the creation of novel views [Sch99]. The field distinguishes itself from NVS by its scale and requirements. The number of images is most often limited to only two [Kan98].

The following paragraphs will present a few of the early IBR approaches. This will serve as valuable context for how modern NVS developed and which components are still relevant today.

Geometric Approaches Some early IBR techniques forgo an explicit scene representation and only use the information contained in reference images [Kan98]. These geometric approaches leverage existing relations between the images to cleverly interpolate between them [Fau94, SVDV96]. Laveau and Faugeras [Fau94] propose an approach that generates unseen views based on as few as two images of a scene. It uses the properties of epipolar geometry to reproject the existing image information into a new view. Their approach is inherently limited, as it assumes epipolar properties to be known beforehand. The novel camera positions cannot be chosen freely, as they must match view constraints imposed by the original images.

Depth-based Approaches Depth images contain the distance to the camera for each pixel. When an image with depth information is present, it can be used as a 3D representation enabling reprojecting the image from different viewpoints. Shade et al. [SGHS98] introduced the so-called Layered Depth Images (LDI). These extend simple depth images by introducing multiple depth and color values per pixel and can be constructed using a set of depth images. They are essentially a pixel-based 3D model of a scene and can be rendered using a Splatting approach.

Light Field Rendering Proposed by Adelson and Bergen [LM91], the plenoptic function describes the light passing through a scene. The function's name is derived from the words *plenus*, which means full or complete, and *optic*, referring to its purpose of describing the behavior of light.

$$P : (x, y, z, \theta, \phi, \lambda) \rightarrow p \tag{2.1}$$

Equation 2.1 shows the plenoptic function assigning an intensity of light p to every position (x, y, z), light ray direction (θ, ϕ), and wavelength λ [LM91]. An image can be seen as an expression of the plenoptic function. It represents the light captured at the camera's position. The rays hitting the camera lens are represented as pixels, storing the light information in the red, green, and blue channels.

The approaches of *Light Field Rendering* by Levoy and Hanrahan [LH96], as well as the *Lumigraph* by Gortler et al. [GGSC96], use this functional representation of light for the generation of new images. For these approaches to work, all cameras must lie on a plane, facing the same direction. This grid of images allows for a special parametrization of the plenoptic function. The methods resample and interpolate the previously captured light rays to render new views.

Accurate camera poses and properties are a strict prerequisite for most IBR approaches. This imposes a severe challenge to the applicability of IBR techniques. Fortunately, a promising solution for this problem was just over the horizon.

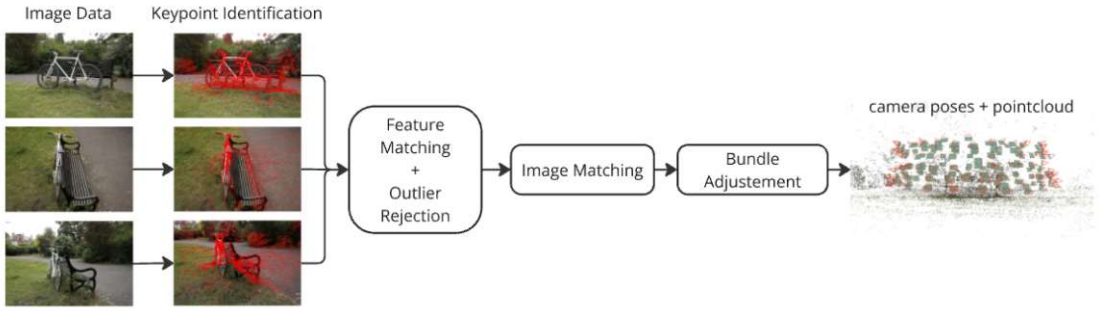

Figure 2.1: High-level overview of the SfM pipeline

2.1.2 Structure-from-Motion

SfM is related to Stereo Vision. In contrast to Stereo Vision, SfM is not reliant on calibrated cameras. Given a set of images taken from arbitrary viewpoints in a scene, it provides a solution to jointly estimate the camera poses. Figure 2.1 shows an abstract overview of the SfM pipeline.

The SfM Pipeline

Brown and Lowe's [BL05] work serves as a great example of a typical SfM pipeline. Using sets of images, they can extract accurate 3D representations of the scene contents. An essential building block of their technique is the Scale Invariant Feature Transform (SIFT) [Low04]. Given an image, SIFT extracts a set of characteristic keypoints. These points come with feature vectors that serve as descriptors. Descriptors aren't unique to a keypoint, but they encapsulate its characteristics, such as orientation and local image topography.

Brown and Lowe [BL05] then use the keypoints to establish relationships between the images. They argue that views with related keypoints are likely similar in terms of camera parameters. This enables the so-called *feature matching* step, in which images with common points are determined. Afterwards, images are sequentially integrated into the 3D representation of the scene. This process is called *image matching* or *image registration*. By registering new images, the relative camera parameters can be estimated using *triangulation*. This includes both the camera positions and the 3D coordinates of the keypoints.

Figure 2.2 shows example images with keypoints that were matched across different views. It can be seen that the points outline the shape of three-dimensional objects in the scene.

As a final step, Brown and Lowe's pipeline [BL05] uses *bundle adjustment* to refine the camera parameters and 3D points. Bundle adjustment works by minimizing the reprojection error, which can be seen in equation 2.2.

$$e = \sum_{i \in I} \sum_{j \in X(i)} f(k_{ij} - p_{ij})^2 \tag{2.2}$$

Figure 2.2: Matched keypoints in image space ("bicycle" scene from Mip-NeRF 360 [BMV+22])

The function describes the total error, that is, the difference between each initial keypoint k_{ij} and its reprojected 3D counterpart p_{ij} in image space. In this formula, $f(x)$ is some robust error function, while I is the set of all images and $X(i)$ is the set of points that can be projected into image i. By optimizing the reprojection of points into image space over all registered views, the robustness and accuracy of Brown and Lowe's [BL05] SfM pipeline is increased. It results in a 3D point cloud approximating the scene geometry and calibrated cameras for each of the reference images. The error is minimized using Levenberg-Marquardt [Lev44].

The Emergence of COLMAP

An example of using SfM at scale is that by Snavely et al. [SSS06]. It comes in the form of their well-known *Photo Tourism* project, which virtualizes popular tourist destinations using photos found on the internet. Building on top of the work by Brown and Lowe, they estimate camera positions and sparse point-cloud representations of the scene. Their system allows users to interactively switch between different images. It uses image morphing to allow smooth camera transitions between the different viewpoints. The *Photo Tourism* project served as a basis for the release of *Bundler*. It is one of the earlier open-source SfM software packages [Sna25].

The open-source SfM scene later saw the arrival of Schönberger et al.'s COLMAP [SF16]. It provides major gains in terms of robustness, accuracy, and scalability. The software does not completely reinvent SfM pipelines, but instead enhances the pipeline across all of its various components. The improvements include the following:

- Enriching the feature matching with a geometric verification strategy, which stabilizes the initial 3D model.
- Improving image registration via a next-best-view selection that maximizes robustness.
- Increasing the performance and robustness of 3D point triangulation.

Figure 2.3: Screenshot of a sparse COLMAP point cloud with calibrated cameras based on the Mip-NeRF 360 "bicycle" scene [BMV+22].

- An iterative bundle adjustment scheme that runs throughout the image registration process, which mitigates accumulated errors.

COLMAP represented a major step forward for general-purpose SfM pipelines [SF16]. The software serves as an important initialization step for many modern NVS approaches [FLK+23]. Figure 2.3 shows a sparse point cloud created by COLMAP, which can be used as an input to train a NeRF or GS model.

COLMAP Extensions

In recent years, further extensions of COLMAP have been proposed. DeTone et al. [DMR18] devise a neural alternative to the SIFT keypoint extractor. Their approach, called SuperPoint, leverages a Convolutional Neural Network (CNN) to identify keypoints and create descriptors. In comparison to other state-of-the-art methods, SuperPoint results in denser keypoint samples and more correct matches across different images.

Sun et al. [SSW+21] propose a feature matching approach that involves using a vision transformer model [DBK+21]. Their technique processes image pairs in unison. The images are fed to a CNN, followed by self-attention and finally a cross-attention layer. This results in a keypoint estimation that is conditioned on both images together. Their architecture enables the detection of keypoints in low-texture regions, which is challenging for conventional feature matchers.

There is also increased interest to alleviate the dependency of NVS methods on COLMAP. Lin et al. [LMTL21] integrate Bundle Adjustment into NeRF, which jointly learns a 3D representation of a scene, while refining imperfect or even registering completely unknown cameras. Fu et al. [FLK+23] leverage the point cloud nature of Gaussian Splat Models as well as the continuity in video sequences to completely forgo the dependency on pre-calibrated cameras.

COLMAP remains an essential component of most NVS approaches. This means that an understanding of its functionality is core to understanding how downstream applications behave.

2.1.3 Neural Radiance Fields

NeRF was introduced by Mildenhall et al. [MST+20] and represents a continuation of plenoptic approaches like light fields. In comparison to the plenoptic function from equation 2.1, NeRF relies on a simplified function to represent the light passing through space. This formula can be seen in equation 2.3.

$$F_\Theta : (x, y, z, \theta, \phi) \rightarrow (r, g, b, \sigma) \tag{2.3}$$

The function F is parameterized by Θ. It maps 3D points (x, y, z) and viewing directions (θ, ϕ) to r, g, b colors and a volume density σ. This means that the function defines how a point in space looks, when viewing it from a given direction. The formulation allows for the same point to look differently from different viewing angles. This is intended to allow for view-dependent effects like reflections. The volume density σ describes a point's opacity. This allows for a volume of space to be fully or partially transparent[MST+20].

NeRF Training

The novelty of NeRF stems from the fact that it combines the emerging field of *Neural Rendering* with that of NVS. Neural Rendering is the practice of using Deep Learning Models to generate images or video [TFT+20]. NeRF's plenoptic function F_Θ is implemented using a Deep Neural Network [MST+20].

Figure 2.4 shows the training process of NeRF. The most important part of the rendering process is that it is differentiable. In simple terms, this means that creating an image using NeRF is a two-way street. Rendered images depend on the weights of the network Θ and in turn, gradients can be propagated from the output back to the weights. This type of image synthesis is called a differentiable rendering algorithm [TFT+20].

To render an image, NeRF determines each pixels' color, by casting a ray. For a given ray, points are sampled from the MLP and accumulated to retrieve the final color [MST+20]. The mathematical formulation of this volume integration process can be seen in equation 2.4.

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt \quad \text{where} \quad T(t) = exp\left(-\int_{t_n}^{t} \sigma(r(s))ds\right) \tag{2.4}$$

The color of a ray $C(r)$ is computed by integrating over the color distribution between the near (t_n) and far (t_f) bounds of the ray. At the heart of the formula lies the color computation $\sigma(r(t))c(r(t), d)$. In practice, these values are queried from the neural network. The function $r(t) = o + td$ yields the current point along the ray. $T(t)$ represents the already accumulated opacity by previous sample points [MST+20].

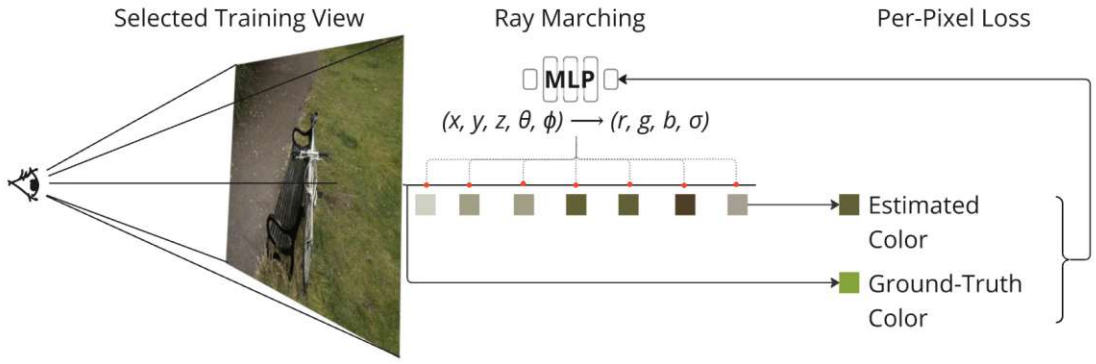

Figure 2.4: NeRF Training Process overview: The figure depicts the ray-based training process of NeRF. For a given training view, a subset of rays are selected. The color of a ray is estimated using ray marching. The Multilayer Perceptron (MLP) is called for a number of points along the ray. The color is accumulated and the loss is computed based on the ray's ground truth color.

The problem with NeRF lies in its reliance on ray marching and multiple neural network calls per rendered pixel. This makes view synthesis prohibitively expensive [RSV+23, CW24]. At the current moment in time, even cutting-edge hardware struggles with providing an interactive experience when rendering NeRFs [KKLD23].

NeRF Extensions

The research community recognizes that the render cost is a limiting factor of NeRF, which is illustrated by the many solutions have been proposed to tackle this problem [TFT+20, LGL+21, HSM+21, DHR+24, YLT+21, RSV+23, WLW+23].

Yu et al. [YLT+21] completely forgo any Neural Network calls during render time. They bake all of the NeRF's information into an octree. In order to achieve view-dependent effects, they use spherical harmonics coefficients. The resulting rendering pipeline is compatible with modern graphics backends like WebGL.

Wu et al. [WLW+23] propose NeRDF, which is a form of knowledge distillation for NeRFs. They train a network that predicts the radiance distributions along rays, based on a teacher NeRF model. This means that each ray only requires a single network call. This architecture yields an approximately 254 times speed-up.

Many spatial neural network approaches use secondary data structures to offload some of the rendering cost [CLI+20, JSM+20, YFKT+21, TLY+21, MESK22]. While this increases the rendering speed, it also leads to a larger memory footprint [MESK22]. Müller et al.'s [MESK22] Instant NGP uses a sparse multiresolution hash structure to store learned spatial features. Their technique can be used to tackle different tasks like Signed Distance Function (SDF), Neural Radiance Caching and NeRF. During rendering, the features are queried from the hash table and are fed to a much shallower neural

network than the one which is required by the original NeRF. This results in a significant speed-up of the rendering process.

Another example of this paradigm is Hedman et al.'s [HSM+21] Sparse Neural Radiance Grid (SNeRG). The method drastically reduces the render time, such that interactive performance can be achieved on commodity hardware. They accomplish this by baking the NeRF outputs into a sparse voxel grid. At render time, only a single network call per ray is executed to account for view-dependent effects. Multiple strategies to reduce the baked file size, like encouraging voxel-grid sparsity, smaller network outputs, and compression, are utilized.

Reiser et al.'s [RSV+23] recently proposed Memory Efficient Radiance Fields (MERF) shows that the memory footprint of large scenes can be reduced significantly without having to sacrifice visual quality. Their method achieves real-time rendering performance by building on top of SNeRG's sparse voxel grids. A space contraction scheme and a coarse-to-fine parametrization strategy further reduce the memory footprint.

MERF was then extended by Reiser et al.'s [RSV+23] in the form of Streamable MERFs (SMERF). The method enables the capture of large scenes, by using a scene partitioning scheme. It inherits the memory efficiency of MERF, while allowing for new scene content to be continuously streamed in at render time. At the same time, the method retains the real-time performance of SNeRG.

The current state-of-the-art of NeRF shows that the technique advances rapidly. Its weaknesses are addressed effectively via ongoing research. This means that NeRF remains competitive with GS. There is a large degree of "cross-pollination" going on between the two areas of research. This will become increasingly apparent in the next section.

2.2 Gaussian Splatting

Naturally, the technology of GS [KKLD23] is at the heart of this thesis. This section will review the different scientific and technical components of this NVS approach. First, a short overview of how Gaussians evolved from point representations will be given. Then, the model definition, rendering, and training paradigm are presented. The last two parts of this section will focus on novel developments in the field and enhancements of the approach.

2.2.1 From Points to Gaussians

GS can be seen as an extended form of point-cloud rendering. One of the earliest examples of point-cloud rendering was proposed by Grossmann and Dally [GD98]. They refer to the technique as *Point Sample Rendering*. To visualize the point samples, they select a subsample of the surface points and display them as single pixels. They point out the apparent weakness of this visualization technique, which is the visible gaps between the points. To this end, the authors also describe the process of splatting. It is the

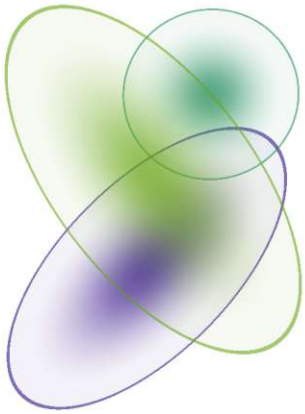

Figure 2.5: A simple model consisting of 3 points. Each point is rendered using Elliptical Weighted Average (EWA) [ZPvBG02] splatting, with unique Gaussian parameters. The rings represent the outer boundary of the splat.

practice of mapping points to multiple pixels around their centers in screen space. Point representations of this kind are often referred to as splats or sometimes as primitives.

The next step in the evolution of point-cloud rendering towards GS is introduced by Zwicker et al. [ZPvBG02]. Their method, called Elliptical Weighted Average (EWA) splatting, introduces the point parametrization and rendering technique that sits at the base of GS. They parameterize points using anisotropic Gaussian normal distributions. Their splatting approach, therefore, leads to smooth point representation, which implicitly acts as a low-pass filter that reduces aliasing artifacts, while retaining visual fidelity. Figure 2.5 shows an example point cloud that is rendered using EWA splatting. Each point represents a 3D Gaussian distribution in space.

Before the arrival of GS, there was already research that introduced *differentiable renderers* for point-based techniques [YSW+19, WGSJ20]. These renderers enable the backpropagation of gradients from the rendered image to the point-cloud representation. GS combines this Neural Rendering approach with the visual fidelity of EWA splatting [KKLD23]. Figure 2.6 clearly shows that the anisotropic nature of the Gaussian splats is able to create smooth transitions and fill gaps.

Kerbl et al.'s GS [KKLD23] can be rendered very efficiently using a typical rasterizer architecture. This makes their method interoperable with most mesh-based rendering frameworks. It is also the key to GSs's rendering speed, which exceeds that of an unoptimized NeRF model by a factor of over 1000.

2.2.2 Model Definition

Kerbl et al.'s GS [KKLD23] directly builds on top of SfM. Their approach requires both the calibrated camera parameters as well as the sparse point cloud. Since the COLMAP

Figure 2.6: Side-by-side view of a point-cloud model and a GS model. The point-cloud model shows visible gaps and other artifacts.

points represent the scene's surface geometry, Kerbl et al. use them as an initial point cloud for the GS model.

GS [KKLD23] represents each splat with a three-dimensional normal distribution centered around point μ.

$$G(p) = e^{-\frac{1}{2}(p-\mu)^T \Sigma^{-1}(p-\mu)} \tag{2.5}$$

Equation 2.5 denotes the intensity of the splat at point $p = (x_p, y_p, z_p)$. Furthermore, the parametrization of each splat includes the opacity α. This value is multiplied with the intensity during rendering, which enables the capture of semi-transparent surfaces.

Each GS [KKLD23] primitive defines a diffuse color r, g, b. The color information can optionally be extended using spherical harmonics coefficients, a practice that was adapted from baked NeRFs [YLT+21, YFKT+21].

Kerbl et al.'s [KKLD23] implementation allows for a variable degree of spherical harmonics coefficients. This means that the amount of directional lighting information that can be captured in the model is configurable. The number of spherical harmonics coefficients based on the level $l \in \{0, 1, 2, 3\}$ can be seen in equation 2.6.

$$n_c = 3 \cdot (l+1)^2 \tag{2.6}$$

A spherical harmonics level of 0 means that only the diffuse color is represented [KKLD23].

The original GS release includes an option to export models as `.ply` files [Tur94]. Each splat is written to the file with the following attributes [KKLD23]:

- 3 attributes for the center μ

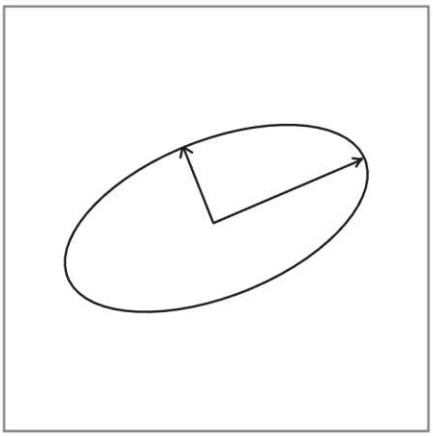

Figure 2.7: Exemplary image of a bounding box around an anisotropic Gaussian distribution.

- 3 attributes for the normal
- 1 attribute for the opacity σ
- 3 attributes for the scale
- 4 attributes for the rotation
- 3 attributes for the diffuse color
- 0 to 45 attributes for the spherical harmonics

All of these values are represented using 32-bit floating-point numbers. The 3D covariance Σ can be reconstituted from the three scale attributes and the rotation quaternion [KKLD23]. The file size grows linearly with the number of splats. This linear growth rate, combined with the long bit length of the attributes, means that models often result in large file sizes.

2.2.3 Rendering

As previously mentioned, GS is compatible with conventional rasterizers. This is because Kerbl et al. [KKLD23] represent each splat by four vertices and two triangles that form a rectangle. During rendering, the three-dimensional covariance is projected into a two-dimensional form based on the viewing transformation W. Equation 2.7 shows the projection of the covariance.

$$\Sigma' = JW\Sigma W^T J^T \tag{2.7}$$

J refers to the Jacobian of the affine approximation of the projective transformation. Given the 2D covariance matrix, the vertices are projected such that the quad forms a bounding box around the Gaussian.

Figure 2.7 shows how the bounding box is formed. Each vertex position is multiplied by $\lceil 3 \cdot \sqrt{max(\lambda_1, \lambda_2)} \rceil$, where λ_1 and λ_2 are the eigenvalues of the 2D covariance matrix [KKLD23].

Gaussians are rendered back-to-front in a process also referred to as the *Painter's Algorithm* [NNS72]. Sorting and blending incur heavy costs during render time [KKLD23].

2.2.4 Training Paradigm

NVS deals with deriving novel views, based on a limited set of existing views of a scene. When phrasing this problem in the terminology of machine learning, the known images represent our training set.

Kerbl et al. [KKLD23] use an iterative scheme to train GS models. Each iteration consists of picking an image and its associated camera parameters. During the forward pass, the model generates a view based on the camera parameters. Then the loss between the rendered output and the ground-truth image is computed. Equation 2.8 shows Kerbl et al.'s [KKLD23] loss function.

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{2.8}$$

The standard $\mathcal{L}_1$ loss is combined with the Structural Similarity Index (SSIM) loss function $\mathcal{L}_{D-SSIM}$. SSIM will be discussed further in section 2.3. For now, it is sufficient to say that it is a loss function that aims to capture the representation of structural image features. Kerbl et al. set λ to be 0.2 for their experiments [KKLD23].

GS models are initialized using the sparse point clouds that are created as an output of the initial SfM pipeline. To capture finer detail and close gaps in the representation, Kerbl et al. [KKLD23] developed an *adaptive density control* strategy, also referred to as *Densification*, which is applied at regular intervals. This strategy has the following goals:

1. Populate unoccupied regions with splats to add geometric detail where it is missing.
2. Break apart splats that approximate the scene geometry too coarsely.
3. Remove Gaussians that are virtually transparent.

To achieve the first two goals, their optimization technique keeps track of the view-space positional gradients of the splats. These represent regions that are likely not perfectly reconstructed. If the magnitude of a splat's positional gradient exceeds a threshold value, it is selected for densification. What happens to a splat if it is selected depends on its size [KKLD23]:

- **Small splats** are cloned. The newly created splat is moved in the direction of the positional gradient. This increases the volume of space covered by the splat model.

Figure 2.8: Examples of floaters in different datasets

- **Large splats** are split. Their scale is divided by a factor of 1.6 and they are evenly distributed according to their initial probability density.

The decision whether to select, clone, or split is governed by different threshold values. This densification heuristic is highly reliant on the choice of these hyperparameters [KRS+24].

To remove unnecessary splats, Kerbl et al. [KKLD23] propose a mechanism called the *opacity reset*. Every 3000 iterations, the opacity value of all splats is set close to 0. Their rationale is that a splat which recovers its opacity quickly is likely to approximate the actual scene geometry, while a Gaussian that stays transparent is likely unneeded. After 100 iterations, splats with an opacity value below a certain threshold are deleted.

A common problem for both GS and NeRF [WYZ+24, DHR+24] are so-called floaters. These are regions with cloud-like artifacts. Figure 2.8 shows examples of floaters in different GS models. The original GS implementation relies on a growing model size and constant opacity resets to slowly chip away at these artifacts [KKLD23].

Figure 2.9 shows the growth curve of the number of splats over the training of the truck scene from the Tanks and Temples dataset[1]. This is a typical development for the default densification strategy. We can see that the number of splats generally rises until the densification stops at 15K iterations. The number of splats does not increase monotonically. Every 3K iterations, splats are removed by the opacity reset. It becomes clear, however, that the model size increases over the training time. This increase in model size, combined with the number of parameters per splat, can lead to large file sizes. Lee et al. [LRS+24] state that most GS models trained on real-world scenes exceed one gigabyte in storage size. They also assert that primitives are often redundant. A motivating factor of their work is to alleviate this shortcoming of GS, which arises from the reliance on large models to achieve visual fidelity.

[1] https://www.tanksandtemples.org/

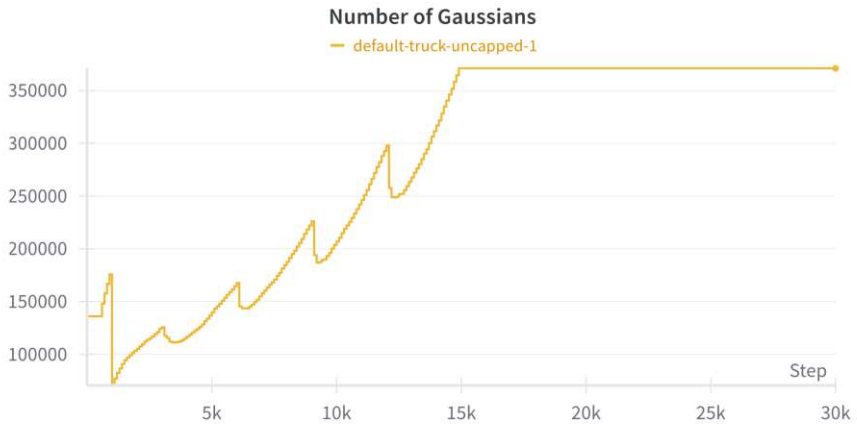

Figure 2.9: Development of the number of Gaussians while training on the Truck scene of Tanks and Temples

2.2.5 Gaussian Splatting Extensions

It is safe to say that the introduction of GS garnered a lot of attention, not just from computer vision researchers, but also within the larger 3D graphics community. The area has seen many contributions in the form of research as well as open-source projects. The following paragraphs will review some noteworthy contributions, which illustrate the capabilities of GS.

Large-scale GS While compact model sizes are a hurdle to getting GS production-ready, some research focuses on pushing the technology to create even larger scans. Kerbl et al. [KMK+24] developed a space-partitioning scheme and a Level of Detail (LOD) technique, which enables efficient training and rendering for large scenes. Liu et al. [LLF+25] also use LOD as well as a divide-and-conquer optimization approach to train city-scale models.

Combining GS with Meshes GS can be combined with mesh structures. Gaussian Opacity Fields (GOF) leverages the implicit occupancy representation of GS models as a framework for mesh reconstruction [YSG24]. Gaussian Frosting extends GS's capabilities by attaching splats to estimated mesh representations, enabling animation [GL24]. Kocabas et al. [KCG+24] propose capturing individual humans from a short monocular video stream. Their GS models are combined with the Skinned Multi-Person Linear Model (SMPL) [LMR+15], which integrates the scans into a flexible human animation framework.

Compact Data Representation One avenue for creating smaller models is to compress and quantize stored attributes. A number of techniques have been proposed that leverage the inherent redundancy and 3D structure of splats to encode them in an efficient

manner. Quantization and compression are used to reduce the file size dramatically [LRS+24, CWL+24, YYX+24]. The *Hash-grid Assisted Context for 3D GS Compression* by Chen et al. manages to achieve a 75-fold reduction in size when compared to the original approach [CWL+24]. Some of the proposed compression methods rely on adjustments made to the training pipeline to create optimal structures for their respective size reduction schemes [CWL+24, LRS+24]. This reduces the applicability of these techniques to the current landscape of GS models, which have been trained using a diverse array of splatting pipelines.

Data Formats The open-source community has contributed efficient data formats for splat models. Notable examples include the open-source release of Scaniverse's [Sca24] `.spz` file definition. Their approach leverages quantization in a clever way by prioritizing attributes that have a larger impact on perceived quality. Another example comes in the form of Mark Kellogg's [Kel25] open-source 3DGS web viewer and accompanying `.ksplat` file format. It enables dynamic streaming of the contents and displays the results to the users, just as they arrive on the client. The format is also highly configurable.

2.2.6 Reformulated Gaussian Splatting

In section 2.2.2, a clear outline for the data format of GS models is presented. Compression methods and new data formats modify the structure of the original format. Attributes are sometimes changed in the way they are represented, but all formats refer to the original set of properties. The following paragraphs describe GS approaches that break with this pattern of data representation. They completely reinterpret GS and the optimization pipeline to create different models and rendering procedures.

A less drastic derivation from the regular GS paradigm can be seen in Niedermayr et al.'s [NSW24] work regarding compression. Their approach reduces the model size by utilizing vector clustering on multiple attributes. Afterwards, they use k-bit quantization and fine-tune the model using the quantized representation. This means that the model is optimized based on the compression approach.

Lee et al.'s [LRS+24] work aims to drastically reduce the size of GS models. They use vector quantization for the scale and rotation attributes. The codebook for the vector quantization is learned during training, which means that it is optimized to fit the reference images. Their approach also forgoes all of the spherical harmonics parameters by using a grid-based neural field instead. This means that their color representation depends on a neural network.

Lu et al.'s [LYX+23] Scaffold-GS represents a more pronounced departure from the typical GS paradigm. They use a sparse voxel grid that holds a number of feature vectors. Gaussians are then spawned during rendering based on the camera parameters and the feature vectors. The splat parameters like mean, rotation, color, and so on are computed on-the-fly using neural networks. This reduces the number of primitives drastically, as Gaussians can adapt their appearance across different views.

The presented approaches are only a small sample of the universe of methods that reinterpret GS in their own way. These approaches are often very promising, but are hard to integrate into existing GS pipelines and rendering systems due to their wildly different requirements. This work will focus on approaches that conform to the original data specification, since this makes comparisons between methods much more straightforward.

2.2.7 Densification Improvements

As mentioned earlier, primitives in GS models can often be redundant. The default optimization procedure is not optimal, which means that an approach that makes smarter decisions about splat placements is needed [CW24]. There have been numerous releases that propose novel densification schemes. The following paragraphs will highlight some of these approaches.

General Improvements The previously mentioned GOF by Yu et al. [YSG24] redefines the splat-splitting criteria to reflect each splat's true contribution to the final render. This enhances the clarity in previously blurry regions.

Bulò et al. [BPK24] make several improvements to the original densification heuristic. This includes a weighted splitting and cloning criterion to prioritize splats, which are responsible for large errors. They also revise the opacity reset to be more gradual and introduce a growth control strategy for the number of splats.

Anti-aliased Gaussian Splatting As GS models grow in detail and granularity, they can suffer from aliasing, especially in distant views. Several strategies have been proposed to mitigate these issues, including multi-scale splatting [YLCL24], analytic splatting [LZH+24], and spatial-adaptive splatting [SZY+24].

Another technique that will become relevant later in this work is Yu et al.'s Mip-Splatting [YCH+23]. Mip-Splatting leverages a 3D smoothing filter that adjusts the size of Gaussians based on the maximum sampling rate, as well as a 2D Mip Filter to adjust for oversmoothing. Their approach also includes the new splat-splitting criteria introduced by GOF [YSG24], since it represents a general improvement on the splatting heuristic.

Pruning-based Approaches Fang and Wang [FW24] introduce a new splat-splitting criterion to reduce blur by leveraging depth information and improving spatial distribution. They also developed a pruning system that removes insignificant splats, producing more compact models. Girish et al.'s EAGLES [GGS23] introduces a novel influence metric for splat pruning, further refining model size by eliminating splats that don't significantly impact the final image.

Hybrid Approaches Niemeyer et al.'s RadSplat [NMR+24] uses NeRF as a prior to optimize point selection in GS, combining the strengths of both techniques for better performance and model size reduction. Similar work by Xiang et al. [XLL+24] uses SDFs

to guide the GS optimization process. SDFs are a natural fit for improving the geometric accuracy of models, since they are specifically designed to estimate surface structures and occupancy.

Li et al.'s Geo-Gaussian [LLD+24] has a similar goal to the hybrid approaches. It attempts to improve the splat placement by approximating the scene geometry more closely. The approach revises the cloning and splitting procedure. Thin Gaussians are used to approximate the surface geometry. Newly created Gaussians are then placed on the tangent space of the original splat. The loss function is extended to improve the placement of splats and encourage a smooth surface geometry in local neighborhoods. This means that the technique does not rely on any other neural techniques or data structures to achieve geometric accuracy.

Discovery-oriented Techniques Discovery-oriented approaches focus on uncovering new geometric features of the scene. One such approach, called 3D Gaussian Splatting as Markov Chain Monte Carlo (3DGS MCMC), is that of Kheradmand et al. [KRS+24]. They switch the traditional Stochastic Gradient Descent (SGD) over to Stochastic Gradient Langevin Dynamics (SGLD) [WT11], which stabilizes the optimization. 3DGS MCMC also utilizes a clever noise-injection strategy, which makes discovering under-reconstructed geometry much more efficient.

Gaussian-Pro, proposed by Cheng et al. [CLY+24], combines the priorities of 3DGS MCMC and Geo-Gaussian. It focuses on the discovery of under-reconstructed regions, while also trying to place splats more faithfully to the actual scene geometry. Using depth estimation and patch matching, pixels are reprojected back into the model, in areas where geometric detail is lacking. A similar loss function to Geo-Gaussian is employed to encourage neighboring splats to share normal information.

This section establishes the context for the techniques used in this study. It provides a broad overview of GS, covering its core principles, mathematical formulation, rendering, and training. Key extensions and reformulations that enhance GS are examined with a focus on densification improvements aimed at compactness. This provides the basis for understanding the comparative analysis presented in the following chapters.

2.3 Objective Image Quality Assessment

Digital images are often subject to distortions, which can occur during acquisition, compression, transmission, processing, and reproduction [WB06a]. The science of IQA deals with the identification and quantification of the resulting loss in visual quality [WB06a]. IQA is a core part of NVS. The 3D representations are trained based on the loss between the rendered image and the reference image. Here, the loss represents the degradation in image quality, which is meant to guide the optimization [TTM+22]. This section will present the goal of objective IQA, contrast it with subjective evaluation, and introduce metrics relevant to this study.

2.3.1 Taxonomy of IQA

Any visualization is ultimately meant to be viewed by humans. This means that an IQA system aims to capture the image quality as it is perceived by potential viewers. The most straightforward solution is to test a visualization in a study involving human participants. Capturing image quality this way is called subjective IQA and is often very costly and work-intensive. This problem is addressed by objective IQA. The field proposes a number of computational models, which automatically grade an image's visual quality. These models are based on the science of the human visual system and ultimately try to approximate the quality as it would be perceived by human beings [WB06a]. This section will give an overview of objective IQA.

IQA metrics can be classified as either full-reference or no-reference methods. Full-reference metrics consider both a pristine image and the generated images in unison to determine a score. No-reference metrics base the score solely on the features that are present within the generated image [MF21].

2.3.2 Relevant Quality Metrics

NVS can be seen as a form of supervised machine learning. The limited set of existing views serves as the reference images. This is why full-reference IQA measures are especially important for training techniques like GS. The following paragraphs will present relevant objective IQA metrics.

Mean Squared Error The Mean Squared Error (MSE) error function is very prevalent in machine learning . It does not specifically approximate any perceived qualities in the image, but it is an essential part of loss functions employed in NVS approaches like NeRF [MST+20]. Equation 2.9 shows the MSE for every pixel in P, based on the estimated color value $\hat{C}_p$ and the real color C_p [WB09].

$$\mathcal{L}_2 = \frac{1}{|P|} \sum_{p \in P} ||\hat{C}_p - C_p||_2^2 \tag{2.9}$$

Some approaches, including GS, actually use the mean absolute error as part of their loss function [KKLD23].

Peak Signal-to-Noise Ratio The Peak Signal-to-Noise Ratio (PSNR) expresses the ratio between the power of a signal and the power of the corrupting noise. In the case of IQA, the signal is the correctness of the generated image, while the error is the deviations from the reference. Equation 2.10 shows that the PSNR is derived from the MSE and does not add any new information. It simply rescales the error based on the dynamic range L of the image [WB09].

$$PSNR = 20 \cdot \log_{10} \left(\frac{L}{\sqrt{MSE}} \right) \tag{2.10}$$

Structural Similarity Index Measure Pixel-by-pixel errors are not analogous to perceived visual quality. The SSIM applies a new philosophy to tackle this problem. The central assumption is that the human visual system is especially adapted to recognize structural information [WBSS04]. To compute the SSIM, the image is divided into smaller patches. Each patch's SSIM is determined and aggregated across the image. Equation 2.11 shows the formula for computing the structural similarity index.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{2.11}$$

The variables x and y refer to aligned patches taken from both the generated and the reference image. Other components of the formula are defined as follows [WBSS04]:

- μ_x and μ_y reference the sample mean of the image, which expresses the local intensities.
- σ_x and σ_y are the local intensity variance and are a measure of contrast.
- σ_{xy} is the covariance of the intensity and expresses structural similarity.
- C_1 and C_2 are very small constants that depend on the dynamic range. They are used to stabilize the division.

The formula shows how the SSIM takes into account average local intensity, variance, and similarity. By averaging the local windows across the entire image, a measure of overall similarity is achieved.

Learned Perceptual Image Patch Similarity Learned Perceptual Image Patch Similarity (LPIPS) was introduced by Zhang et al. [ZIE+18]. Its goals are similar to SSIM, but it leverages a deep CNNs to determine the image quality, rather than a simple function. The input images are passed to a pretrained CNN. Pixels aren't compared directly. The metric is based on the differences between the computed features at multiple stages of the CNN. These differences are then weighted by learned perceptual weights, which adjust the contribution of different layers. Finally, the computed distances across all layers are averaged.

Blind/Referenceless Image Spatial Quality Evaluator Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) is a no-reference IQA metric introduced by Mittal et al. [MMB12]. They show that it outperforms the full-reference counterparts PSNR and SSIM, when approximating results from subjective IQA measurements. Their metric quantifies image distortions by focusing on deviations from the statistical properties of natural images. First, Mean-subtraction and Contrast Normalization (MSCN) is applied to the image, which yields MSCN coefficients. Natural, synthetic, and corrupted images tend to express different distributional patterns in their MSCN coefficients. Statistical

features are extracted and passed to a Support Vector Regressor (SVR) to obtain a final quality score. The original version of the SVR was trained on the LIVE IQA Database [MMB12]. This means that BRISQUE can also be seen as a pretrained IQA metric similar to LPIPS.

2.4 Subjective Image Quality Asssment

Subjective IQA aims to reliably retrieve quality scores from assessed items like images [MF21]. Several methodologies are standardized by the official recommendations given by the International Telecommunication Union (ITU)[2]. The ITU is a specialized agency of the United Nations responsible for issues related to information and communication technologies. It maintains a set of guidelines to inform researchers about how to conduct user studies to determine image quality [IR12, IT08]. These guidelines offer a variety of options for researchers when designing a subjective assessment. This section will give an overview of possible assessment items, score computation, and previously conducted studies in the field of NVS and Point-based Splatting.

2.4.1 Assessment Methodologies

Subjective assessment methodologies can be characterized by the following properties. Does the methodology...

> ...focus on absolute ratings or relative rankings of the images?
>
> ...present the participant with a single, a pair, or multiple stimuli?
>
> ...present the participant with a reference image to enable comparisons to the ground truth?
>
> ...use a continuous or a categorical variable to measure the user judgments?

These categories open up a diverse array of combinations. The most commonly used methodologies are presented in the following list [POMZ+20]:

- **Absolute Category Rating (ACR)** for single stimuli
- **Double Stimulus Impairment Scale (DSIS)** for two stimuli
- **Double Stimulus Quality Scale (DSCQS)** for two stimuli
- **Pairwise Comparison (PWC)** for two stimuli

[2] https://www.itu.int/en/about/Pages/default.aspx

Figure 2.10 shows examples for the different rating methods. Sometimes, the presentation of stimuli will be interleaved with a gray screen lasting around three seconds. This timeframe is also referred to as the Inter-Stimulus Interval (ISI) and is supposed to reset the viewer's visual system between stimuli [Bul14]. Each technique has different use cases based on its strengths and weaknesses.

Absolute Category Rating ACR measures the overall quality of a stimulus. The rating is recorded as a categorical variable. No reference to the presented image is provided. This means that its quality is supposed to be judged in isolation. The technique is useful for large-scale evaluations, when many single stimuli need to be evaluated relatively quickly [IT08, IT13].

Double Stimulus Impairment Scale DSIS measures the perceived degradation of an image in comparison to the ground truth. The rating is recorded as a categorical variable. This technique involves a comparison, yet it measures absolute quality, since the impairment is measured against a standard representing the ground-truth. This technique is suited for scenarios when testing the robustness of a system, for example, the quality of a compression algorithm or a transmission system [Bul14].

Double Stimulus Continuous Quality Scale This measurement technique presents users with a very fine-grained rating scale. It also involves two stimuli that are presented either back to back or next to each other. The technique enables users to express preferences for one of the stimuli, even if the perceived quality difference might be small. This makes the method great when researchers want to discriminate between two imaging systems with similar outcomes [Bul14].

Pairwise Comparison This is another approach that examines the quality differences between two stimuli. No reference is provided, only two distorted images. The rating scale is much simpler than for the other methods. This is its greatest strength, since it lowers the cognitive burden on participants. The method can also be used to discriminate between the images when the relative quality difference isn't too pronounced [POMZ+20].

2.4.2 Quality Score Computation

The previously presented rating techniques all yield quality scores that quantify the viewers' experience. ACR and DSCQS result in the so-called Mean Opinion Score (MOS). The MOS is a rational number, which expresses the mean quality assessment given by viewers. To this end, all judgments are mapped to an integer scale. Then the mean rating can be computed.

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N} \tag{2.12}$$

Equation 2.12 shows the relatively simple formula for this measure, where N is the number of viewers and R_n is a rating mapped to a natural number [IT13].

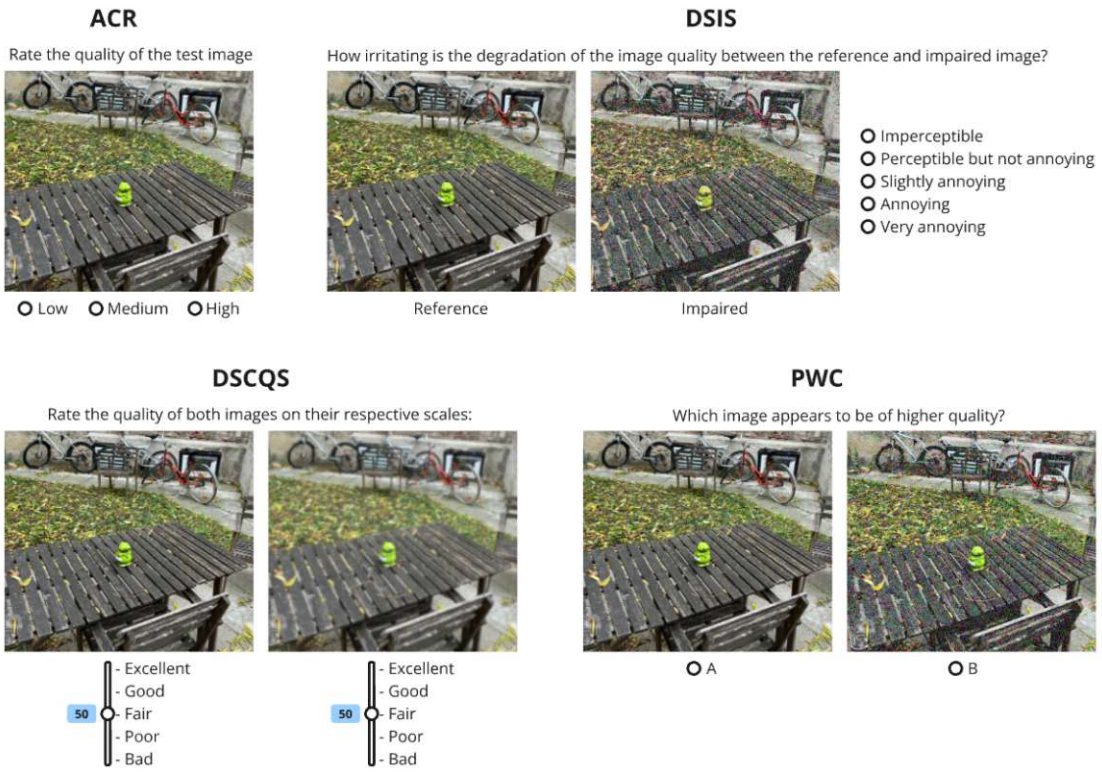

Figure 2.10: Example rating items for each of the different assessment methodologies [POMZ+20, Bul14]

DSIS yields the Differential/Degraded Mean Opinion Score (DMOS). It functions almost exactly like the MOS, but instead, it expresses the difference to the ground truth.

$$DMOS = \frac{\sum_{n=1}^{N} 5 - R_n}{N} \tag{2.13}$$

Equation 2.13 shows how it is computed, for a five-point rating scale [IT08, IT13, POMZ+20].

In comparison to the other methods, PWC leads to relative judgments. This means that computing scores is not as straightforward. The just-objectionable-difference (JOD) score is a numeric measurement of the quality, which can be derived from a set of preferences. Suppose that each image that has been tested using PWC has an underlying true quality score $q = (q_1, ..., q_n) \in \mathbb{R}$. From the experiments, the $n \times n$ comparison matrix C can be computed, where each element c_{ij} expresses how many times viewers preferred image i over image j. The probability that image i is better than image j is expressed by $\hat{p}_{ij}$ [POM17].

$$\hat{p}_{ij} = \frac{c_{ij}}{c_{ij} + c_{ji}}; \quad i \neq j \tag{2.14}$$

Given these probabilities, the underlying qualities can be modeled as a discriminal process as laid out by L.L. Thurstone [Thu27]. This model makes the assumption that the true quality scores are normally distributed. The discriminal process yields score distances between the images, from which the JOD can be computed using least-squares or maximum likelihood estimation [POM17].

A common way to compute the JOD includes mapping the distance of one JOD unit to a 75% certainty in the expressed preferences. For example, if image i is one JOD unit above image j, an average of 75% of all reviewers find image i objectionably better than image j. Taking this further, a distance of two JOD units corresponds to 91% agreement and a distance of three units to 98% agreement between reviewers [POM17].

2.4.3 Related Studies

Subjective assessment of NVS is a relatively young field [TAA24]. The following paragraphs will present a number of studies that show similarities, which showcase the common conventions that are evolving in this area.

Compression-related studies

Model size budgets can be set at different levels. Similarly, compression approaches also reduce model size. In the case of lossy compression, this creates multiple size levels and accompanying fidelity trade-offs. The following studies use subjective IQA to analyze this relationship.

The first relevant study does not deal with NVS, but instead with point-cloud rendering. Zerman et al. [ZGO+19] investigate compressed volumetric videos. Their study employs both PWC and DSIS as assessment items. Comparison pairs are formed based on different compression levels, in order to analyze the perceptual effect of compression. The study presents the correlation between subjective and objective measures.

In the field of GS, there is the study by Yang et al. [YYX+24]. They devise a lossy, graph-based compression approach for splat models and test it using a subjective assessment. Compressed models are contrasted with the uncompressed reference counterpart using DSIS. The study finds that more robust objective metrics are needed to quantify GS model distortions.

A similar compression study is conducted by Xing et al. [XYY+24]. They compare multiple NeRF approaches with different compression schemes across a number of datasets and quality levels. DSIS is used to compare compressed and reference video sequences.

The three studies above show that DSIS is a common choice for studies that deal with the assessment across different quality levels. All of the evaluations use rendered video sequences as a representation technique, which are then rated by the participants.

Model-related studies

The following paragraphs present a number of comparative evaluations concerning NeRF. Their goal is to compare the visual fidelity of different approaches. These studies also have to rely on pre-rendered videos, since many NeRF approaches cannot achieve real-time performance [KKLD23].

Martin et al. [MRAQ24] use DSCQS as an assessment technique, while Liang et al. [LWH+24] use PWC. Both of these works render videos based on the exact camera path that is outlined by the reference data. The reasoning is that the rendered and the reference video are displayed side by side for the comparative evaluation. If the camera paths were to diverge, comparison would be much more cognitively taxing.

Tabassum et al. [TAA24] truly leverage the NVS capabilities of NeRF, by rendering and presenting video sequences from unseen viewpoints. They use PWC as assessment items. The findings indicate that the perceived quality of the same scene can be highly variable across different camera paths. This indicates that the evaluation of novel viewpoints highlights different aspects of a scene's visual fidelity.

Summary

The presented studies show that in the field of subjective NVS assessment, both DSIS and PWC are established options. Most presented works use rendered videos, based on the camera path contained in the training data. This generally yields medium to high correlations between subjective and objective quality indicators [LWH+24, MRAQ24, XYY+24]. Tabassum et al. find a discrepancy between the metrics produced by the different evaluation approaches [TAA24]. This shows that the perceived visual quality can be highly variable when introducing unseen viewpoints [TAA24, QLC+24].

CHAPTER 3

Methodology

The advancement of GS has been rapid, and many novel densification techniques have been proposed. The aim of this work is to analyze which new techniques are particularly promising. This chapter will start by introducing and justifying the evaluation approach. Afterwards, the tested densification methods will be presented. Lastly, the implementation of the evaluation will be described.

3.1 Methodological Justification

Novel optimization and densification procedures for GS models tend to fall into one of three categories based on their goal:

1. Extensions that add new capabilities [KMK+24, LLF+25, GL24, KCG+24, FFS+24, WYF+24]
2. Techniques that aim to increase visual fidelity [YSG24, KRS+24, ZZX+24, LLD+24, CLY+24, BPK24]
3. Techniques that aim to create more compact models [FW24, GGS23, MGK+24, LRS+24, LYX+23, VPS+24]

Methods in the first category are not immediately relevant to this work. The main priority of these techniques lies in providing new features, rather than improving upon the existing optimization process. This study deals with approaches that improve upon GS, by tackling the problem of improper splat placement. Therefore, the focus will be on the second and third categories.

All of the methods cited in the relevant categories quantify their performance gains only in terms of objective quality metrics. This makes sense, since conducting a subjective evaluation for every new technique would be very work-intensive.

Figure 3.1: The first 8 observations from the "Truck" scene of the Tanks and Temples dataset. Test observations are marked in red.

It is important to look at how these test statistics are computed in order to understand why this approach might lead to faulty assumptions. Reference datasets, like Mip-NeRF 360 [BMV+22] or Tanks and Temples[1], usually contain between 100 and 1100 images. To create the default train and test splits, every 8th image is assigned to be a test observation. The images are often retrieved from a high-resolution video captured by a camera that traverses the scene.

Figure 3.1 clearly shows that train and test observations are rather similar. This is because the video frames from which they were sampled are in close proximity. Add to that the stable camera path in the video, and it is no wonder that the train and test images don't differ dramatically.

The reliance on similar train and test sets might yield an unrealistic impression of the real-world performance. A more robust quality evaluation could be helpful to verify the correctness of the objective indicators. Previous studies in this field (see section 2.4.3) often relied on pre-rendered videos that follow the camera path of the training data. Therefore, no completely novel viewpoints are presented. At best, the camera path contains semi-novel views in the form of interpolations between training cameras. The visual quality assessment of an NVS model is incomplete without showing it from a diverse set of angles [TAA24]. Therefore, this study enables viewers to manipulate the camera freely. Participants will be able to get a better sense of the scene's visual quality by exploring it according to their needs.

There is another shortcoming that occurs when new methods for efficient GS optimization are presented. The techniques cited above all showcase their efficiency improvements based on models with wildly varying sizes. This is problematic, since the actual efficiency gain can't be compared across techniques. Many of the models focus on uncontrolled model size reductions, rather than creating models with specified size constraints. Kheradmand et al. [KRS+24] use size constraints to show the efficiency of their densification solution by employing clear size budgets. This shows that size constraints can be a useful tool for evaluating optimization techniques with regard to compact models.

[1] https://www.tanksandtemples.org/

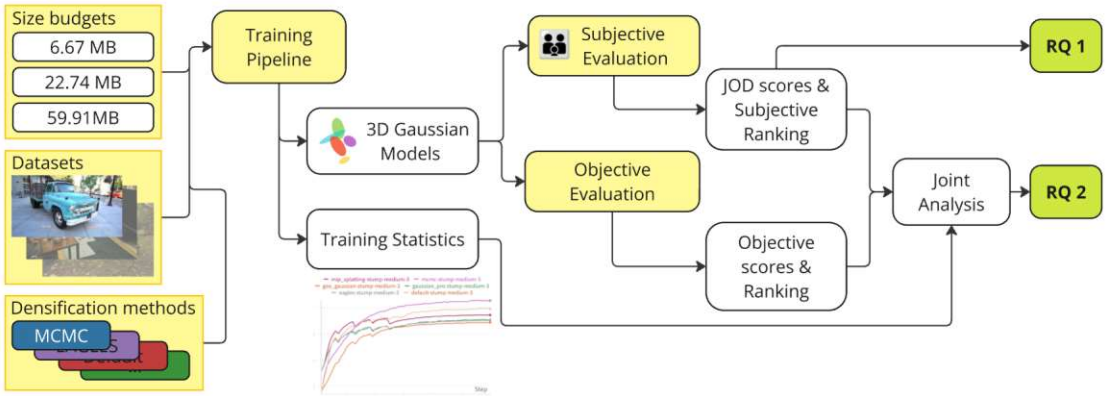

Figure 3.2: Diagram of the methodology used to answer the research questions.

Given a selection of methods, size budgets, and appropriate training data, comparable models can be created. A fundamental aspect of this work is that comparisons are made between models with fixed scenes and size budgets. Given three scenes and three size budgets, nine separate rankings are created, which are then aggregated to make overall statements about the methods. A scene and size budget combination will from here on out be referred to as a circumstance.

The approach of fixing both the scene and size budget has several advantages. These comparisons ensure that differences in model performance are solely attributed to the chosen splatting method, which enables fairer rankings. Additionally, cross-circumstance analysis provides insights into method consistency and reliability. The next section will show how this methodological concept can be used to answer the research questions.

3.2 Methodological Framework

There have already been a number of studies which dealt with the subjective evaluation of NVS and splatting techniques (see 2.4.3). This study's goal is novel, but its approach is firmly rooted in previous research. The following paragraphs will present its design and outline the reasoning behind it.

Overall Structure

Figure 3.2 shows the methodological approach as a diagram. The three central components are the seven chosen densification methods (3.3), three size budgets (3.4.1), and three datasets (3.4.2).

The *Training Pipeline* runs on the Vienna Scientific Cluster[2]. Each of the chosen densification methods has an accompanying optimization algorithm. The source of these implementations is outlined in appendix I. In order to run these distinct software

[2] https://vsc.ac.at/home/

packages, each is embedded in its own specific runtime environment. Using the pipeline, 63 unique models are trained. During the optimization process, *Training Statistics*, including performance curves, are created as a byproduct.

Afterward, the models are used in the evaluation. The evaluation compares models across different techniques to establish which technique produces the best results. For each comparison, the size and dataset variables are fixed in order to focus on the differences caused by the optimization technique.

The *Subjective Evaluation* results in JOD scores, which can be used to answer research objective one. JOD scores represent a relative ranking between the models. This can be used to determine which of the models has the best image quality in the eyes of the survey participants. The subjective evaluation is conducted using a visual application described in section 3.5.2.

The *Objective Evaluation* results in SSIM, PSNR, LPIPS, and BRISQUE scores. They are computed using the software component described in section 3.5.1.

Research objective two can be answered by looking at the subjective and objective metrics jointly. The correlation between the different measures describes whether SSIM, PSNR, and LPIPS are good approximations for the perceived quality. The training statistics are also considered, to analyze the behavior during optimization.

Subjective Assessment Items

Section 2.4.3 shows that both DSIS and PWC are established approaches for the subjective assessment of NVS methods. This study will utilize PWC. The assessment item entails a simpler task, which makes it suitable for non-expert participants. During initial in-person tests, it was found that a five-point rating scale led to occasional confusion. Due to inter-person biases, rating scales like DSIS can lead to large variations in the rating patterns. Simple comparative tests streamline this process and guarantee a ranked list based on the comparisons [POM17].

Active Sampling

PWC has another advantage. It can be combined with adaptive sampling procedures to reduce the number of comparisons that have to be made [POM17]. This study uses the Active Sampling for Pairwise Comparisons (ASAP) approach proposed by Mikhailiuk et al. [MWPO$^+$21]. As the name suggests, ASAP is an active sampling approach that can be used throughout the evaluation. After each comparison, their algorithm selects a new comparison pair based on the maximum information gain regarding the JOD scores. This approach increases the efficiency of the evaluation procedure.

Model Presentation

GS models encode the scene geometry explicitly and can be viewed from arbitrary viewpoints. Participants are presented with a real-time visualization of the models and

are able to manipulate the camera freely. The camera controls follow an orbital control scheme that also allows for panning and transitioning the camera origin. GS rendering also integrates seamlessly with traditional rasterization approaches. This will be used to display the models in a web-based application. By presenting the evaluation in this way, it becomes easier to reach more participants. It also demonstrates the capability of GS to deliver great visual experiences across different devices.

3.3 Chosen Densification Methods

This study's primary goal is to identify the best optimization procedures for constrained model sizes. To this end, a number of densification procedures have been selected as candidates for a comparative evaluation. This section presents the included methods, the reasons for inclusion, and the implementation adjustments that had to be made.

3.3.1 Inclusion Criteria

To ensure that GS models are meaningfully comparable, they must be sufficiently similar in structure. This study focuses specifically on analyzing efficient splat placement. Accordingly, the selected methods must differ primarily in how they position splats. Approaches that meet this criterion are discussed in Section 2.2.7.

While several techniques modify the rendering process or data representation (see Section 2.2.6), including them would introduce additional variables that affect visual quality beyond splat placement. To isolate the impact of splat positioning, this study only considers methods that retain the original data format (see Section 2.2.2). This ensures that splat placement remains the sole factor influencing model quality.

The second criterion requires that the code for the optimization process be both available and functional. Faithfully reproducing densification procedures based solely on a paper is impractical. Therefore, a working code release is essential to ensure the process can be reliably replicated.

To recap, the criteria are:

1. The optimization procedure must support the original data format.
2. The code must be available and working.

3.3.2 Chosen Methods

The following paragraphs will present the chosen optimization procedures and the reasoning behind their inclusion.

Default [KKLD23] The baseline for this analysis is the original approach as proposed by Kerbl et al.. Its opacity reset strategy already provides a method to remove superfluous

splats. By comparing its performance to more specialized densification techniques, it can be determined whether the new approaches actually yield any visual improvements. This method will also be referred to as the *Default*.

Mini-Splatting [FW24] Fang and Wang's Mini-Splatting aims to drastically reduce the model size of the trained models. Their approach introduces a new splitting pattern that covers under-reconstructed regions, which is called *Blur-Splitting*. Furthermore, a stochastic influence pruning approach removes splats that have a low impact on the rendered images. There is also a model reinitialization step called *Depth-Reinitialization*. It estimates depth values based on the current splat model and then selects new splats by reprojecting randomly sampled image pixels back into 3D space. This process redistributes splats across the entirety of the scene geometry. The method aggressively pursues the removal of unneeded splats, discovering new geometry and distributing detail across the scene. Therefore, it is a perfect fit for this study.

EAGLES [GGS23] Another technique that focuses on small models is Girish et al.'s EAGLES. This method uses a progressive coarse-to-fine strategy that first aims to capture larger aspects of the scene before moving on to fine details. It also employs influence pruning to reduce the number of primitives in the scene. This technique has a different focus than Mini-Splatting, as it employs a smoother and less disruptive training strategy. It is therefore another valuable addition to this study.

MCMC [KRS+24] *3D Gaussian Splatting as Markov Chain Monte Carlo* was introduced by Kheradmand et al.. It bridges the gap between compactness and discovery. On the one hand, it is one of the only optimization techniques that purposefully creates models of restricted size, while on the other, enabling better discoverability through a probabilistic optimization approach. It uses the same opacity reset technique for splat removal as the default technique. This method represents an elegant way to frame GS optimization, which is ideally suited for working under a size constraint. Throughout this work, this technique will be referred to as *MCMC*.

Gaussian-Pro [CLY+24] Another promising discovery-oriented technique is Chen et al.'s Gaussian-Pro. Their approach aims to discover new geometry and accurately estimate properties like surface alignment. To this end, they employ a patch matching technique and normal modeling. The approach has less of a focus on small model sizes, but it has the potential to excel in this task. Its discovery mechanics can lead to a quicker convergence and accurate modeling of the scene geometry. This could avoid local minima and improve upon the default, which is why it was selected for this analysis.

Geo-Gaussian [LLD+24] Li et al.'s Geo-Gaussian's sole focus lies on the faithful representation of the scene geometry. It uses geometric constraints, a special loss function, and a custom propagation approach. This is another technique that is not focused on smaller models, but its mechanics could yield models that are more concise. It is similar

to *Gaussian-Pro*, but fully commits to the idea of geometric consistency. This makes it another useful addition.

Mip-Splatting [YCH+23] The last included approach is Yu et al.'s Mip-Splatting. Their method has completely different priorities than the previously presented ones. It utilizes a 3D smoothing filter and 2D Mip filter to prevent aliasing artifacts. These mechanics have the effect that splats tend to fill up more space. In addition, the method includes an updated splat-splitting criterion. The approach also does not focus on small model sizes, but it leads to models that are well-defined in high-frequency areas, without introducing small artifact-like splats. This leads to improvements when viewing the models from novel camera positions and represents a general improvement of the optimization pipeline. Its inclusion serves as a check if the focus on compact models actually pays off when contrasting it with other quality advantages.

3.4 Training Setup

This section presents aspects essential for the training. This includes the size budgets for the models, the selected datasets, and adjustments made to the training methods.

3.4.1 Size Budgets

The size budgets serve as the basis for the comparisons made between the different models. In order to establish these parameters, inspiration is taken from factors that negatively impact the adoption of GS.

There are two scenarios where model size restrictions are especially relevant. The first occurs when training in a memory-limited environment. Training is not feasible if models outgrow the imposed memory limit. The second scenario arises when considering web applications. Without a bounded model size, the download times can become so long that they negatively affect the user experience.

The first scenario is only relevant for users who want to train their own models. This is most likely to be the case for researchers and developers. The second scenario is more relevant as a limiting factor since it affects all users. This is why it is the focus of this study.

Long load times, especially in interactive applications, can lead to frustration for the users. This makes GS a less attractive option as a 3D representation technique. Therefore, the budget parameters will be established in the context of a web application.

In order to formulate budgets, three questions have to be considered:

1. What are real-world internet speeds?
2. What are acceptable loading times for an interactive web application?

Quantile	Speed in Mbps
20%	17.80
50%	60.64
80%	159.75

Table 3.1: Internet speeds by quantile. Retrieved from RTR-Netztest on 2.12.2024.

3. How is the number of splats connected to the model size?

By answering all of these questions and considering them together, a relationship between the number of splats and the impact on real-world load times can be established. The following paragraphs will go over how these questions were answered and how the size budgets were established.

Real-world Internet Speeds

There are a number of platforms that track internet speeds across different regions. The *Speedtest Global Index* by Ookla presents a global leaderboard of countries by their internet speeds[3]. In Austria, there is the *Breitband-Atlas* by the *Bundesministerium für Finanzen*. This tool shows a fine-grained map of Austrian internet speeds, which are resolved down to the level of neighborhoods[4]. While these reports are powerful, they are not suitable for this analysis. This study requires information about how the access to internet speed is distributed. Thankfully, there is a service that tracks real-world internet speeds as they are experienced by users in Austria. This report is maintained by the Austrian Regulatory Authority for Broadcasting and Telecommunications (RTR)[5] and it is called the *RTR-Netztest*[6].

The data from the RTR-Netztest is broken down by internet service provider and quantile. This allows a more fine-grained analysis of the experienced internet speed. Only speed data from browsers were included in this analysis. The data was retrieved on the 2nd of December 2024. In order to aggregate the data, it was averaged across the internet service providers, taking the number of samples per provider into account. The final speed estimates can be seen in table 3.1.

Acceptable Load Times

User satisfaction and retention are connected to Page Load Time (PLT). Estimating a PLT that is perceived to be tolerable is non-trivial, since it depends on the context and the specific user [KRBD17]. This means that the acceptable PLT is a distribution. One of the largest studies of PLT was conducted by Akamai Technologies and SOASTA

[3] https://www.speedtest.net/global-index
[4] https://breitbandatlas.gv.at/
[5] https://www.rtr.at/rtr/wer_wir_sind/Organisation/Organisation.en.html
[6] https://www.netztest.at/en/Statistik

[TS17], which are both private companies specialized in digital performance management. They show that loading times between 1.8 and 2.7 seconds are ideal. These were the times when the conversion rate, which is the percentage of customers who make a purchase, was the highest. Other influences come from sites like *Sketchfab*[7], *PolyCam*[8], *KIRI Engine*[9] or *Luma AI*[10]. These sites already host 3D models, including GS variants. Many of the models that are on display have much longer load times than three seconds.

Based on the research by Akamai and the experience with other 3D web platforms, the maximum load time for this study was set to 3 seconds. GS models are able to load progressively, which means that users will be able to interact with the model as soon as they begin streaming in. The interactive experience minimizes user frustration. On the other hand, the load time imposes a relatively strict constraint on the models. These circumstances will yield insights into how well the optimization techniques can adapt.

Splat Count vs. Model Size

In order to impose a size limit, the relationship between model size and splat count has to be established. For this work, the number of spherical harmonics will be limited to degree 0. This means that no directional colors will be included. The reason for this decision is that splat models generally require a large number of splats to converge. Therefore, more splats are prioritized over directional colors.

Given the number of attributes that a model includes, it is trivial to establish a relationship between the number of splats and the final file size. Equation 3.1 shows how the file size in megabytes can be computed, for a spherical harmonics level of 0.

$$m = n \cdot 17 \cdot 32 \cdot \frac{1}{8} \cdot 10^{-6} \tag{3.1}$$

Where m is the file size in megabytes, n is the number of splats. It is clear that the file size is linearly dependent on the number of splats. This formula does not account for the file header, since its size is negligible.

The problem with the computation above is that it does not account for compression. Files can be losslessly compressed using algorithms like DEFLATE [Deu96]. The files in this study use the compression scheme that was outlined by Niedermayr et al. [NSW24]. They sort the splats using their Morton order and then employ the DEFLATE algorithm.

In order to model the relationship between the number of splats and the compressed file size, a number of candidate splat models were analyzed. The original file sizes of the models range from 30,000 bytes to 942 megabytes. The models were compressed and their file sizes measured. Then, a linear model between the number of splats per model and the compressed file size was computed. The outcome for this model can be seen in

[7] https://sketchfab.com/
[8] https://poly.cam/tools/gaussian-splatting
[9] https://www.kiriengine.app/
[10] https://lumalabs.ai/interactive-scenes

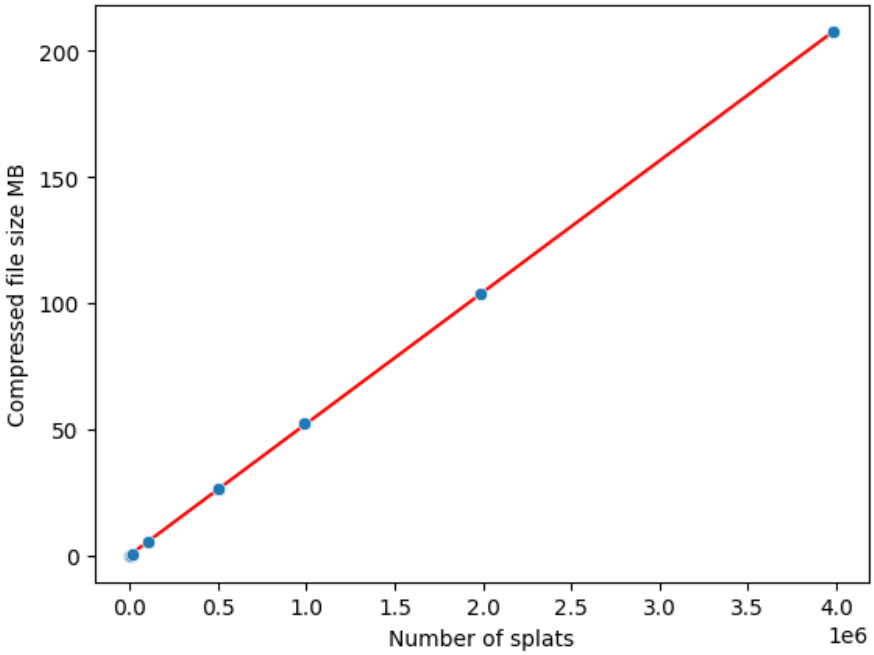

Figure 3.3: Linear relationship between compressed model size and the number of splats

figure 3.3. The relationship is almost perfectly linear, which is also indicated by the R^2 score of 99.99%.

Computing the Size Budgets

Now that the internet speeds, the acceptable load times, and the relationship between model size and splat count have been established, the size budgets can be computed. The relationship between the maximum wait time in seconds t and the number of splats n is as follows:

$$\frac{m_\beta(n)}{s} \leq t \tag{3.2}$$

Where $m_\theta(n)$ is the model describing the size in megabytes and s is the speed in megabytes per second. This can be rewritten such that the budget size becomes explicit:

$$n \leq \frac{t \cdot s - \beta_0}{\beta_1} \tag{3.3}$$

Given this equation, the estimated linear relationship between the number of splats and the model size, the maximum load time, and the internet speed, the maximum number of splats can be computed. Each of the three quantiles in table 3.1 has its own splat limit. In the case of the 20% quantile, the formula is as follows:

$$n \leq \frac{3 \cdot 2.22 - 0.23625}{5.459 \cdot 10^{-5}} = 117856 \tag{3.4}$$

Name	Quantile	Speed in Megabytes/s	Number of Splats
Low	20%	2.22	117856
Medium	50%	7.58	412235
High	80%	19.97	1093135

Table 3.2: Final size budgets

The variables have been replaced with their respective values:

- $t = 3$ seconds
- $s = 2.22$ megabytes/second for the 20% quantile
- $\beta_0 = 0.23625$ megabytes
- $\beta_1 = 5.459 \cdot 10^{-5}$ megabytes/splat
- 117856 is the maximum number of splats for the 20% quantile

The maximum number of splats is then used as a hyperparameter for the optimization procedures. Table 3.2 shows the final size budgets that were used for this study.

3.4.2 Dataset Selection

The choice of datasets was informed by the literature in the field. Over the past few years, a number of datasets have become staples in NVS research. They appear frequently in studies and serve as a standard for quality measurements. The captured images contained in these datasets tend to be of good quality, and the COLMAP database is included. This means that different approaches work with the same baseline.

Three scenes were picked from two different datasets. The datasets that were included are Mip-NeRF 360 [BMV+22] and Tanks and Temples[11]. Both of these datasets have become standards in the NVS community, since they are the baseline of a diverse array of studies [KKLD23, FW24, GGS23, KRS+24, CLY+24, YSG24, CWL+24, YLX+24]. In fact, most studies in the field use either one of these datasets at least once.

In order to pick a number of scenes for this study, the following aspects were considered:

1. The scene content
2. The number of images per scene

The aim was to achieve diversity in both of these aspects. This creates variations in the circumstances, which yields a more interesting quality profile. Examples of the chosen

[11] https://www.tanksandtemples.org/

Figure 3.4: Examples of the chosen scenes

datasets can be seen in figure 3.4.

Stump represents a scene with dense foliage and little to no reflections. It has the lowest image count, coming in at 125 distinct views. *Truck* is a mix of foliage, solid surfaces, and open sky. It contains diffuse and specular materials and has a slightly larger image count of 251. *Room* has 311 images and is therefore the largest dataset. Its content, however, is the smallest in terms of real-world scale. The scene shows a single room, which is filled with different objects that have diverse material properties.

The three scenes represent a good mix of different content types. There is a natural outdoor scene with dense foliage, a transitional scene with a mix of foliage and constructed surfaces, and an indoor scene. This choice represents common scenarios for NVS applications.

Tanks and Temples contains scenes with even larger real-world scales and image counts. These were not considered, since such a large amount of images can lead to problems regarding the graphics card memory.

The image resolution was not a factor in the consideration. All optimization pipelines automatically rescale images to a maximum width of 1600 pixels. The considered scenes exceed this threshold. Therefore, the image resolution is essentially the same for all scenes.

3.4.3 Training Adjustments

In order to create a sensible training setup for all of the optimization procedures, a number of adjustments were made to their respective codebases and hyperparameters. References to the modified code can be found in appendix I. This section will give a brief description of the changes.

Maximum Number of Splats

All of the optimization procedures come with a hyperparameter that enables a splat limit. The implementation of this hyperparameter is flawed. Instead of enforcing an upper bound on the number of splats, it simply ceases all densification-related operations as soon as the given threshold of splats is exceeded.

This approach has two major drawbacks. Firstly, the splat count often ends up exceeding the maximum limit. The densification operation that occurs just before the splat limit is reached can add any number of splats. So a situation where the newly created splats and the existing splats surpass the desired amount regularly occurs. The hyperparameter doesn't guarantee a size budget, rather it can be seen as a type of stopping criterion.

The second issue stems from the fact that the opacity reset, which is often the only pruning operation, is also halted as soon as the splat limit is reached. This means that superfluous splats cannot be removed.

Given these obvious limitations, the implementation of the hyperparameter has been revised. The procedure is relatively simple. Each time the densification takes place, the difference d between the current number of splats and the maximum cap is computed. The clone and split procedures select splats based on their accumulated screen space gradients. If the number of selected splats exceeds d, then only the Gaussians with the top d gradients are actually used for the duplication.

Another effect of this method is that the pruning procedure runs every single densification iteration. This means that low opacity splats are always removed even if the splat count exceeds the limit. When unneeded splats are removed, new capacity for the creation process is automatically allotted. Using this technique, the splat count threshold cannot be exceeded, and pruning continues even if the cap is reached.

Mini-Splatting provides the only exception to the described paradigm. It introduces another splitting procedure called *Blur-Splitting*. Here, splats that have a large contribution to the rendered images are tracked and selected for splitting. This identifies and splits blurry-looking Gaussians. In order to consider this mechanic, the approach described above was modified slightly. The computation of the difference d remains. A hyperparameter called $\lambda_d \in [0, 1]$ is introduced. For a given densification step, the maximum number of splats that are selected for blur splitting is $\lfloor \lambda_d \cdot d \rfloor$. After blur splitting, d is recomputed to allow for the normal creation process to continue. This allows for the maximum number of splats to be created in case the number of splats eligible for Blur-Splitting is lower than $\lfloor \lambda_d \cdot d \rfloor$. In all of the experiments, λ_d was set to 0.2.

Model Initialization

GS models are initialized based on the sparse point cloud of the scene's COLMAP scan. In some cases, the number of COLMAP keypoints exceeds the size threshold. When this occurs, the point cloud is randomly subsampled in order to conform to the maximum number of splats. All methods use the same subsample for initialization.

Hyperparameters

In order to capture the intent and decisions made by the original authors of the respective optimization procedures, hyperparameters are not changed unless absolutely necessary.

The only warranted change was made to Mini-Splatting. To achieve the smallest models possible, the authors stop the densification at 15K iterations. This is halfway through the training process. In the context of this study, stopping the densification this early is not needed. The size constraint is enforced by the splat budget and not by pruning. Therefore, the densification was extended until 29K iterations.

Behavior Tracking

The idea behind this work is not just to understand which method works best, but also how and why methods work. To this end, the optimization approaches' logging was extended to provide insight into the development of the splat count. At each iteration, the number of created and deleted Gaussians is reported. The result of this logging procedure is presented in section 4.4.

3.5 Implementation Details

Apart from the training adjustments, there are two more implementation-related aspects of this work. Both the objective and subjective evaluation rely on software components. The design of these components will be described in the following paragraphs.

3.5.1 Objective Evaluation

The evaluation requires the objective quality metrics for every scene, model, and size combination. To compute these metrics, a dedicated software component was designed. The algorithm that computes these metrics is outlined in appendix B.1. The goal of the algorithm is relatively simple. Based on each of the datasets, the test set is loaded. Then, the objective quality metrics are computed for each method and each size.

BRISQUE is a no-reference metric, which means that it does not rely on a ground-truth reference image. It is computed for the test views as well as N_p additional perturbed poses, which are created per reference camera. Perturbed poses slightly modify the original, by varying the camera center and rotation. For the conducted experiments, N_p was set to 5, camera locations were perturbed by sampling from a 3D normal distribution with a standard deviation of 0.01 and camera rotations were completely random. This strategy for computing the BRISQUE score is designed to give a better sense of scene quality outside of the camera path, which is implicitly traced by the training set.

In comparison to the training pipelines, the evaluation renderer is built for efficiency. Improvements come in the form of a more performant dataset loader and disabled gradient computation. It also uses the `gsplat`[YLK+24] rendering backend because of its increased computation speed. By selecting a common rendering framework for the metrics computation, any latent effects on the image quality are minimized. This is an important issue, since the different methods all use slightly modified rendering pipelines, which can lead to slightly different metric results.

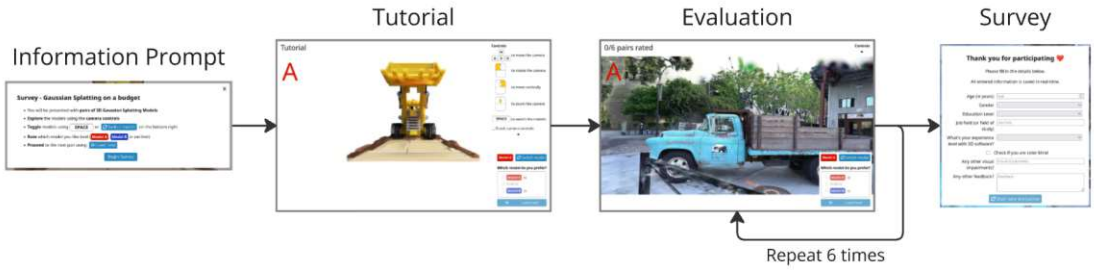

Figure 3.5: Outline of user flow to complete evaluation

3.5.2 Subjective Evaluation

To make the subjective evaluation a seamless and smooth process, a dedicated evaluation interface was developed. The interface handles all aspects of the evaluation, including instructing the candidates on how to use it, presenting the assessment items, and collecting the data. The tool is available over the internet[12] via any modern browser and will be supported for the upcoming year after the release of this work. The following paragraphs will detail the design decisions for the interface and the architecture of the application.

Interface Design

The interface of the evaluation application presents viewers with everything they require to participate. Its layout is structured to accommodate desktop devices and laptops. The interface aims to guide the users' exploration through the application and build understanding. The user flow, as can be seen in figure 3.5, starts with a briefing pop-up, leads to a tutorial, and then finally to the evaluation section. The tutorial provides a very clear-cut assessment example that teaches the application's functionality, such that the learning experience does not interfere with the actual evaluation. After the participant has rated six pairs of models, they are presented with a brief demographic survey.

Figure 3.6 shows the layout of the application when in the evaluation mode. The interface presents the viewers with a fully interactive view of the 3D scene. Each evaluation step presents a pair of 3D models, which share the scene they have been trained on and the size constraint hyperparameter.

In the top left, general information, like the number of rated items and the currently viewed model, is displayed. Participants can toggle between model `A` and `B` using the space bar or the buttons in the lower right. At the top right, the camera controls are explained. This prompt automatically minimizes when entering the evaluation section, but can be expanded at any point should the need arise. The rating item can be found in the lower right part of the screen. It presents the user with some redundant information about which model is currently visible. This is to avoid situations where users might be confused about which model they are viewing at the moment.

[12] `https://gs-on-a-budget.firebaseapp.com/`

Figure 3.6: User interface elements of the evaluation application

The interface is structured so that important elements lie on the top-left to bottom-right diagonal. This directs the attention to the model presented and the rating item. Auxiliary information is presented on the top right as to not be too distracting. Users have full control over the 3D position of the camera. When toggling between the two models, the camera position is retained. This way, users can easily spot differences in the models, which helps especially when trying to make out subtle differences.

The application is designed to leverage the intuitive nature of PWC. Its online accessibility and easily explanatory nature help to reach as many participants as possible.

System Architecture

A major priority for the development was that the application would be easy and fast to deploy. The application does not require complex functionality in the backend, which is why a microservice architecture was chosen. For hosting, deployment, cloud storage, and database services, the application relies on Firebase[13]. Figure 3.7 shows a component diagram of the application. The following paragraphs will describe each system component.

Frontend

The frontend implements the interface described previously as a single-page web application. All 2D components are built using React[14]. The 3D viewer leverages `three.js`[15] for the camera controls and scene management. The GS models are rendered using the GS rendering integration for `three.js`, which is developed and maintained by Mark Kellogg and members of the open-source community[16].

[13] `https://firebase.google.com/`
[14] `https://react.dev/`
[15] `https://threejs.org/`
[16] `https://github.com/mkkellogg/GaussianSplats3D`

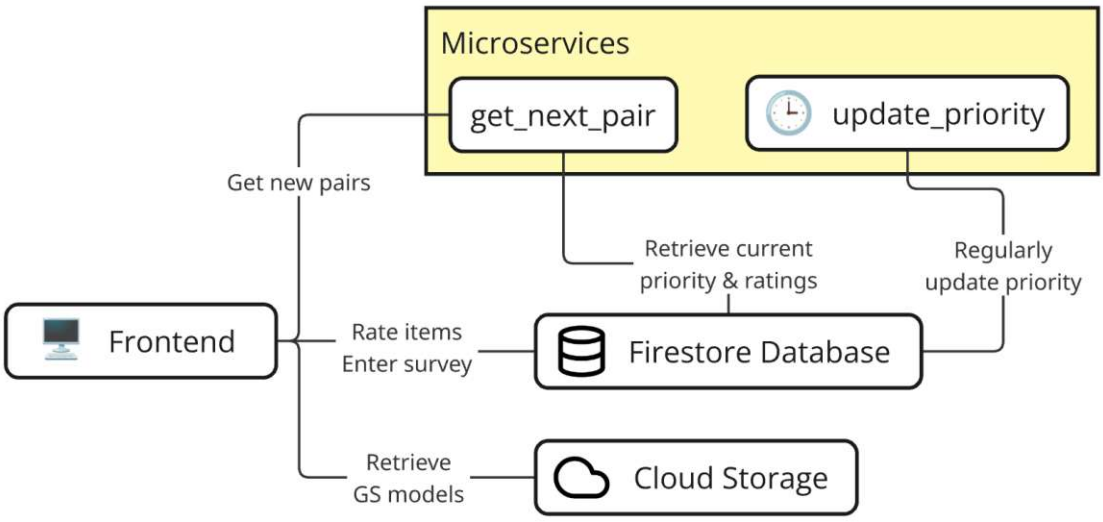

Figure 3.7: Component diagram of the evaluation application

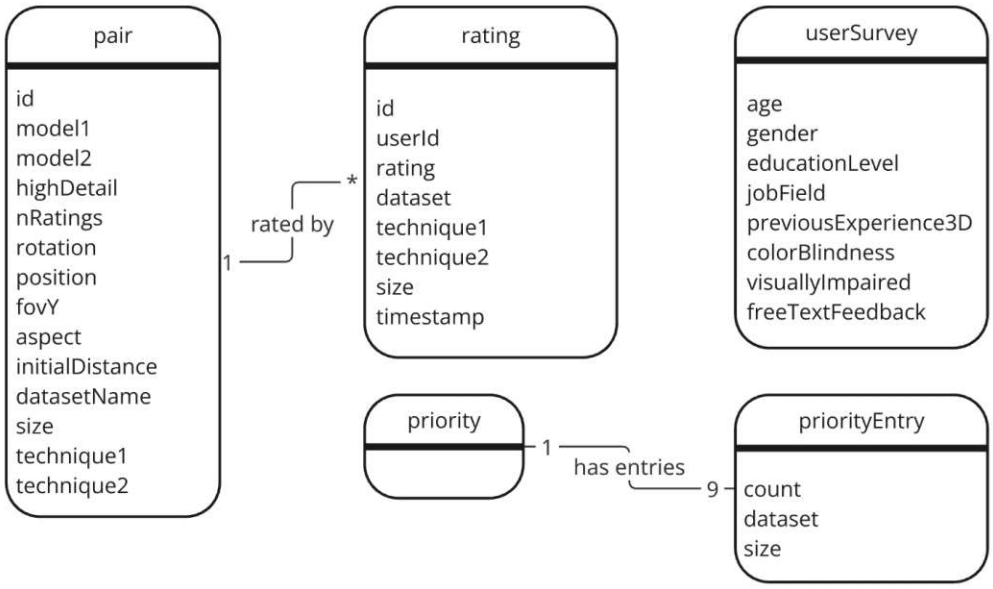

Figure 3.8: Entity relationship diagram of the evaluation software's data model

Firestore Database

The Firestore database holds all relevant data for the evaluation. The different entities and their attributes can be seen in figure 3.8. The `pair` entity defines a pair of models that can be compared. It holds the model locations, some simple metadata, and camera parameters for rendering. Ratings are stored using the `rating` entity. It refers back to the pair, holds the preferred model, and some metadata. The data gathered by the user

survey is stored in the `userSurvey` entity.

A relatively complex aspect of the application is the priority system. Its job is to balance the number of ratings over all the scene and size combinations. The `priority` entity acts like a cache, which manages a number of `priorityEntry` entities. These hold information about how many ratings each scene and size combination has already received.

Priority Updates

Priority updates are essential for balanced evaluation results. The update functionality is implemented by the `update_priority` cloud function. Three times per day, the function recomputes the rating count grouped by scene and model size. It then writes the result into the `priority` entity.

Next Pair Retrieval

Every time users have rated a pair, the frontend calls the `get_next_pair` function. The function then picks the least rated scene and size combination based on the latest `priority` entity, and computes the best next pair using ASAP (see 3.2). The new pair is sent to the client, where it can then be displayed.

Cloud Storage

The cloud storage is where the GS models are stored. The frontend application can simply create download links for the models if it knows the path to the storage bucket of a specific model. The paths are stored in the `pair` entity.

CHAPTER 4

Results

An aspect of this study is that it compares GS optimization methods in different circumstances. As previously outlined, the circumstances are characterized by the scene and the model size constraint. Comparisons are based on the relative model performance in a single circumstance. This analysis presents models both from the viewpoint of singular and across multiple circumstances. This highlights situation-specific strengths and weaknesses, but also shows general performance. The analysis is split into multiple sections, each presenting an aspect of the gathered data.

4.1 Subjective Results

This section presents the results of the subjective evaluation. First, the overall subjective ranking is presented, followed by the rankings per size budget. Then, the cross-circumstance agreement is analyzed. Next, clusters of circumstances are formed based on the agreement. Lastly, the reliability of the subjective evaluation is analyzed. These results are key to answering the first research question, because the perceived quality represents the gold standard for the evaluation of visual content [WB06b].

4.1.1 Overall Subjective Ranking

When evaluating the best option out of a lineup of possibilities, ranked lists are a great way of presenting the results. One tool that will be used across this entire chapter is the Borda count. This voting system allows for the aggregation of rankings across different circumstances. Each model is assigned a score by adding the ranks it received. In the context of this study, higher scores are better.

In order to get a sense of the variability, multiple versions of the data are created. JOD is computed using the probabilistic model of Thurstone's Case V. The ratings retrieved with ASAP are relatively sparse. These factors make the score computation sensitive

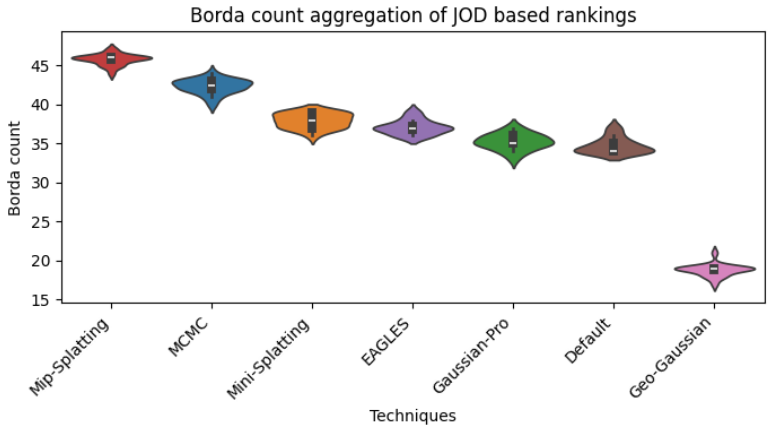

Figure 4.1: Overall subjective rankings aggregated by Borda count and 20-Fold Cross-Validation, higher is better

to oversampling, which amplifies noise in the data and can lead to exploding scores. Therefore, bootstrapping is unsuitable in this scenario. Instead, 20-Fold Cross-Validation is used. Each fold yields its own Borda count, which is then visualized using violin plots.

Figure 4.1 shows the overall subjective ranking. The visualization presents a surprising result. Mip-Splatting is ranked the highest, followed by MCMC and Mini-Splatting. This could indicate that the general quality improvements introduced by Mip-Splatting outweigh efficient splat placements.

Another interesting aspect is the pronounced last place of Geo-Gaussian. This is a very clear indicator that the strategy resulted in identifiably worse results.

4.1.2 Rankings per Size Budget

The overall ranking provides only a partial picture. To better understand how each method performs under different size constraints, it is useful to examine how the rankings evolve as the size budget changes.

Figure 4.2 illustrates how the rankings shift across varying size budgets, highlighting differences compared to the overall ranking. It becomes clear why Mip-Splatting is ranked so highly. Across all of the size budgets, it is ranked relatively high, never falling below third place. The most surprising result is the performance of EAGLES. Its rank increases drastically as the size budget increases. MCMC and Default tend to fall off as the size increases. Especially MCMC is ranked highly in the low and medium size budgets, but falls off in the high size budget. Geo-Gaussian always occupies a spot in the bottom two ranks. To view the rankings as violin plots, refer to figure C.1 in the appendix.

The results indicate something surprising. When considering the lowest size budget, Default is the second-highest ranked method, which indicates that the original densification scheme already outperforms most other models when considering strict size limitations.

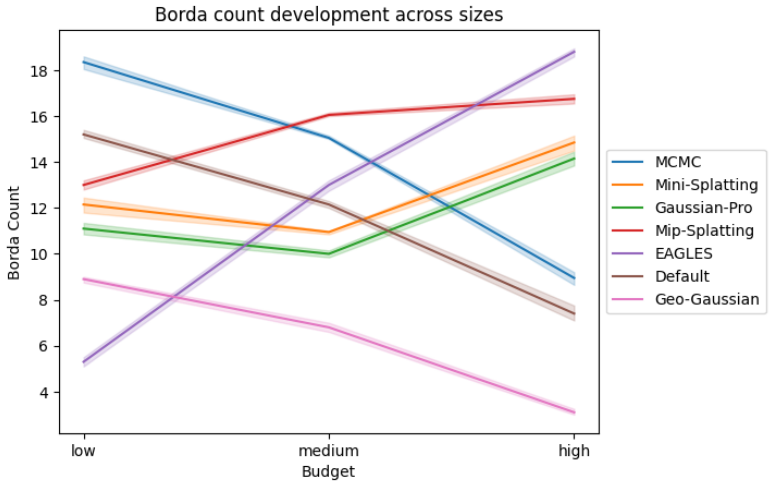

Figure 4.2: Evolution of subjective rankings aggregated using Borda count across size budgets. Higher values indicate better rankings.

MCMC seems to be a straightforward improvement over the Default, as it is always ranked higher. It is noteworthy that these techniques are highly related, since they both use the exact same pruning strategy and MCMC only modifies the splat placement.

Other methods, like EAGLES, Mini-Splatting, and Gaussian-Pro, tend to perform better when the size budget is larger. This indicates that these methods might not be the best candidates for size-constrained optimization, as they rely on higher splat counts for relative performance improvements.

4.1.3 Cross-Circumstance Rankings

The previous sections demonstrate that the rankings vary across different conditions. This variation raises the question of how consistent the rankings are between these circumstances. To investigate this, the rank correlation between each pair of conditions is computed using the Spearman's Rank Order Correlation Coefficient (SROCC) [Spe04].

SROCC was selected for this analysis because it does not assume linear relationships, making it well-suited for comparing rankings. The results of the correlation analysis are shown in the appendix in figure C.2. The figure presents a heatmap, where each cell indicates the correlation between a pair of conditions.

With an SROCC of 0.097, the overall correlation is so close to 0 that it clearly indicates a highly divergent ranking structure across the circumstances. This raises the question of whether the Borda count ranks tell the whole story. It could be that the best method actually turns out to be a very different one when looking at a subset of the data. In order to find a fitting subset of the data, clusters of circumstances based on their correlation were identified. The SROCC scores are hierarchically clustered using the Ward method [WJ63]. The resulting clustering is presented in figure 4.3.

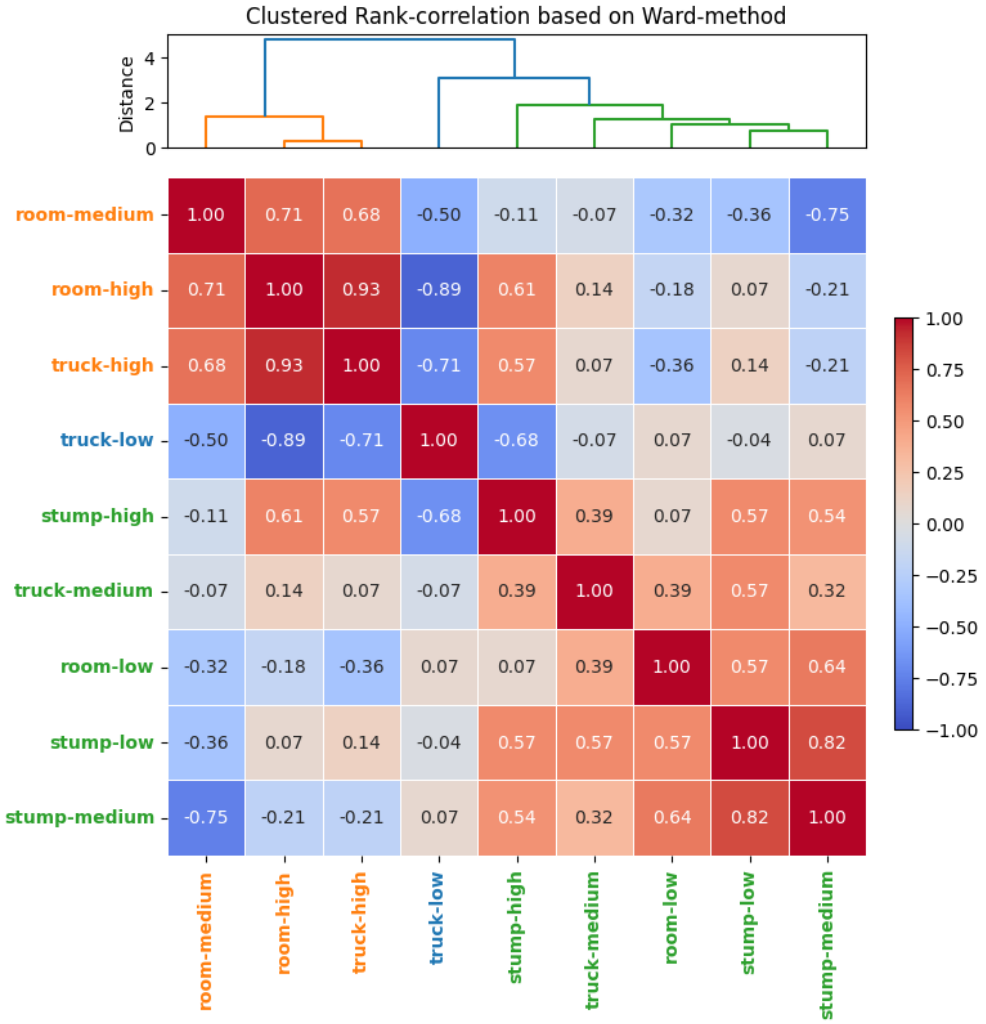

Figure 4.3: Rank correlation based on JOD, clustered using the Ward method [WJ63]

When looking at the clustering results, there are two pronounced clusters and an outlier. In the following paragraphs, these subsets will be referred to as described in the following list:

1. *Cluster Orange* contains room-high, room-medium, and truck-high.
2. *Cluster Green* contains all circumstances in the stump scene, as well as truck-medium and room-low.
3. *Circumstance Blue*, which is truck-low, appears to be an outlier.

Using these clustering results, new rankings can be established. By analyzing the ranking within similar subsets, we can determine whether a different trend is present in the data.

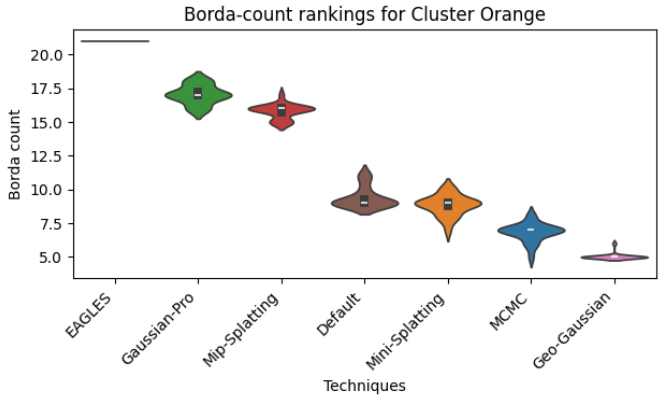

Figure 4.4: Borda count rankings for Cluster Orange

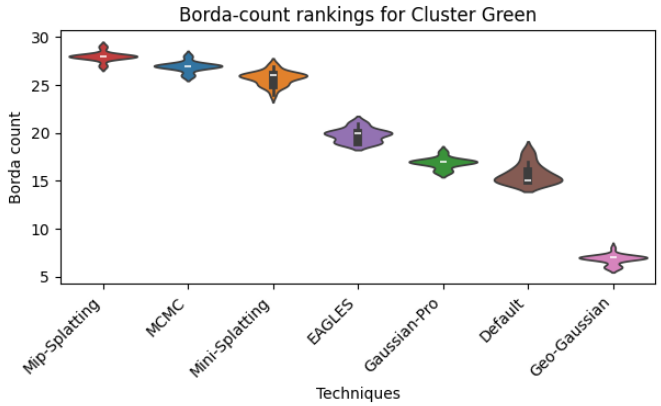

Figure 4.5: Borda count rankings for Cluster Green

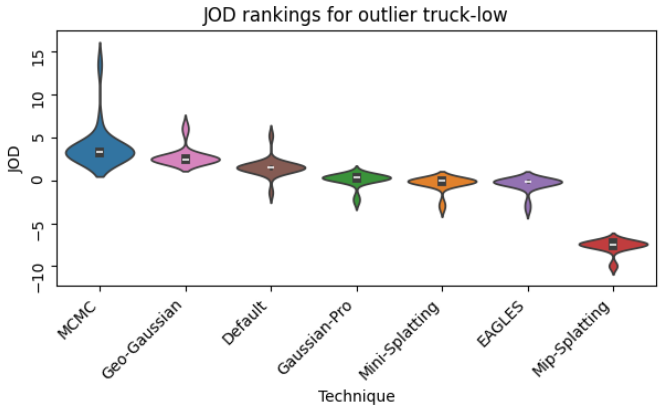

Figure 4.6: JOD-based rankings for truck-low

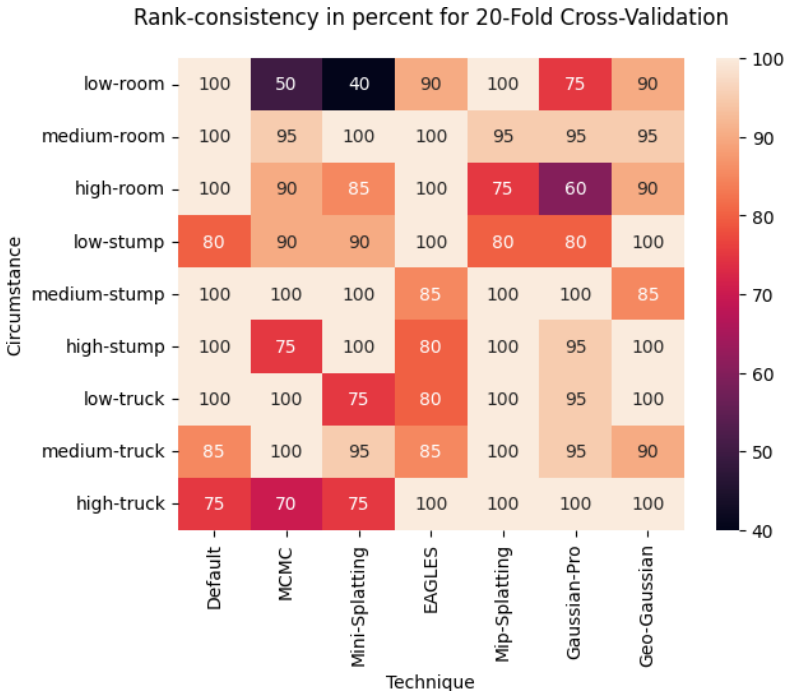

Figure 4.7: Rank consistency of JOD scores for 20-Fold Cross-Validation

The figures 4.4, 4.5, and 4.6 illustrate how diverse the rankings can be when considering different clusters. Cluster Green's ranking closely reflects the overall ranking shown in figure 4.1. In contrast, the ranking for Cluster Orange is completely different. Here, EAGLES is the unequivocal leader, while the suspected overall leaders, MCMC and Mip-Splatting, are ranked among the lowest. The truck-low circumstance completely breaks with previous trends, ranking Geo-Gaussian relatively highly and Mip-Splatting as the lowest overall. It essentially presents us with a complete reversal of the trends that can be found for the rest of the data.

4.1.4 Reliability

The subjective metrics represent an important part of the evaluation. They serve as the basis for answering both research questions. Therefore, the reliability of the subjective measures is of great importance.

An important aspect of the subjective scores is the resulting ranking. Reliable JOD scores should be internally consistent. To assess reliability, the variability of the rankings is investigated.

Figure 4.7 shows the rank consistency of every method across the different circumstances. The figure was generated by splitting all of the ratings using 20-Fold Cross-Validation. Each cell describes the percentage of folds in which a method received the same rank. A cell with a value of 100% represents a method that received the same rank across all

20 folds, while a cell with 40% represents a method that only had the same rank in a maximum of 8 folds.

The median rank consistency is 95% with a standard deviation of 12.93%. These measures indicate that the rankings derived from JOD are highly consistent. This suggests that the computed scores are reliable.

4.1.5 Insights

To summarize this section, it helps to recall a number of the insights from the subjective evaluation. The following list summarizes the most important findings:

- Mip-Splatting is the overall winner when considering the subjective evaluation, due to its consistency.
- MCMC and Mini-Splatting are ranked closely behind Mip-Splatting.
- MCMC is the best performer in the low and medium size budgets.
- MCMC is a straight-up improvement over the Default, when aggregating scores across the scenes.
- EAGLES is a strong contender for the high size budget.
- Geo-Gaussian is consistently ranked as the worst method.
- The rankings differ significantly across circumstances, indicated by the low SROCC of 0.097.
- Within different subsets of the data, the rankings can be completely different.
- The within-circumstance rankings are highly consistent across 20-Fold Cross-Validation, which indicates that the subjective scores are reliable.

4.2 Objective Results

The following sections will discuss the rankings of the techniques based on objective indicators. These metrics stem from the software component described in section 3.5.1. The exact values and aggregated rankings can be found in appendix D. First, the overall ranking will be described, then the rankings per size budget will be presented. The section will conclude with a discussion of the cross-circumstance rankings.

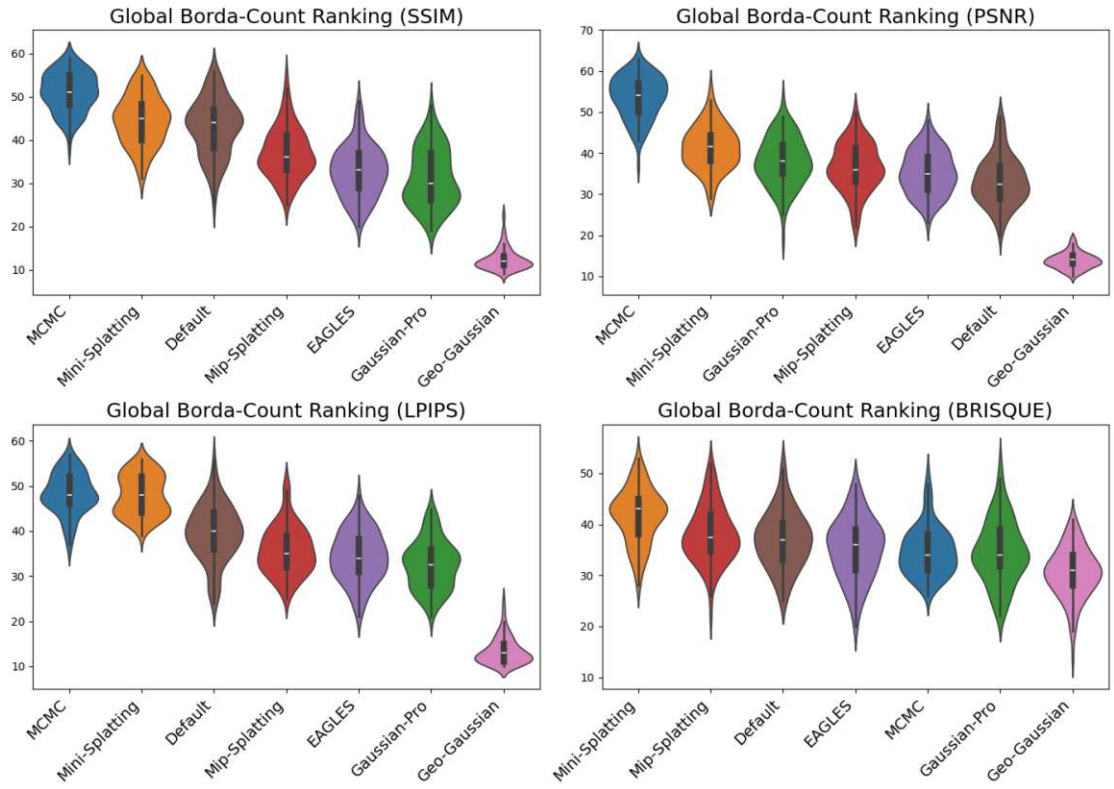

Figure 4.8: Overall objective rankings aggregated by Borda count and Bootstrapping, higher is better

4.2.1 Overall Objective Ranking

In order to give a robust estimate of the overall objective performance, Bootstrapping and Borda count are used. In total, 100 Bootstrap samples are drawn from the metrics. For each unique sample, circumstance, and metric type, a ranking is computed. Afterwards, the Borda count is used as a voting mechanism to collect the rankings across all of the circumstances, which leads to 100 Borda counts. Figure 4.8 shows the results as violin plots. The plots give a sense of the variability and distribution of the aggregated measure. This shows clearly that there is a degree of uncertainty in the global ranking.

The reference metrics show MCMC in a consistent leadership position. This indicates that MCMC works well in fitting the model to the camera path traced by the dataset. The only measure that does not rank MCMC as the highest is BRISQUE. As a no-reference metric with perturbed camera poses, it presents another perspective on the visual quality outside of the typical camera poses. BRISQUE instead elects the runner-up, Mini-Splatting, as the overall winner. Mini-Splatting also closely follows MCMC's performance according to the LPIPS metric. Both of these facts indicate that Mini-Splatting is a strong contender according to objective indicators. Default's performance is noteworthy, since it takes third

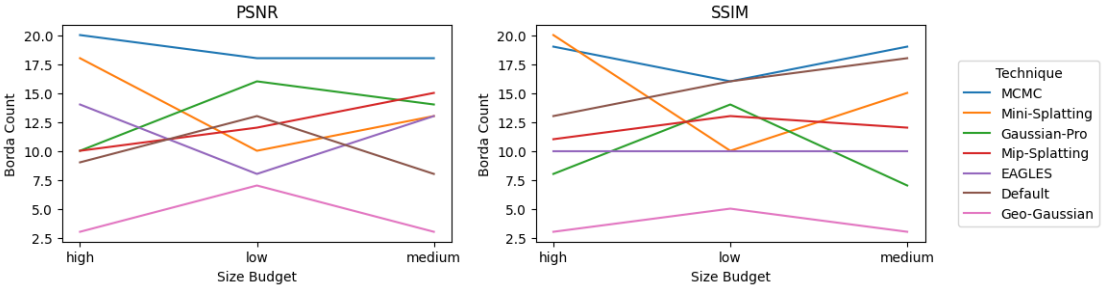

Figure 4.9: Development of objective rankings aggregated by Borda count across the size budgets, higher is better

place in three out of four metrics. The overall worst method seems to be Geo-Gaussian. Every single metric ranks it as the lowest performer.

4.2.2 Rankings per Size Budget

The overall ranking seems to favor MCMC, but how does the ranking look when drilling down to the individual size budgets? Figure 4.9 shows the aggregated rankings for the SSIM and PSNR metrics, as they develop with increasing model sizes. PSNR clearly favors MCMC, while the SSIM shows a more complex pattern. Here, MCMC is located at the top of the distribution, but its spot is contested by Mini-Splatting and Default. The figures show many lines, which are located very close to each other, for the middle of the leaderboard. This indicates that the metrics might not serve as an ideal basis for discriminating between the methods.

The same plots for the LPIPS and BRISQUE measures can be found in the appendix in figure D.1. It presents an inconclusive picture. Many of the methods are intermixed or observe rapid jumps in their aggregated rankings, which indicates instability.

What becomes clear for the SSIM, PSNR, and LPIPS metrics, is that MCMC is consistently one of the best performers. SSIM and LPIPS also present Mini-Splatting as the leader for the low-size budget. Geo-Gaussian is consistently ranked as the worst method.

4.2.3 Cross-Circumstance Rankings

One problem that becomes immediately apparent when looking across individual circumstances is that the rankings differ wildly. Figure 4.10 shows some prominent examples that vary both in terms of scene and budget size. In order to get a full picture of rank correlation, a higher-level view is useful. Figure 4.11 and 4.12 show how complex the correlation structure is when considering all circumstances and metrics. These plots follow the same structure as the correlation analysis presented in the previous section. The following paragraphs explore the data structure of the plots.

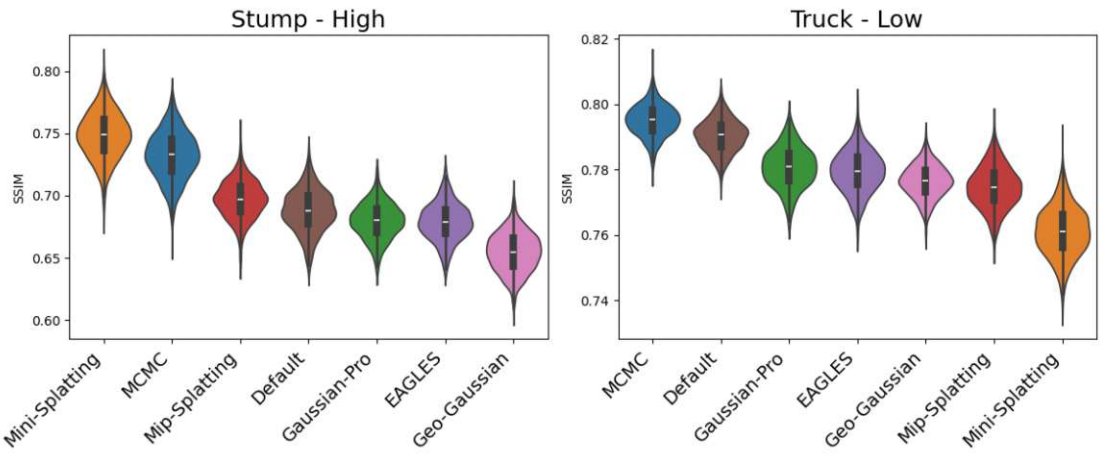

Figure 4.10: Examples of highly discordant rankings in different circumstances

The average SROCC for SSIM, PSNR, and LPIPS ranges between 0.375 and 0.488. This shows that there is some degree of ranking inconsistency for these measures. BRISQUE has the lowest average SROCC at -0.078.

The most consistent group is made up of circumstances belonging to the stump scene. For SSIM, LPIPS, and BRISQUE, all of the correlations with each other are above 0.9. This indicates that methods behave consistently across size budgets for this dataset.

Both SSIM and LPIPS result in a block of circumstances that are highly correlated. These include the room scene at the high and medium size budgets, as well as all of the stump circumstances. In comparison, the PSNR metric seems rather noisy. BRISQUE shows a tendency for higher within-scene correlations while having lower correlations across scenes.

4.2.4 Insights

To summarize this section, it helps to recall a number of the insights that could be gathered from the subjective evaluation. The following list summarizes the most important findings:

- MCMC is often the leading method according to SSIM, PSNR, and LPIPS across all size budgets.
- Mini-Splatting is a strong contender for the low-size budget, when considering the SSIM and LPIPS metrics.
- Geo-Gaussian is consistently ranked as the worst method.
- There are indications that the BRISQUE strategy performs consistently within circumstances of the same scene.

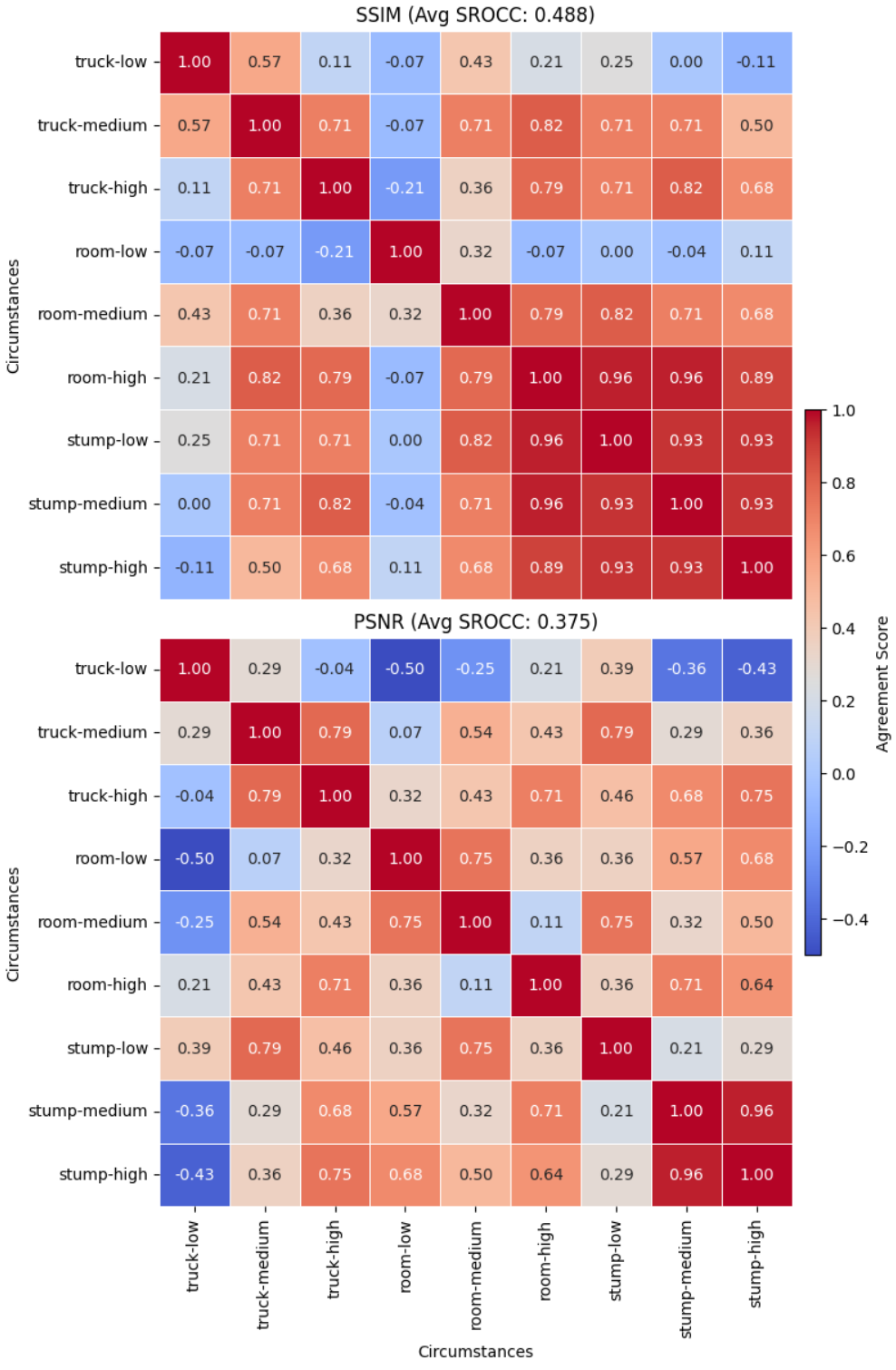

Figure 4.11: High-level correlation analysis for training metrics

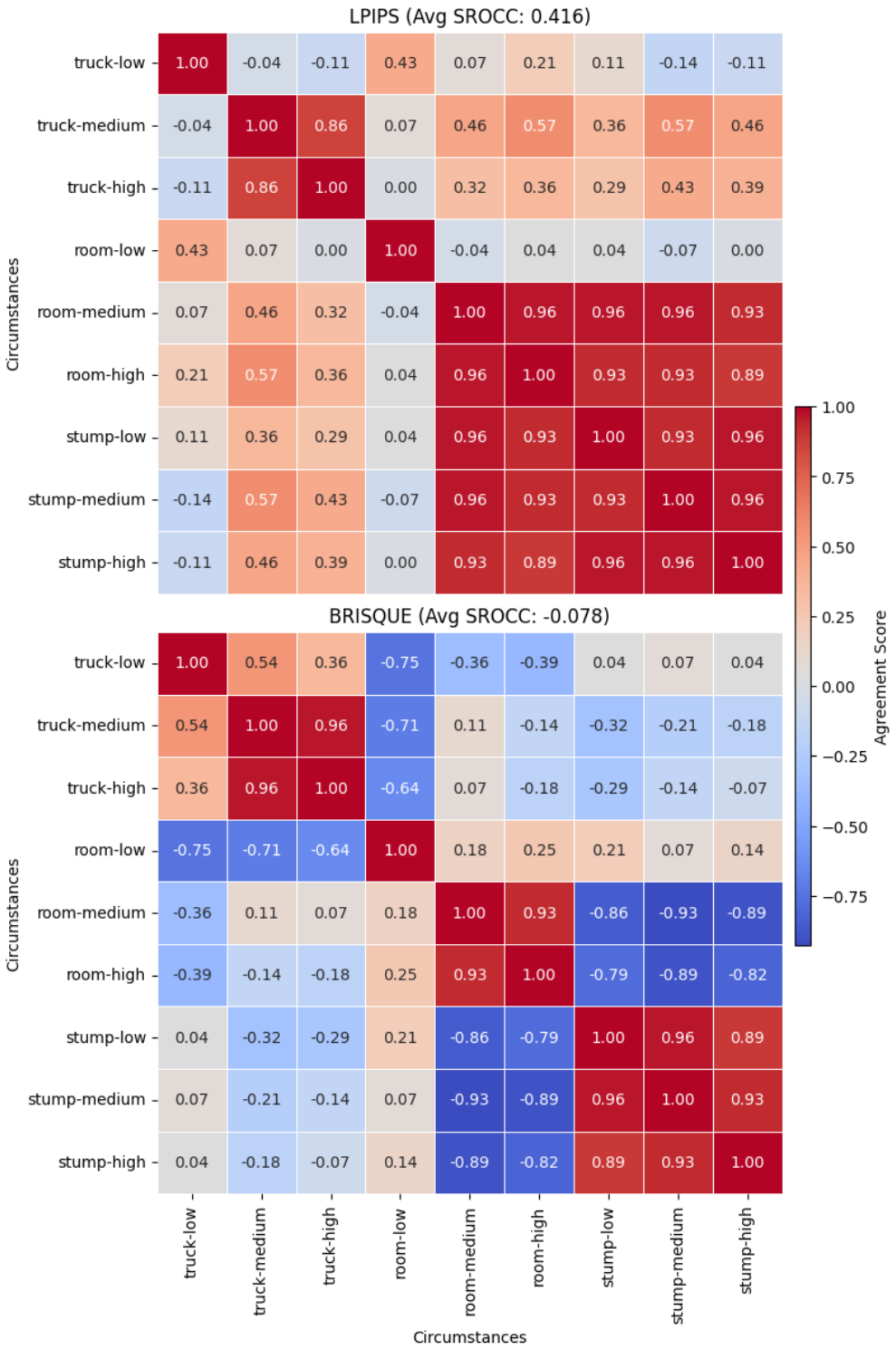

Figure 4.12: High-level correlation analysis for additional metrics

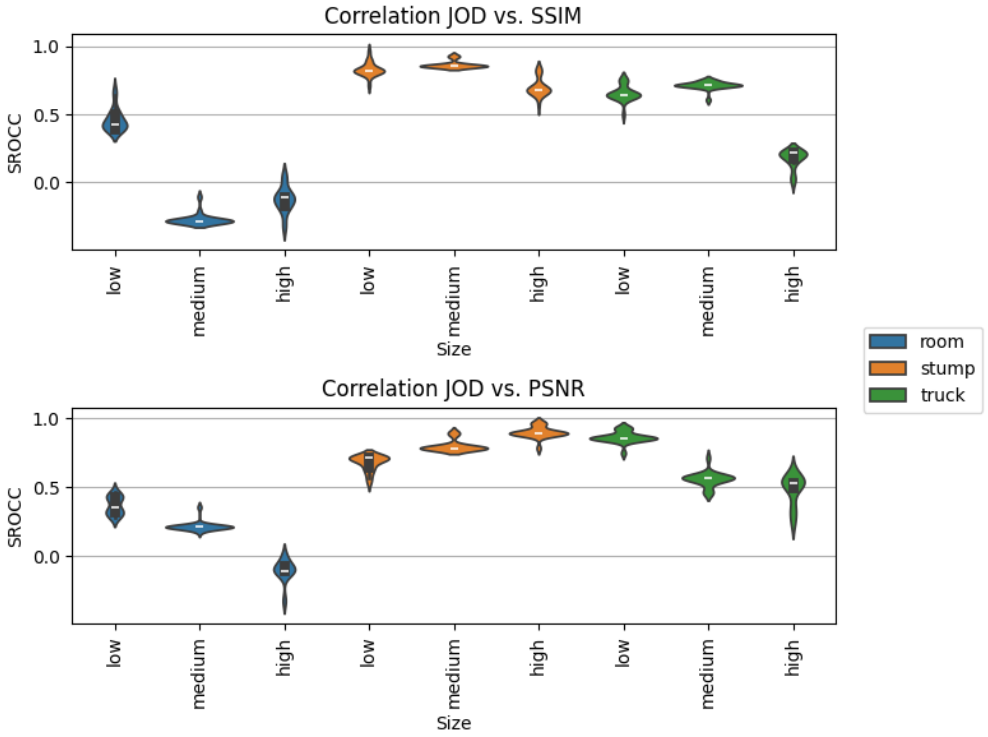

Figure 4.13: Correlation between objective measures and JOD based on 20-Fold Cross-Validation

- Stump circumstances generally correlate highly with each other.
- Both the overall Borda count aggregate as well as the ranking per size budget show a high degree of instability, indicating that the objective metrics are not an ideal basis for discriminating between the methods.

4.3 Joint Analysis

The previous sections presented the results of the subjective and objective evaluations. The subjective evaluation is based on the JOD scores, while the objective evaluation is based on SSIM, PSNR, LPIPS, and BRISQUE. The goal of this section is to analyze the relationship between these two evaluations. To this end, a correlation analysis is performed. Afterwards, the specific insights from the subjective and objective evaluations are contrasted.

4.3.1 Correlation Analysis

The objective quality measurement technique, involving test set PSNR and SSIM, aims to approximate true perceptual quality. To determine whether this approach is effective,

the objective and subjective measurements must be compared.

The SROCC was chosen as the correlation measure because the relationship between metrics is unlikely to be linear. The overall correlation between objective and subjective measures is very low, if not nonexistent. SSIM and JOD have an SROCC of 0.031, while JOD and PSNR have 0.169. This suggests little to no relationship between these different indicator types. To further investigate this issue, the correlation was computed within each circumstance. Figure 4.13 shows the correlation structure observed across circumstances for PSNR and SSIM.

Some circumstances show high correlations, while others exhibit low or even negative correlations. The SROCC mostly exceeds 0.5 when considering the stump scene, as well as the truck-low and truck-medium combinations. In these cases, the IQA strategies work in tandem. The room dataset shows more problematic results, with correlations even falling into the negative range.

The correlations for LPIPS and BRISQUE are presented in the appendix in figure E.1. For these measures, lower correlations indicate a better fit, as both metrics quantify some form of loss. The stump dataset also exhibits the best approximation performance. The other circumstances are very noisy and do not replicate any patterns seen in PSNR and SSIM.

To summarize the correlation analysis, some facets of the data display correlations similar to those found in other studies [LWH+24, YYX+24, QLC+24], while others, especially the room scene, show little or even negative correlation. This suggests that objective and subjective measures can noticeably diverge for certain scenes. Both LPIPS and BRISQUE generally performed worse at approximating the subjective measures.

4.3.2 Contrasting Insights

The metrics aren't the only things that can be contrasted in the joint analysis. In order to get a better understanding of the differences between the assessment methods, the insights from the subjective and objective evaluations can be compared. The following paragraphs highlight a number of aspects on which the two evaluations either align or diverge.

Overall Winner The subjective evaluation ranks Mip-Splatting as the overall winner, due to its consistency across different circumstances. MCMC leads across all objective metrics (SSIM, PSNR, LPIPS), while Mip-Splatting mostly occupies a spot in the middle of the leaderboard. MCMC is also ranked highly in the subjective evaluation, coming in at second place overall and being the best performer in the lowest size budget.

Best Method for Low Size Budget The subjective evaluation ranks MCMC as the best method for the low size budget, closely followed by the Default approach. The objective evaluation ranks MCMC and Mini-Splatting as the best methods for the low

size budget. This shows that MCMC is a strong contender for the low size budget in both evaluations.

Best Method for High Size Budget The subjective evaluation ranks EAGLES as the best method for the high size budget, while the objective evaluation ranks MCMC as the best method. This shows an example of divergence between the two evaluations.

Overall Worst Method Both evaluations rank Geo-Gaussian as the worst method. It is consistently outperformed by the other methods, regardless of the evaluation paradigm.

Cross-Circumstance Correlation The subjective evaluation shows a very low SROCC of 0.097, indicating that the rankings differ significantly across circumstances. The objective evaluation shows a higher SROCC of 0.375 to 0.488. This means that there is some consensus across the circumstances. In essence, this could mean that objective measures are more consistent, while subjective measures are more sensitive to the specific circumstances.

Rank Certainty The certainty of the rankings cannot be directly compared. The subjective metrics has clear indicators for reliable rankings and seems to serve as a consistent basis for discriminating between the methods. On the other hand, the objective metrics show some degree of instability. In the end, this comparison cannot be drawn directly. This is because the subjective metrics are based on a probabilistic process and leverage Cross-Validation for many visualizations. All the while, the objective metrics use bootstrapping to create a distribution of the rankings, which introduces more noise. Contrasting the discriminatory power of different metrics is outside the scope of this study.

4.4 Training Behavior

During the training process, a number of statistics are reported regularly. These measures give insights into how the optimization techniques perform. The recorded metrics are SSIM and PSNR for both the train and the test set. The train statistics are logged every iteration, while a full run of the test set is conducted every 500 iterations. Because of its heavy memory and performance costs during runtime, the LPIPS metric was excluded from the logged metrics. When it comes to the densification behavior, the number of created and deleted splats, as well as the current splat count, is reported in every iteration when densification takes place.

Previous sections shed light on the performance of the different methods. But how is this performance achieved? By looking at the training behavior, a better understanding of what truly differentiates the methods can be achieved. The following section will analyze how the performance curves develop during training. Then the densification behavior

will be presented. Lastly, a broader summary of the diverging training behavior will be given.

4.4.1 Performance Curves

During training, the model performance develops as the iterations progress. Figure 4.14 shows that the performance curves can vary drastically depending on the optimization procedure. The following paragraphs will look at a number of summary statistics of these curves.

Curve Smoothness

As can be seen in figure 4.14, the performance curves are not always smooth or monotonic. There are clear valleys, which are caused by mechanics like pruning, depth reprojection, and the opacity reset. The question arises, whether the methods distinguish themselves by the smoothness of their curves.

In order to model the smoothness of the test performance curves, a 5th-degree polynomial model was fit for each model, metric, and circumstance. The R^2 of the model tells us how well the relatively smooth approximation captures the ground truth. This therefore serves as a measure of smoothness. A number of example visualizations of the fitted models can be found in the appendix in figure G.1.

Figure 4.15 shows a bar plot of the R^2 values. The average R^2 value for each method is marked above the bars as a percentage value. Most methods have a relatively smooth curve, according to this technique. Mini-Splatting is the only outlier, which means that the erratic changes that can be seen in figure 4.14 are a consistent feature of the method. This means that Mini-Splatting's optimization mechanisms have a noticeable effect on the stability of its performance throughout the training process. It's clear that it's the most erratic and unpredictable method, but its relative performance remains high, which indicates that the method is able to recover from these instabilities.

Metric Correlation

Figure 4.14 shows that SSIM and PSNR are generally correlated, though there is also an indication of some exceptions. SSIM and PSNR express different priorities of the training process. A high correlation of the metrics tells us that pixel-level and structural performance measures show similar variations. This means that the optimization procedure manages to jointly represent larger scene elements, while also improving on smaller details. This is generally a good sign, since both of these aspects are important for improving the overall quality of the model.

Figure 4.16 shows the correlation between SSIM and PSNR for both the train and the test set. The average correlation value for each method is marked above the bars as a percentage value.

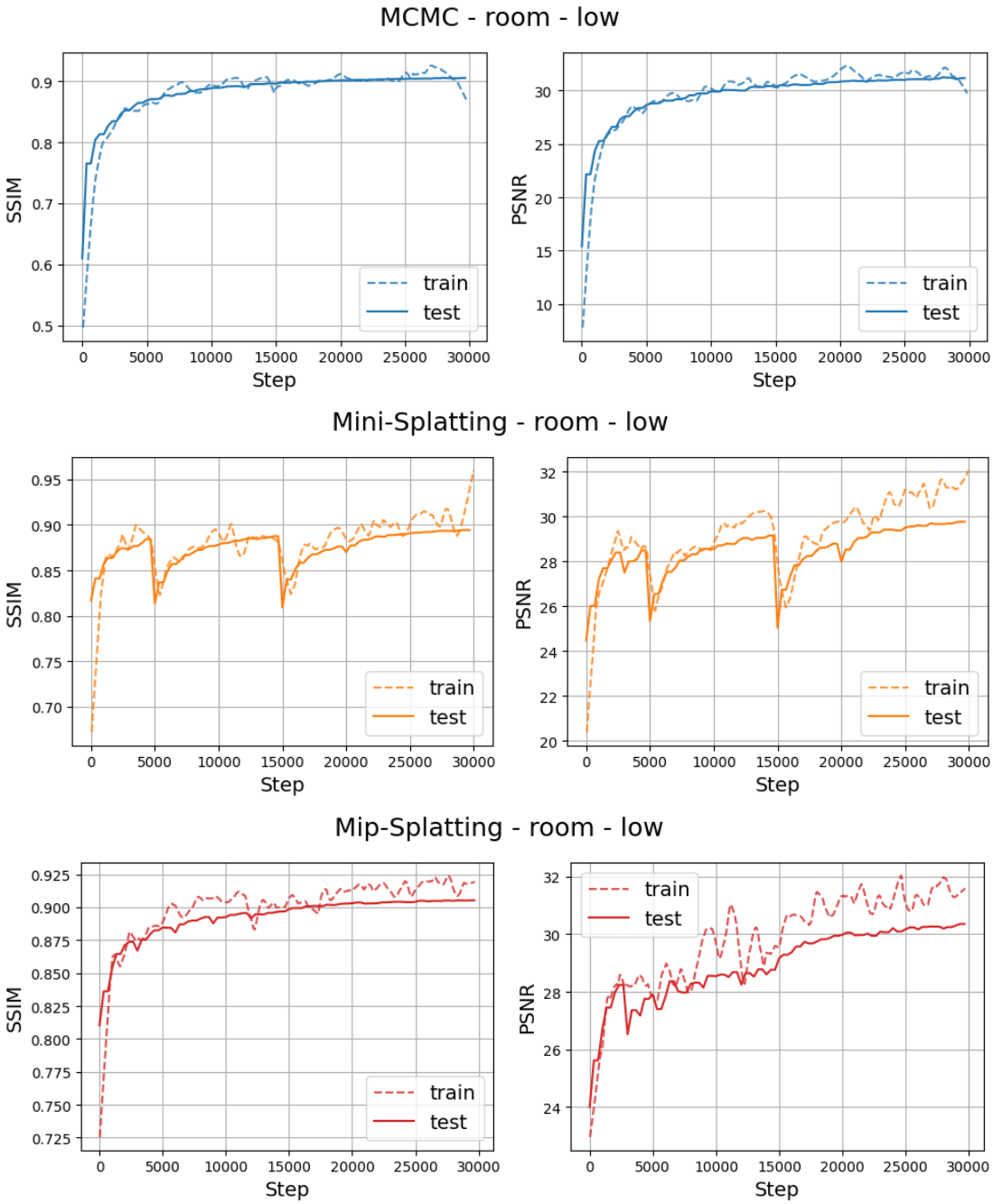

Figure 4.14: Performance curves during training

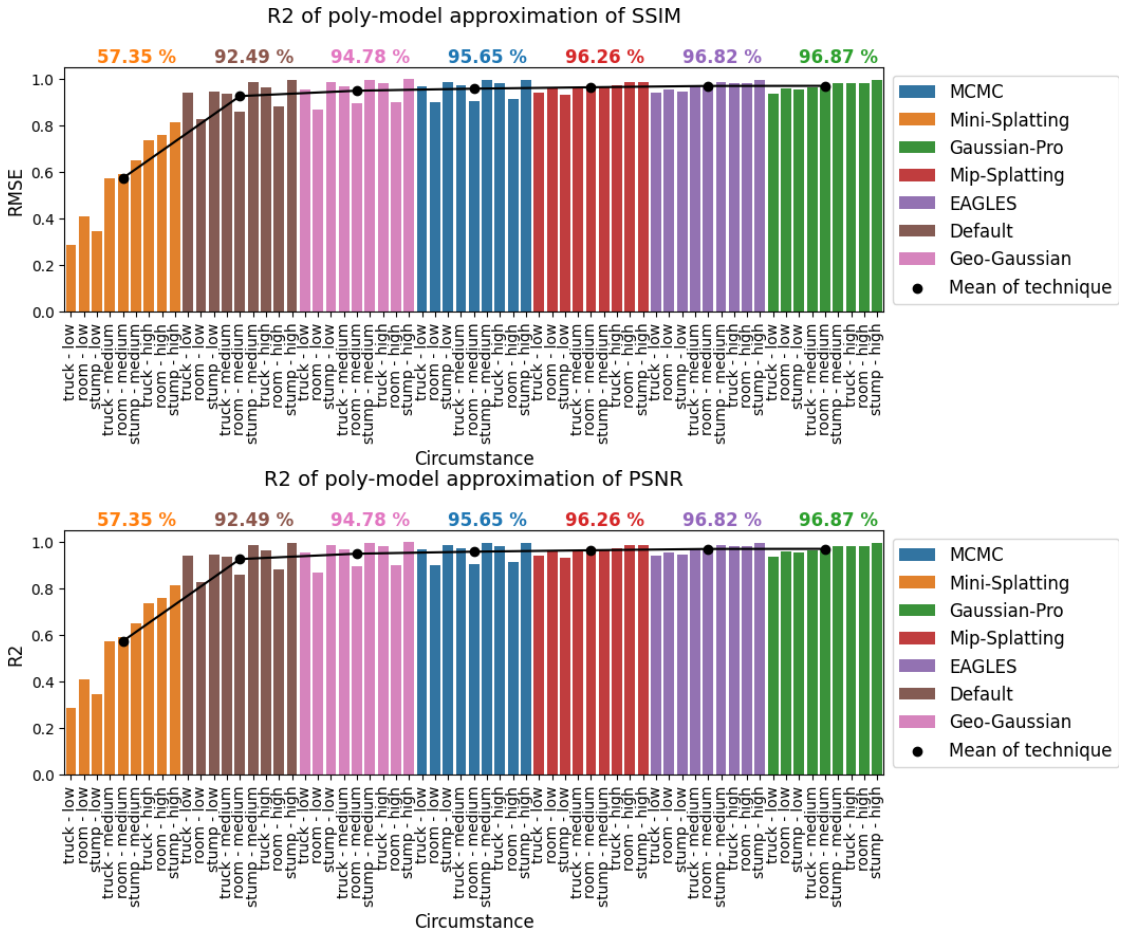

Figure 4.15: R^2 of 5th-degree polynomial model as a measure of smoothness of SSIM and PSNR curves

A low correlation is not necessarily a bad thing. Methods like EAGLES employ a coarse-to-fine strategy, which specifically aims to capture larger scene elements before focusing on details. This could explain why its correlation is the lowest out of all the methods.

The test set correlation is generally higher than the train set correlation. This is a good sign, since it actually indicates a stabilizing effect when the model is evaluated on the test set. This could be interpreted as a positive sign for generalization performance.

Loss of Performance

GS methods rarely use checkpoint mechanics. The reason for this is that checkpoints would incur large storage requirements and long write operations. Therefore, the model is simply exported at the end of the optimization procedure without any regard to the best score throughout the training process. This can lead to situations where the final model's performance is worse than the best model achieved during the training.

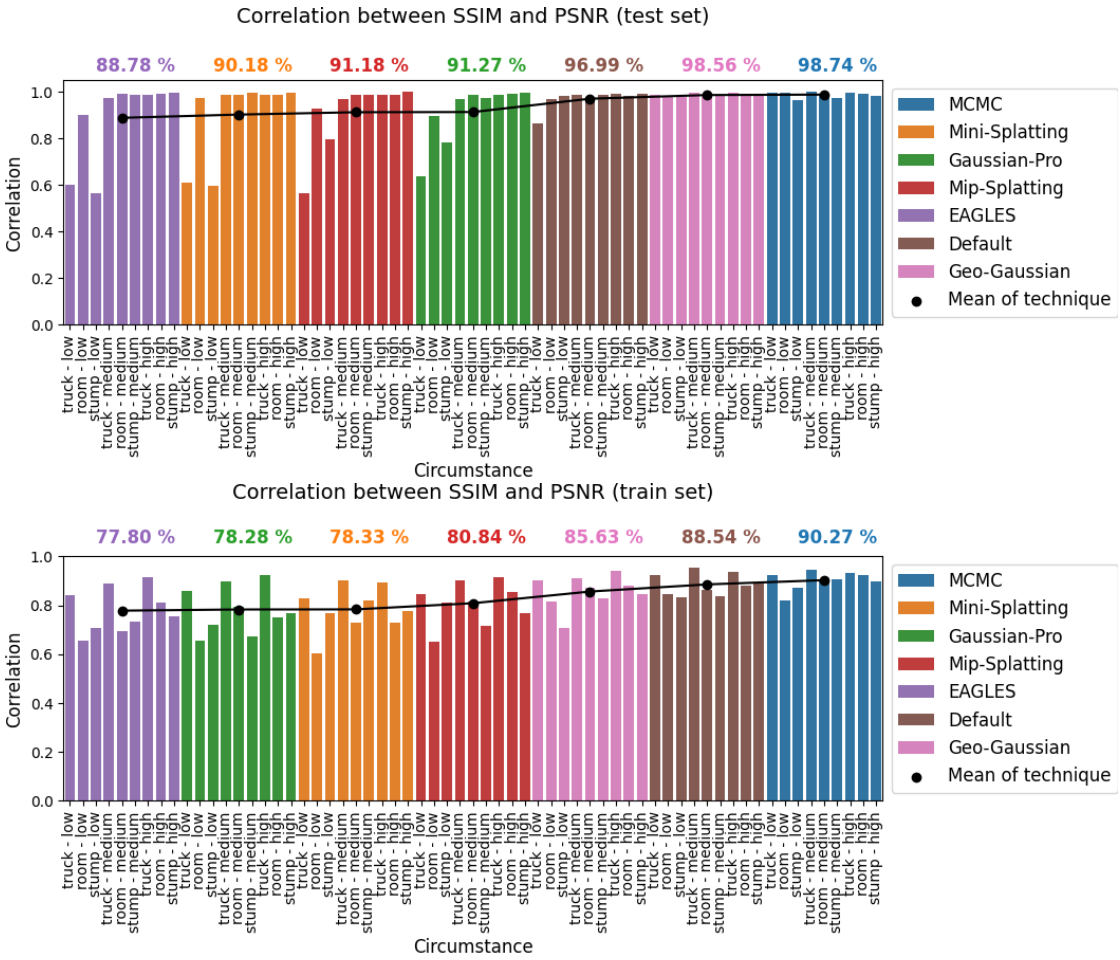

Figure 4.16: Correlation between PSNR and SSIM during optimization process

Figure 4.17 shows the disparity between the best and the final value of the measured metrics. It is important to note that the scale of the difference for both PSNR and SSIM is relatively small. In rare cases, Mini-Splatting shows pronounced losses in both metrics. This speaks to the fact that the method provides multiple relatively disruptive mechanics in order to change the structure of the model. It is also important to note that these cases only occur for models with the low size budget, which hints at the source of the instability being the size constraint. The other differences seem unremarkable given the scale of the data.

4.4.2 Densification Behavior

Each of the optimization methods has a unique densification behavior. It is characterized by the splats that are deleted, created, and the splat count.

Figure 4.18 shows a clear example of the densification behavior of MCMC when running

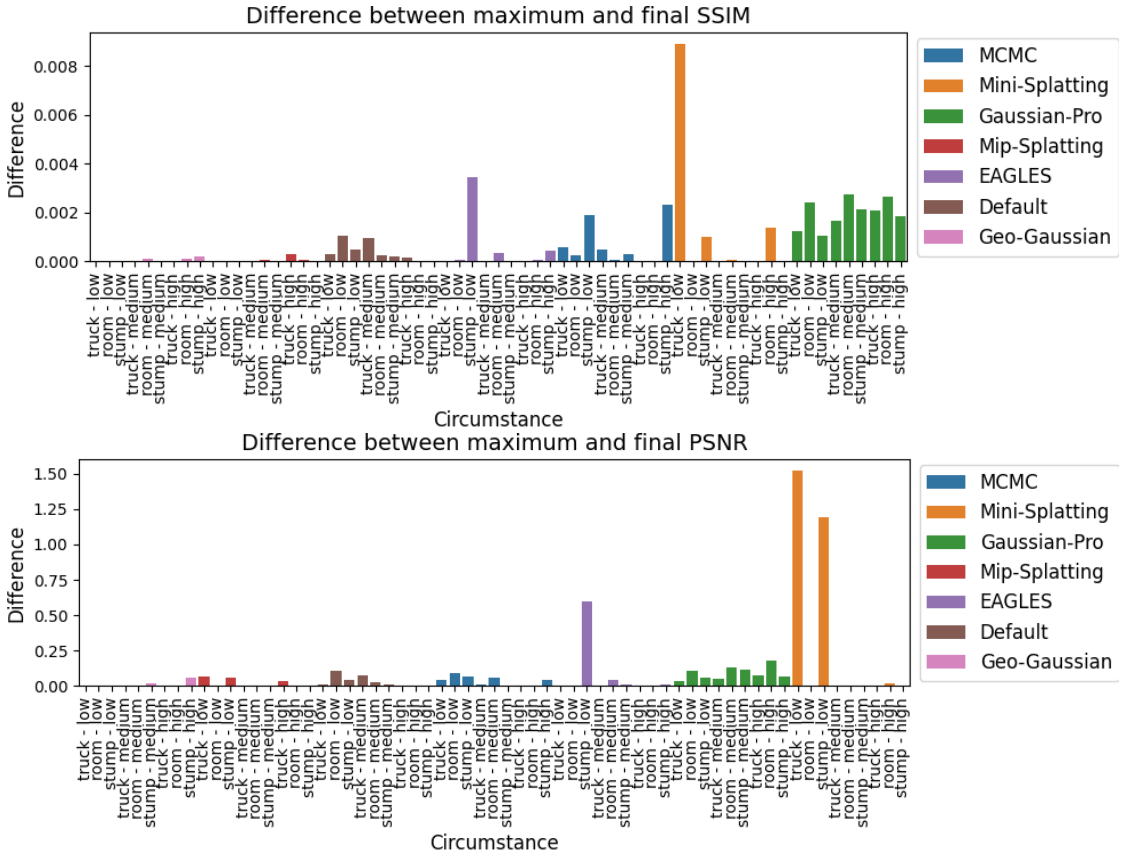

Figure 4.17: Difference between best and final metrics as an indicator for lost performance

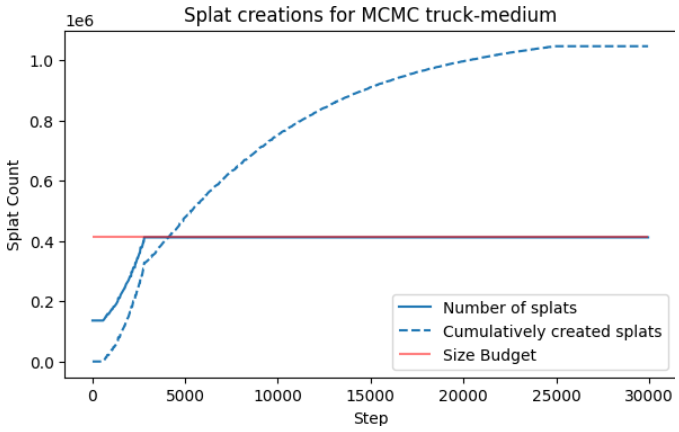

Figure 4.18: Densification continues despite the size limit being reached

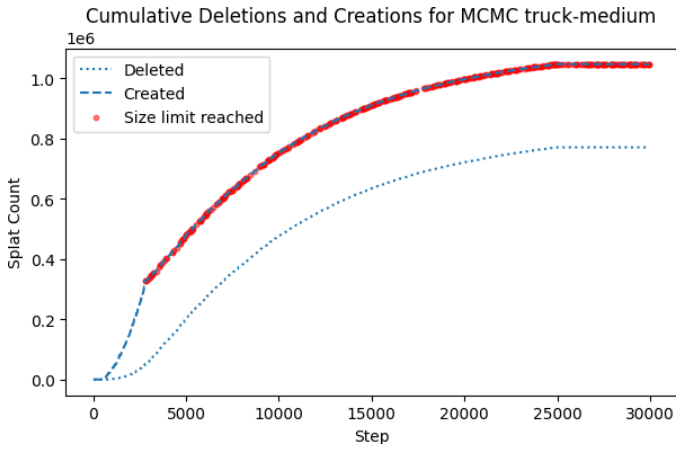

Figure 4.19: Deletions make room for new splats despite the size budget

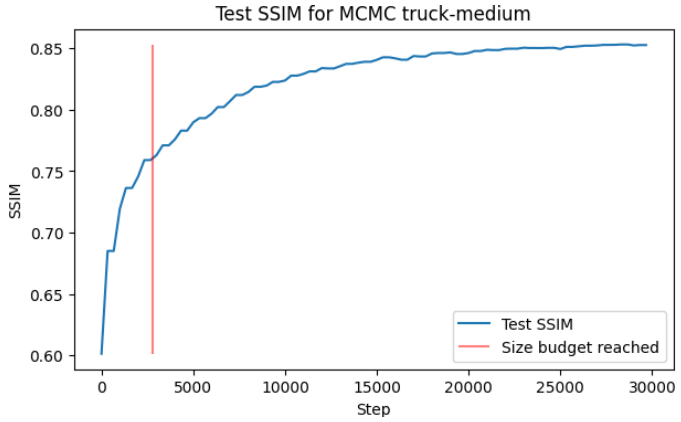

Figure 4.20: Test statistic increases even after the size budget is reached

on the truck dataset at the medium size constraint. The size budget is quickly reached, but the densification does not stop. Instead, splats are constantly being created. Figure 4.19 shows how deletions make room for new splats even if the size limit is reached. The test set SSIM also increases despite the constraints imposed by the cap, as shown in figure 4.20. This is congruent with the findings of Bulò et al. in their revision of Densification [BPK24].

MCMC has a very smooth profile when it comes to cumulative deletions and creations. They occur throughout the training process, but the rate of change progressively decreases. Other techniques have different mechanisms that lead to more erratic changes. Mini-Splatting, for example, employs multiple pruning stages. This leads to rapid jumps in the number of splats, as can be seen in figure 4.21, which explains the valleys that can be seen in the performance statistics.

The following sections will explore the densification behavior in more detail. Two types of summary statistics will be presented, which will help to characterize the behavior of

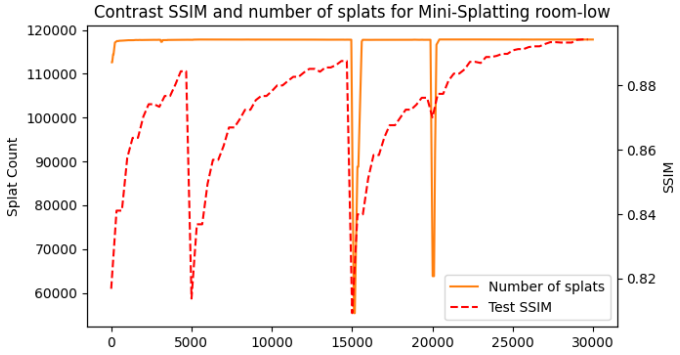

Figure 4.21: Pruning stages drastically decrease the number of splats

the different methods.

Magnitude of Deletions and Creations

As described earlier, the optimization continues despite the size limit. One way the methods distinguish themselves from each other is in how many deletions and creations are made in total. Figure 4.22 shows that there are clear differences in these statistics when comparing the different methods. The mean number of deletions and creations is marked on top of the plot for each method.

There is a clear ordering of the methods when it comes to the characterized densification activity. MCMC and Mini-Splatting are the leaders, while the default method actually shows the lowest numbers. This indicates that the newly introduced mechanics in the splatting heuristics lead to more variability in the splat composition.

Usage of Splat Budget

EAGLES is an optimization procedure that uses a more gradual pruning approach, which can lead to a part of the available budget going unused. As can be seen in figure 4.23, the number of splats sometimes slowly dwindles after 15K iterations. Despite this, the performance metrics do not decrease.

To get a general sense of the splat budget's usage during training, the area under the curve for splat count was computed. The value was normalized by the respective budget. Figure G.2 is presented in the appendix and shows the results of this calculation.

Models with a low size constraint have a higher usage. This is to be expected, since their budget is more restrictive and therefore more likely to be filled. The visualization also confirms the observation that could be made in figure 4.23. EAGLES has the lowest overall usage, which indicates that its influence pruning strategy has a profound impact on the number of splats.

The gradual pruning strategy of EAGLES leads to a significantly lower splat count than is imposed by the size budget. Despite this, the performance seemingly does not

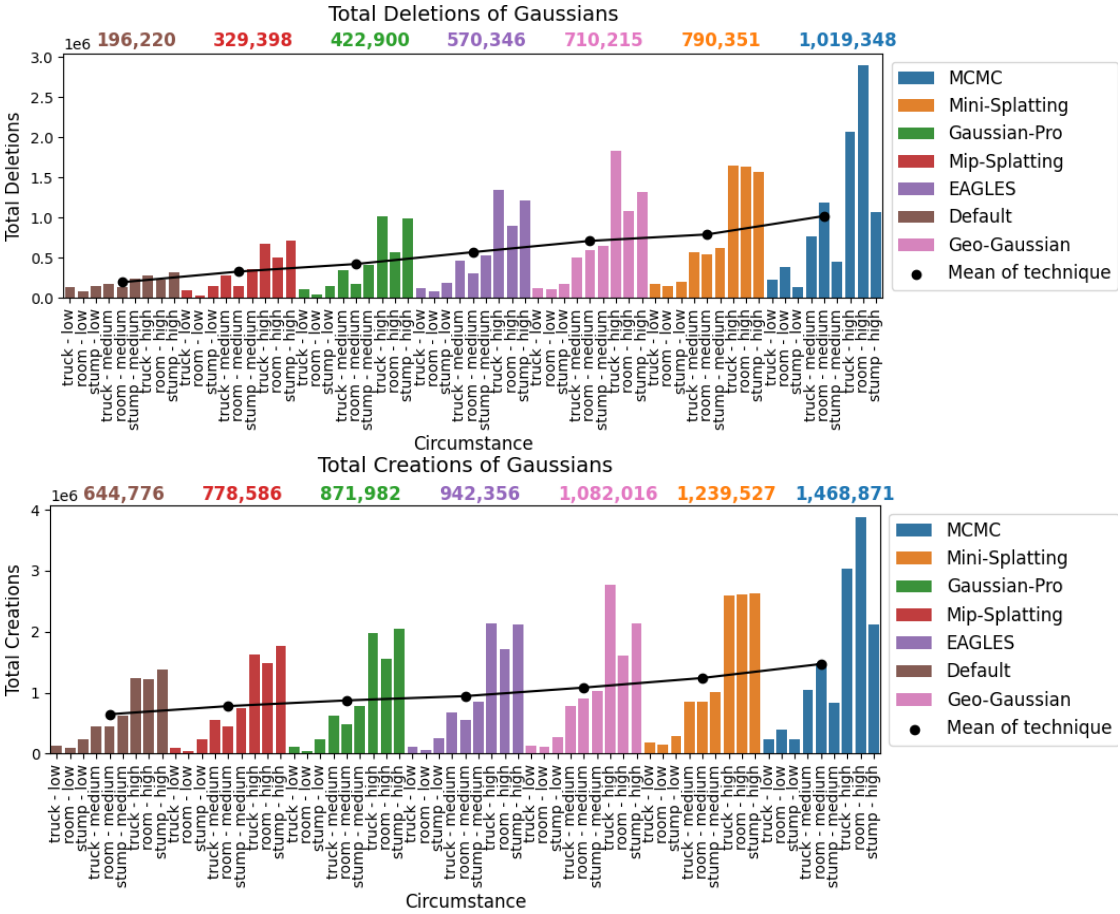

Figure 4.22: Densification behavior characterized by total number of creations and deletions

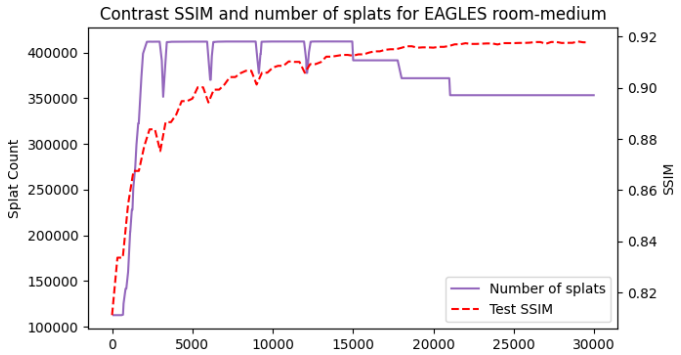

Figure 4.23: Pruning stages drastically decrease the number of splats well below the budget of 412235 primitives

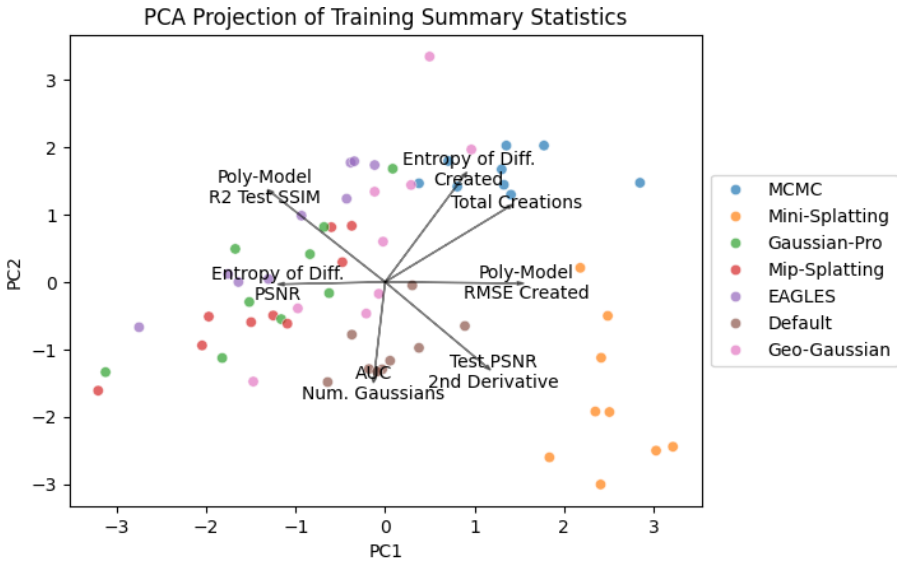

Figure 4.24: PCA of the extended set of summary statistics

suffer. This shows the potential of pruning mechanisms in shrinking the model size. It also indicates that the model could have benefited from a larger pool of splats, which exceeds the imposed size budget. These candidate splats could then be pruned until the desired model size is reached. The problem with this approach is that it is not entirely deterministic, since pruning does not guarantee that a desired splat budget will eventually be reached. This insight still opens up an interesting avenue for future research.

4.4.3 Training Statistics Summary

Using Principal Component Analysis (PCA), a number of the curve statistics can be summarized to make general conclusions about the model training. The different summary statistics are explained in the appendix in table G.1. Figure 4.24 presents the resulting biplot of the first two principal components. The plot shows that the methods have clear differences, even based on the relatively simple summary statistics. Especially Mini-Splatting, MCMC, and Default are easily separable. EAGLES, Mip-Splatting, Gaussian-Pro, and Geo-Gaussian are located in a diffuse cluster. The takeaway from this visualization is that the adjusted methods show clear differences in their training behavior compared to the baseline.

4.4.4 Insights

To conclude this section, a number of insights about the training statistics are formulated:

- Methods keep exchanging splats and improving performance, despite the introduction of budgets (see also [BPK24]).

- All methods show characteristic differences in their overall training behavior compared to Default.
- Especially MCMC's and Mini-Splatting's overall training behavior is distinctive.
- Mini-Splatting's pruning behavior can lead to irrecoverable losses in performance in some circumstances.
- SSIM and PSNR correlation is high for MCMC, Default, and Geo-Gaussian, indicating a balanced trade-off between structure and detail.
- EAGLES shows a slightly lower correlation between SSIM and PSNR in many instances, hinting at its coarse-to-fine approach.
- In general, MCMC deletes and creates the most splats, despite having only the opacity reset as a pruning mechanism.

CHAPTER 5

Discussion

5.1 Interpretation of Results

The data analysis yields a number of insights about the training and the rankings. Section 4.4 shows that the methods behave differently from the Default heuristic. The objective metrics generally favor MCMC, Mini-Splatting, and Default. A problem emerges when looking at the correlations between different circumstances. Many rankings aren't highly correlated, which indicates that different circumstances favor different techniques.

The subjective metrics also paint a complex picture. At first glance, the Borda count ranks Mip-Splatting, MCMC, and then Mini-Splatting as the best methods. When considering the SROCC between circumstances, it becomes clear that there are highly divergent rankings across the evaluation. The commonalities in rankings neither show trends when looking only at a single size budget nor a single scene.

The potential for different rankings is also reflected when looking at the distribution in figure 5.1. Mip-Splatting is a consistent measure. MCMC and EAGLES excel in different circumstances, while showing occasional weak points. Mini-Splatting is a very inconsistent measure. These tendencies are similar to the ones found by analyzing groups based on ranking similarity, as done in section 4.1.

So, how can the first research question *"Which optimization procedure delivers the best visual fidelity when the number of splats is constrained?"* be answered? To gain a better understanding of how the rankings came to be, a closer look at the generated models is essential. The following paragraphs will present some of the visual features of the models and use this analysis to answer the first research question.

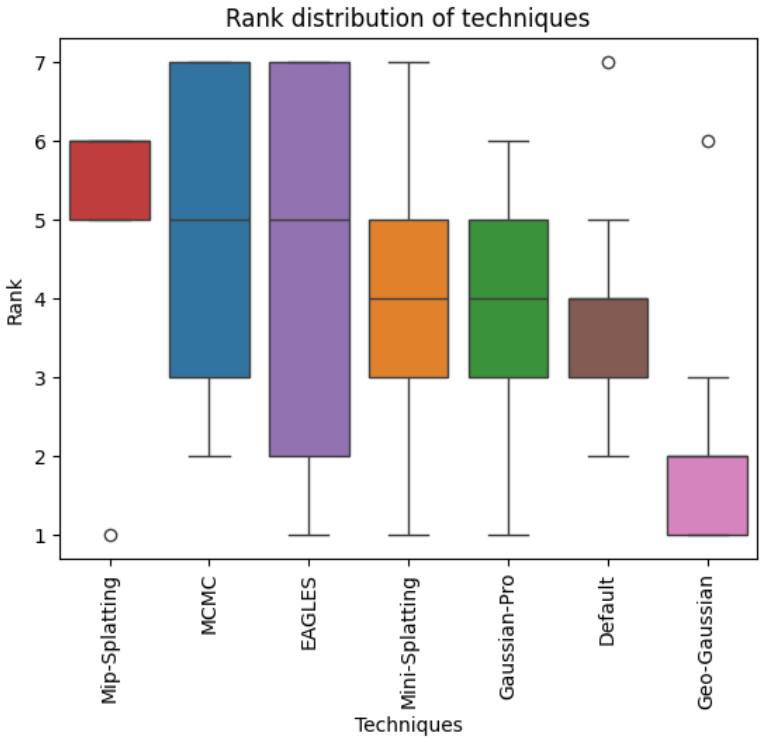

Figure 5.1: Overall rank distribution of the chosen methods

5.1.1 Qualitative Analysis

In order to present visual examples, a clear strategy is being followed. To analyze a method's performance and its stability, examples from both its best- and worst-ranked models will be shown. Each example will also be contrasted with another model from the same scene and size budget, which is either ranked higher or lower than the candidate model. This replicates the experience study participants had and shows the range of a technique's output. The rankings are taken from the JOD scores that can be found in the appendix in table C.1. All of the discussed renders are displayed in appendix F since their inclusion here would take up too much space. All images will be referenced at the point at which they are being discussed in the following paragraphs.

The subjective data analysis highlights four techniques that are of particular interest. These techniques are Mip-Splatting, MCMC, EAGLES, and Mini-Splatting. Geo-Gaussian will also be included in order to understand its position as the lowest-ranked technique.

The most consistent method in the subjective rankings is Mip-Splatting. Positively ranked examples are characterized by enhanced detail in areas, which appear as weak points in other methods' splat models (F.1 and F.2). Despite its stability in almost all circumstances, it also produced a model that is very blurry and plagued by artifacts, while other models created much better results (F.3 and F.4).

MCMC is one of the runner-ups after Mip-Splatting. Its potential lies in resolving far-away detail (F.5 and F.5). It struggles in the room scene, where it produces noisier surface representations (F.8). Overall, this method is very stable, when considering qualitative examples. It is the model that generally shows the least amount of floating artifacts (F.7).

EAGLES shows enhanced performance when it comes to distant details (F.9 and F.10). These gains are, however, not guaranteed, as the relative performance can also struggle in certain circumstances (F.11 and F.12). The method is characterized by its relative stability regarding floating artifacts, though there are instances where it looses out to MCMC (F.7).

Mini-Splatting performs especially well on the stump scene (F.11 and F.13). Here, the resolution of distant details and dense foliage is unparalleled. The method also shows problems, similar to MCMC. Surface representations can be noisy, which is especially apparent in the room scene (F.14).

Geo-Gaussian is the lowest-ranked method in many of the circumstances. The reasons for this are apparent when looking at many of the models. It struggles with blurry regions and distant details (F.15). Some models have issues regarding high-frequency artifacts (F.16) or larger floating artifacts (F.17). Despite these shortcomings, there is a circumstance where Geo-Gaussian's potential is highlighted. In the truck scene at the lowest size constraint, the technique is ranked second. It performs exceptionally well in distributing the splats along the geometric details of the scene (F.18).

The subjective analysis shows that there are trends regarding the technique's performance. When looking at the model's actual output, it becomes apparent that there is always an exception to these trends. This could be due to different reasons. Either the circumstances are well-chosen and provide diverse challenges, or maybe GS optimization is simply a noisy and turbulent process that does not always produce reliable results. In any case, the answer to the first research question is that there are three leading methods: Mip-Splatting, MCMC, and EAGLES. The performance varies by scene, size budget, and most likely other latent factors. Overall, MCMC can be seen as the winner, due to its stability and its subjective performance in the smallest size budget. The lowest size constraint represents an extreme circumstance. MCMC's ability to produce high-quality models even under such strict limitations makes it a premier choice for GS on a budget.

5.1.2 Correlation Interpretation

Section 4.3 shows that the correlation between objective and subjective measures isn't always a given. What is surprising, is that some circumstances show relatively clear correlations, while others show low or even negative ones.

The room scene has the lowest correlations across the board. Figure 5.2 shows an annotated version of the COLMAP scan of the room scene. It becomes clear that the cameras are clustered around two distinct points in space. They are located at opposite

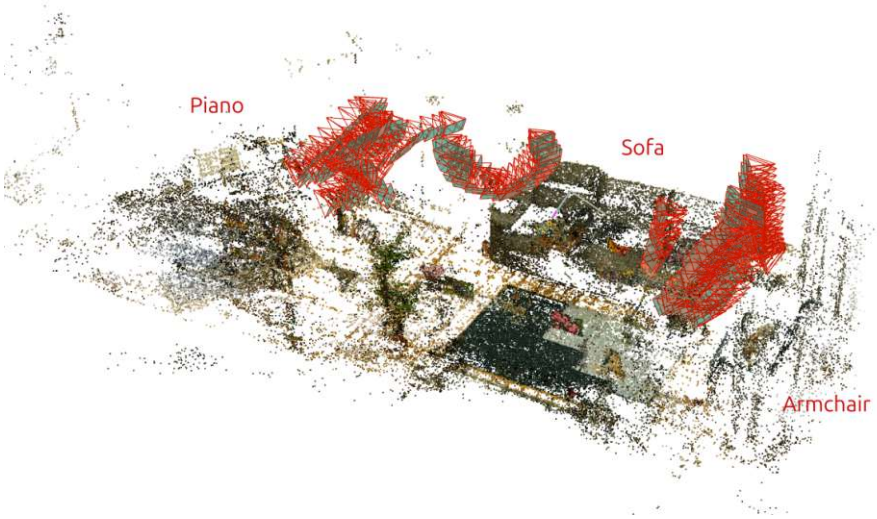

Figure 5.2: Camera distribution of the room scene

ends of the room and are generally angled towards the other side. These camera locations do not capture a full 360-degree camera path. This is in stark contrast to the COLMAP scans of the truck and stump scenes, which can be viewed in the appendix in figures H.1 and H.2. They clearly show camera paths that revolve around a central point, which is the object of interest.

The evaluation tool uses orbital camera controls, which enable the intuitive rotation of the view around a central point in the scene. The most straightforward action is to explore the scene using this rotational approach, as it is bound to the mouse movement. This orbital control scheme mirrors the camera setup of the stump and truck scenes. The cameras, which were used for the objective metrics, therefore "see" the same thing as the survey participants. This would explain, why the two scenes with orbital COLMAP camera locations generally have a positive correlation.

Research question two is *"Do objective and subjective image quality assessment measures align across different techniques and model size restrictions?"*. The findings indicate, that subjective measures align with objective metrics, if the calibrated camera paths mirror the camera positions experienced by participants. If this is not a given, viewers have to base their preferences on other indicators, which means that the ratings are affected by factors, that are likely not in view of test set cameras. Sadly, this research objective can't be answered with full confidence. More research into answering this question would be needed, since this study's scope is relatively small. That said, other research already shows a strong correspondence between subjective and objective measures, when the viewing direction cannot be manipulated by the participants [LWH+24, MRAQ24, XYY+24].

5.2 Limitations

The primary limitation of this study is its relatively small scale. Only three datasets are used in the evaluation. While the data was carefully selected to balance scene diversity, it still restricts the generalizability of the findings. The room scene is a particular outlier, where correlations between subjective and objective metrics were lower than expected. The inclusion of the room scene might introduce some skewness in the global rankings. This impacts the generalizability of the findings to some extent, which means that further research is needed to confirm the results across a wider range of datasets.

Additionally, the study only relies on within-circumstance rankings, where comparisons were fixed in terms of dataset and size. While this approach ensured fairness and consistency, it also had the side effect of amplifying small perceptual differences, which might be less relevant in a real-world setting. Despite this, the ranking approach was still the best choice for the microservice-based architecture of the evaluation platform. Cloud-function runtimes constrain the performance of sampling algorithms like ASAP, which only function on larger-scale evaluations by running on dedicated hardware.

Another limitation stems from the study design. The evaluation was conducted with a pool of mostly inexperienced users (see appendix A), meaning their perceptual judgments may not align with those of domain experts. The evaluation was carried out remotely, so there was no control over participants' viewing environments. This includes factors such as screen quality, ambient lighting, and display settings. These uncontrolled variables introduce noise into the results and could partially explain inconsistencies in user preferences.

Despite these constraints, the study still provides valuable insights into model ranking under constrained budgets. It also accentuates model performance in a real-world setting.

CHAPTER 6

Conclusion

6.1 Summary

This study investigated the visual fidelity of GS optimization methods under clear model size constraints. Existing approaches primarily relied on a hard cap that abruptly halted densification, which is a suboptimal approach for introducing model size restrictions. Six novel extensions to existing methods are introduced, and implementations are provided in appendix I. This contribution is significant and is similar to Bulò et al.'s Revising Densification [BPK24].

A comparative evaluation using three datasets and three size budgets was conducted. It contrasts objective metrics with subjective user preferences, gathered with a custom evaluation tool. The results indicate that dataset viewing paths and camera controls need to be aligned in order to retrieve meaningful correlations between subjective judgments and objective quality metrics. This means that viewers should experience the same perspectives as the camera path used during model generation and evaluation.

The best models are Mip-Splatting, EAGLES, and MCMC. Mip-Splatting consistently achieves the high performances across all circumstances. It demonstrates good stability and perceptual quality. EAGLES and MCMC also produced high-quality results, occasionally outperforming Mip-Splatting in specific scenarios. The qualitative analysis indicates that the two methods are also less artifact-prone than the others. MCMC's subjective performance for the low and medium size budgets is particularly noteworthy, and it therefore presents the best choice for GS on a budget.

6.2 Reflection and Future Work

Promising improvements could be achieved by developing hybrid optimization techniques, which leverage the strengths of different approaches. An example would be the extension

of MCMC with EAGLES-style pruning, while incorporating a stochastic perspective to avoid discarding important structural details. This could unlock special synergies, by preserving splats that are important on a visual or structural level, while removing those that are redundant or too far away from the scene's view center. An avenue to explore could also be to introduce an elimination pool, like in He et al.'s GVGEN [HCP+24]. This could stabilize pruning by reconsidering previously removed splats.

Mip-Splatting's, or more precisely GOF's splat splitting criterion, allows for refining blurry regions. It represents a general extension that showed great promise in this study. MCMC and EAGLES could be extended with this criterion, which could unlock even better fidelity and quicker convergence.

Another possibility could be to adapt the optimization heuristic based on dataset-specific characteristics. Geo-Gaussian and Gaussian-Pro assume clear object geometries. This is not applicable in all scenes. Results on the stump dataset indicate that applying these assumptions to more unstructured, organic environments is suboptimal. Future work could enable the constraints only on scenes that actually contain structured geometric details.

EAGLES showed great promise in the subjective evaluation. When analyzing its training behavior, it becomes clear that the method creates significatly smaller models as is enforced by the splat limit. It does this without losing performance. This indicates that the method could benefit from a larger pool of splats, which could be pruned until the desired model size is reached. This would have to be done in a deterministic way such that the desired splat budget is actually reached and no geometric detail is lost. Maybe a merging approach similar to Kerbl et al.'s hierarchical LOD [KMK+24] technique would be a good fit for this task.

The evaluation approach used in this work could be extended and improved to address its current limitations. Expanding the dataset variety, incorporating expert reviewers, and controlled lab conditions could enhance the reliability of the subjective rankings. In order to create better alignment between the cameras in the data and the views experienced by participants, users could be shown pre-defined camera paths, followed by the ability to take control and explore areas of interest.

New studies would benefit from cross-circumstance pairwise comparisons. Active sampling techniques like ASAP could be used to maintain a manageable study size. Implementing this effectively in a web-based setting was sadly not possible with the resources of this study.

For me personally, this work represents a great learning experience. Experimenting with different heuristic approaches provided deeper insights into how model parameters evolve under different constraints. By actually seeing the techniques' mechanics translate into concrete 3D structures, I gained a better understanding of machine learning more broadly. It's very satisfying to see the abstract concepts described in the literature play out in front of one's eyes.

APPENDIX A

Survey Demographic

A total of 64 people were part of the evaluation. The demographic form was placed at the end of the evaluation, and only 33 participants made valid entries. This is not an unintended consequence, as it was left to the user's discretion whether they felt comfortable sharing their personal information. The following paragraphs give a brief overview of the gathered data. It is important to keep in mind that only a subset of the entire population is represented.

The age range of the participants was between 12 and 61, with a median age of 27. An overall distribution of the age can be seen in figure A.1. Most participants were male, with only 24.2% female. Users are largely from STEM disciplines. The majority are Data

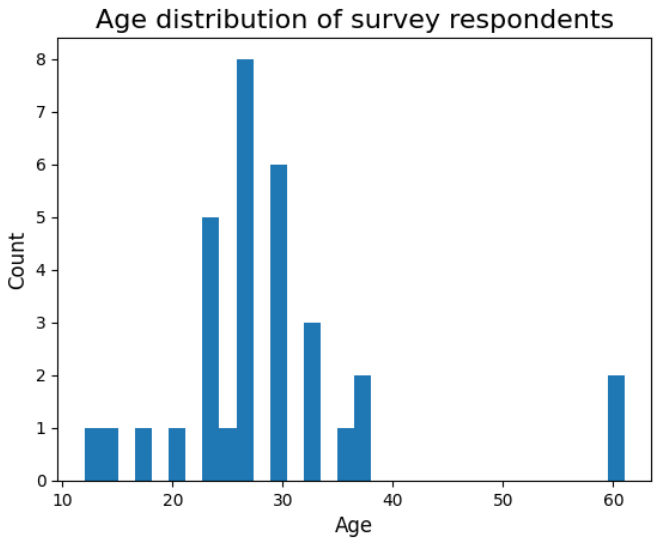

Figure A.1: Age range of the survey participants

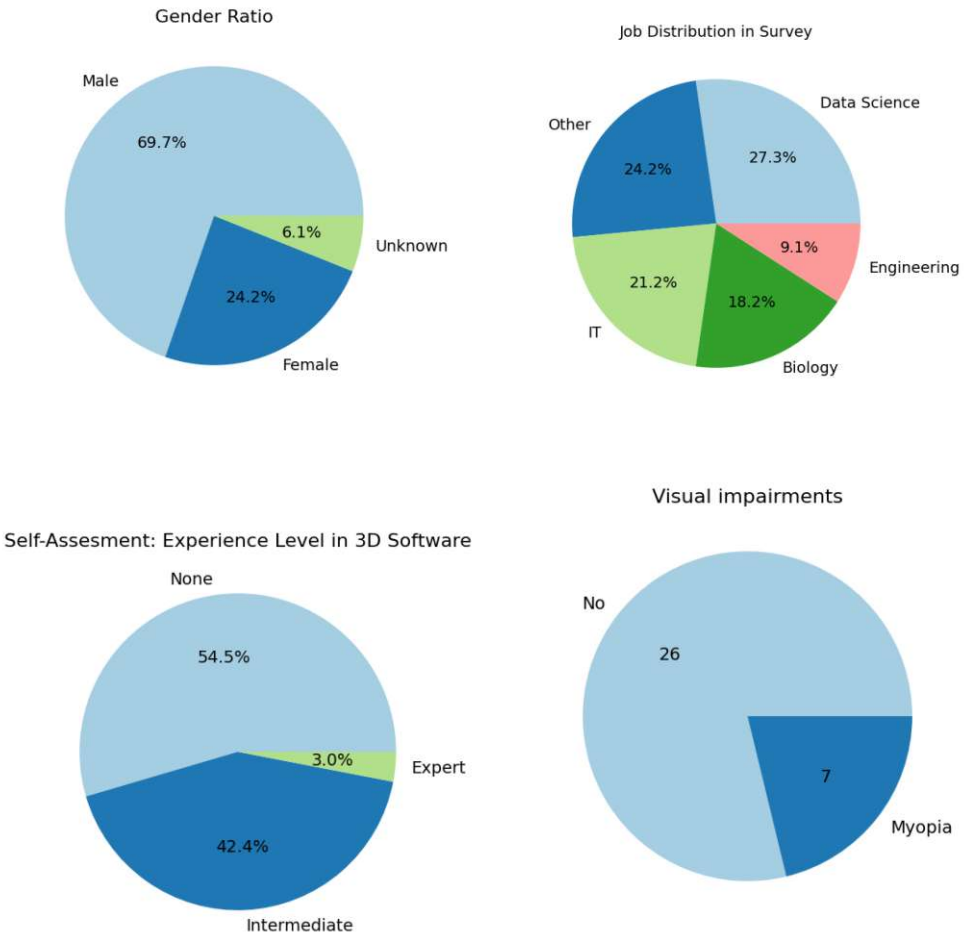

Figure A.2: Ratios of survey demographic

Scientists, Engineers, Biologists, and other IT-related professionals. 24.2% are students and or belong to other uncategorized occupations. In the self-assessed experience level with 3D software, the vast majority of people reported that they had intermediate or no experience. Only three percent classified themselves as experts. Seven participants are myopic, and none are colorblind. For a visualization of the demographics, view figure A.2.

The demographics are clearly skewed towards males, who are in some sort of STEM discipline. Most participants aren't 3D experts, which means that they are likely new to the topic of IQA for 3D models. There is a low level of visual impairment in the study group.

APPENDIX B

Methodological Details

Algorithm B.1: The objective evaluation's algorithm to compute the metrics for every model, scene and size

Data: A set of datasets: {"truck", "room", "stump"}
A set of methods: {"default", "eagles", "gaussian-pro", "geo-gaussian", "mcmc", "mini-splatting", "mip-splatting"}
A set of sizes: {"low", "medium", "high"}
Number of perturbations: N_p
Result: A structured table of computed metrics

```
Disable gradient computation
Initialize an empty list records
foreach dataset in available datasets do
    Load test split from dataset
    Create N_p new camera poses from existing cameras
    foreach method in available methods do
        foreach size in available sizes do
            Load model corresponding to (method, size, dataset)
            foreach view index in test split do
                Retrieve camera pose, ground-truth image, alpha mask, and
                 intrinsic matrix
                Compute background image
                Blend ground-truth image with background
                Render output image using the model
                Compute PSNR, SSIM, and LPIPS metrics
                Compute BRISQUE score
                Initialize a record storing the computed metrics
                foreach perturbed index in N_p do
                    Retrieve perturbed camera pose
                    Render output image with perturbed camera pose
                    Compute BRISQUE score for perturbed image
                    Store BRISQUE score in the record
                end
                Append record to records
            end
        end
    end
end
return records
```

APPENDIX C

Additional Subjective Results

size	scene	Default	MCMC	Mini-Splatting	EAGLES	Mip-Splatting	Gaussian-Pro	Geo-Gaussian
room	low	1.225	0.043	0.009	-0.936	0.504	-0.176	-0.669
room	medium	-2.427	-4.120	-5.309	5.985	4.415	4.969	-3.514
room	high	-0.216	-1.400	0.445	1.378	0.891	0.776	-1.874
stump	low	0.085	1.091	0.589	-1.131	0.847	0.148	-1.628
stump	medium	-5.231	10.146	11.993	-6.914	2.800	-6.086	-6.709
stump	high	-0.113	1.347	3.049	1.453	2.254	1.005	-8.995
truck	low	1.508	3.304	-0.125	-0.201	-7.268	0.304	2.477
truck	medium	-1.900	6.334	-2.351	-1.588	5.380	-3.194	-2.682
truck	high	-0.600	-0.584	-0.320	1.905	0.528	1.033	-1.962

Table C.1: Estimated JOD scores for each model, scene and size

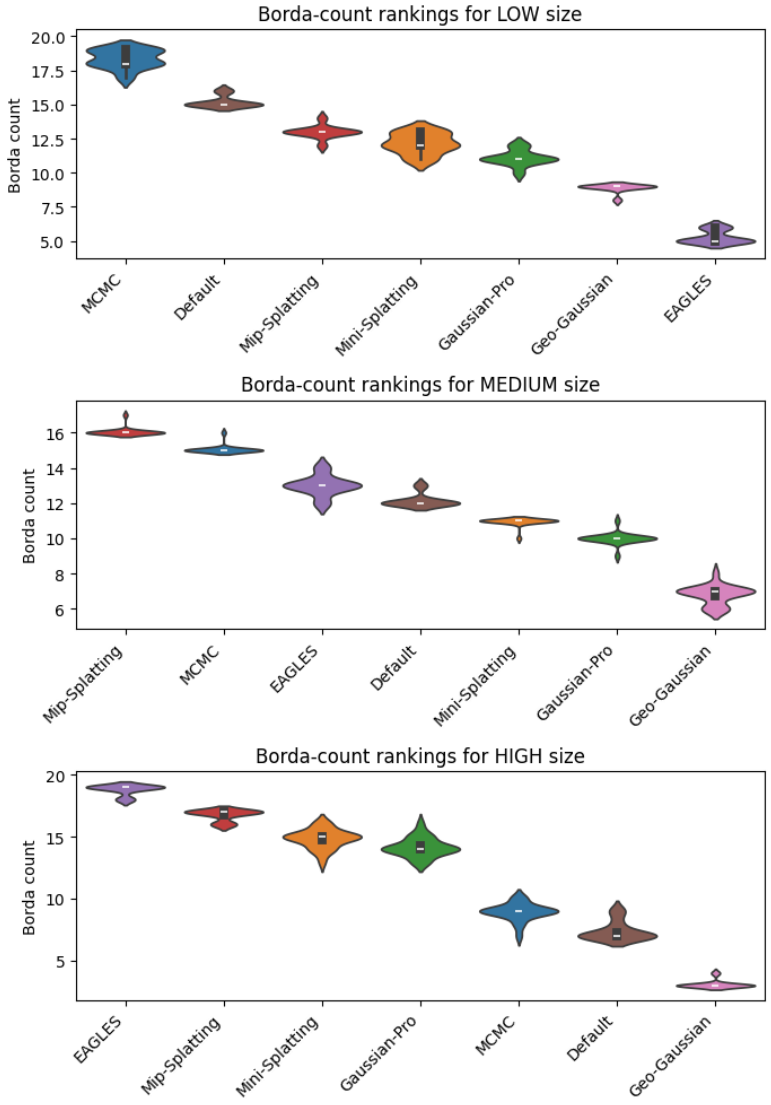

Figure C.1: Subjective rankings by size aggregated by Borda count using 20-Fold Cross-Validation, higher is better

Technique	Count
Mip-Splatting	46
MCMC	43
Mini-Splatting	38
EAGLES	37
Gaussian-Pro	35
Default	34
Geo-Gaussian	19

Table C.2: Borda-count aggregation of JOD values as global ranking

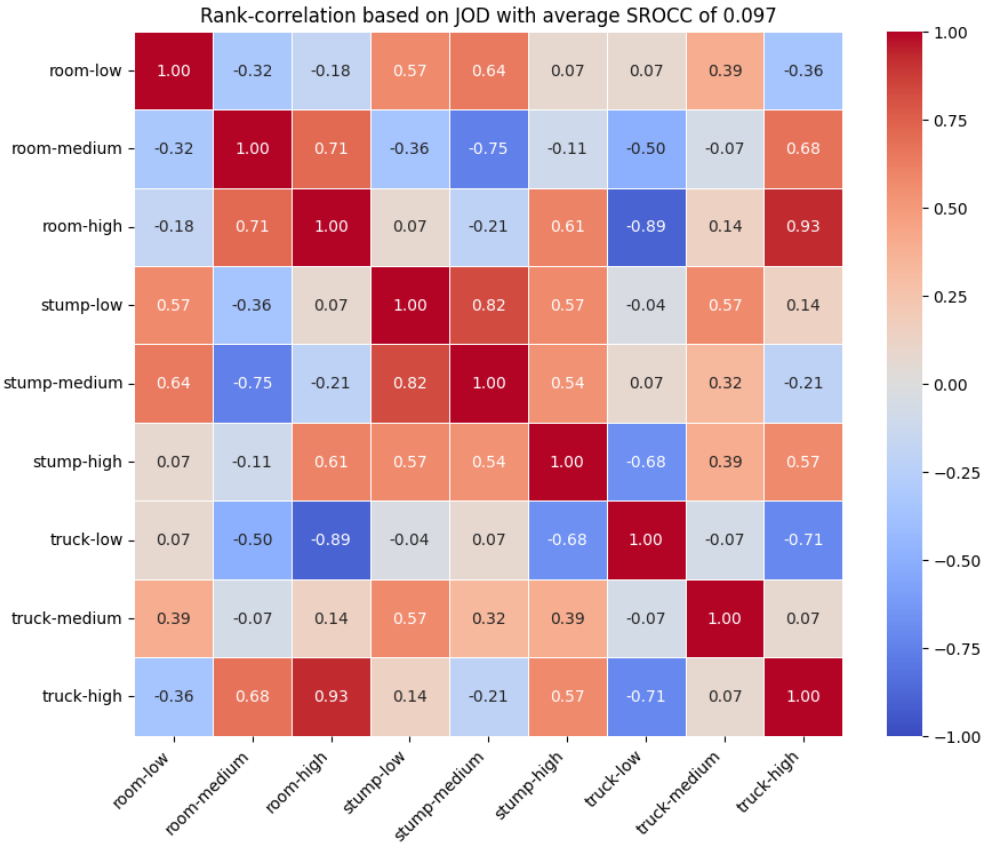

Figure C.2: Rank-correlation based on JOD

APPENDIX D

Additional Objective Results

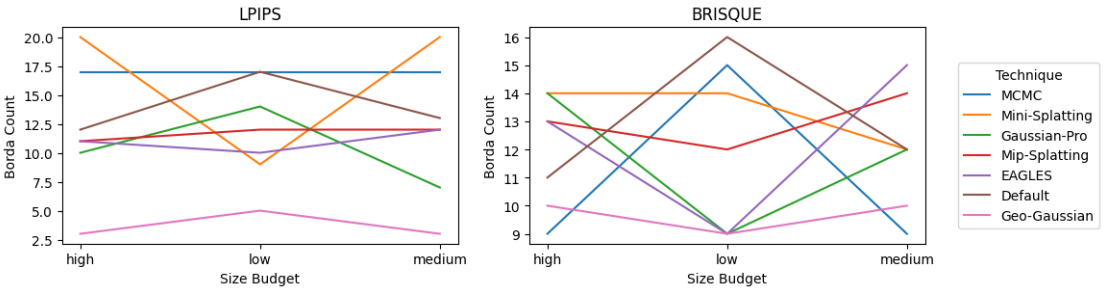

Figure D.1: Development of objective rankings aggregated by Borda count across the size budgets, higher is better

scene	size	Default	EAGLES	Gaussian-Pro	Geo-Gaussian	MCMC	Mini-Splatting	Mip-Splatting
room	low	0.889	0.887	0.889	0.856	0.883	0.886	0.891
room	medium	0.906	0.899	0.902	0.883	0.906	0.904	0.905
room	high	0.913	0.909	0.907	0.887	0.915	0.915	0.911
stump	low	0.588	0.569	0.578	0.542	0.633	0.594	0.584
stump	medium	0.653	0.639	0.632	0.616	0.700	0.702	0.652
stump	high	0.687	0.679	0.680	0.654	0.732	0.747	0.697
truck	low	0.791	0.779	0.781	0.776	0.795	0.761	0.775
truck	medium	0.832	0.830	0.826	0.814	0.833	0.829	0.828
truck	high	0.848	0.849	0.846	0.832	0.850	0.852	0.845

Table D.1: SSIM for each model, scene and size

scene	size	Default	EAGLES	Gaussian-Pro	Geo-Gaussian	MCMC	Mini-Splatting	Mip-Splatting
room	low	28.874	28.595	29.110	27.529	28.977	29.252	29.058
room	medium	29.760	29.846	30.191	28.740	30.073	29.977	30.176
room	high	30.324	30.298	30.250	28.937	30.447	30.357	30.111
stump	low	22.555	22.076	22.703	21.533	23.723	22.041	22.621
stump	medium	23.851	23.891	23.691	22.360	24.677	24.909	24.098
stump	high	24.054	24.579	24.504	22.792	25.138	25.817	24.778
truck	low	22.049	21.477	21.645	21.732	22.703	20.621	21.210
truck	medium	23.120	23.397	23.186	22.801	23.750	23.066	23.148
truck	high	23.628	23.917	23.776	23.321	24.137	23.875	23.669

Table D.2: PSNR for each model, scene and size

scene	size	Default	EAGLES	Gaussian-Pro	Geo-Gaussian	MCMC	Mini-Splatting	Mip-Splatting
room	low	0.140	0.144	0.143	0.161	0.149	0.153	0.143
room	medium	0.115	0.116	0.117	0.130	0.110	0.113	0.115
room	high	0.101	0.104	0.108	0.122	0.095	0.099	0.101
stump	low	0.411	0.430	0.420	0.449	0.347	0.357	0.397
stump	medium	0.303	0.322	0.330	0.338	0.243	0.219	0.293
stump	high	0.238	0.256	0.252	0.282	0.189	0.163	0.222
truck	low	0.203	0.213	0.209	0.217	0.201	0.235	0.221
truck	medium	0.146	0.145	0.147	0.160	0.147	0.138	0.151
truck	high	0.121	0.116	0.118	0.131	0.120	0.106	0.123

Table D.3: LPIPS for each model, scene and size

scene	size	Default	EAGLES	Gaussian-Pro	Geo-Gaussian	MCMC	Mini-Splatting	Mip-Splatting
room	low	116.446	116.466	116.480	116.422	116.465	116.404	116.489
room	medium	116.557	116.536	116.537	116.501	116.596	116.561	116.589
room	high	116.610	116.596	116.564	116.530	116.626	116.622	116.663
stump	low	116.045	116.236	116.247	116.748	115.219	115.979	116.114
stump	medium	114.995	115.338	115.430	115.716	113.502	113.513	114.903
stump	high	113.610	114.046	113.894	114.849	112.576	111.571	113.374
truck	low	116.348	116.387	116.349	116.400	116.374	116.445	116.306
truck	medium	116.030	115.983	116.027	116.083	116.127	116.054	115.897
truck	high	115.809	115.706	115.706	115.869	115.963	115.707	115.642

Table D.4: BRISQUE for each model, scene and size

Technique	SSIM	PSNR	LPIPS	BRISQUE
MCMC	54	56	51	33
Default	47	30	42	39
Mini-Splatting	45	41	49	40
Mip-Splatting	36	37	35	39
EAGLES	30	35	33	37
Gaussian-Pro	29	40	31	35
Geo-Gaussian	11	13	11	29

Table D.5: Borda-count aggregation for objective metrics

APPENDIX E

Additional Joint Analysis

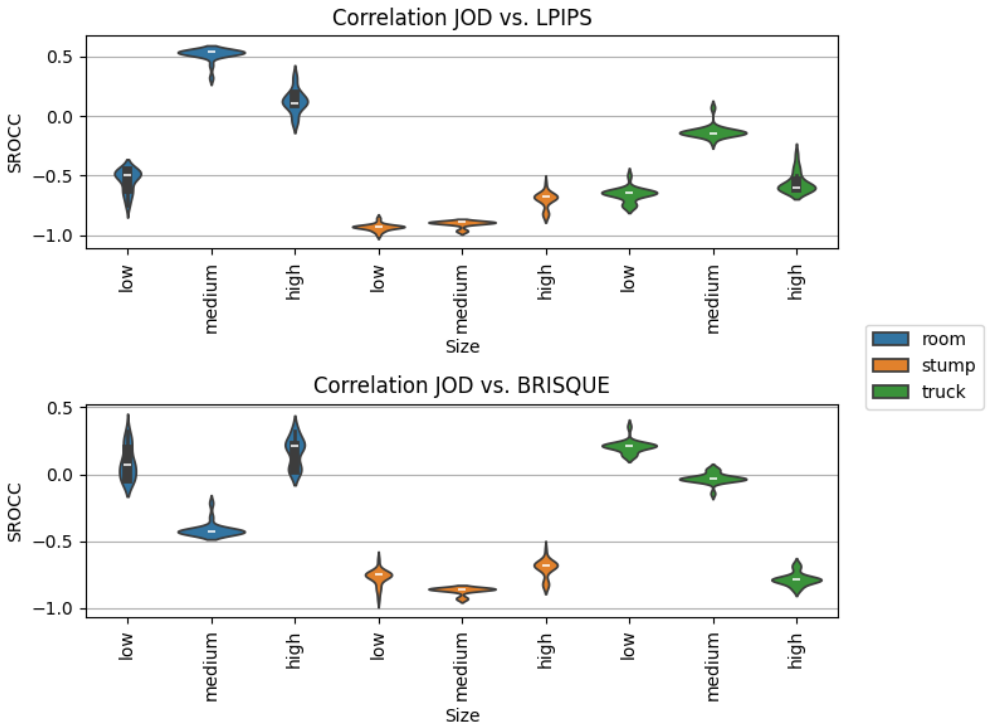

Figure E.1: Correlation between objective measures and JOD based on 20-Fold Cross-Validation, lower is better

APPENDIX F

Qualitative Examples

Figure F.1: Mip-Splatting's relative visual performance for the room scene and the low size budget

Figure F.2: Mip-Splatting's relative visual performance for the stump scene and the medium size budget

Figure F.3: Mip-Splatting's relative visual performance for the truck and the low size budget

Figure F.4: Mip-Splatting's relative visual performance for the truck and the low size budget

Figure F.5: MCMC's relative visual performance for the truck and the medium size budget

Figure F.6: MCMC's relative visual performance for the truck and the medium size budget

Figure F.7: MCMC's relative visual performance for the stump scene and the low size budget

Figure F.8: MCMC's relative visual performance for the room scene and the medium size budget

Figure F.9: MCMC's relative visual performance for the truck scene and the high size budget

Figure F.10: EAGLES's relative visual performance for the room scene and the medium size budget

Figure F.11: EAGLES's relative visual performance for the stump scene and the medium size budget

Figure F.12: EAGLES's relative visual performance for the room scene and the low size budget

Figure F.13: Mini-Splatting's relative visual performance for the stump scene and the high size budget

Figure F.14: Mini-Splatting's relative visual performance for the room scene and the medium size budget

Figure F.15: Geo-Gaussian's relative visual performance for the stump scene and the high size budget

Figure F.16: Geo-Gaussian's relative visual performance for the truck scene and the high size budget

Figure F.17: Geo-Gaussian's relative visual performance for the stump scene and the low size budget

Figure F.18: Geo-Gaussian's relative visual performance for the truck scene and the low size budget

APPENDIX G

Additional Training Statistics

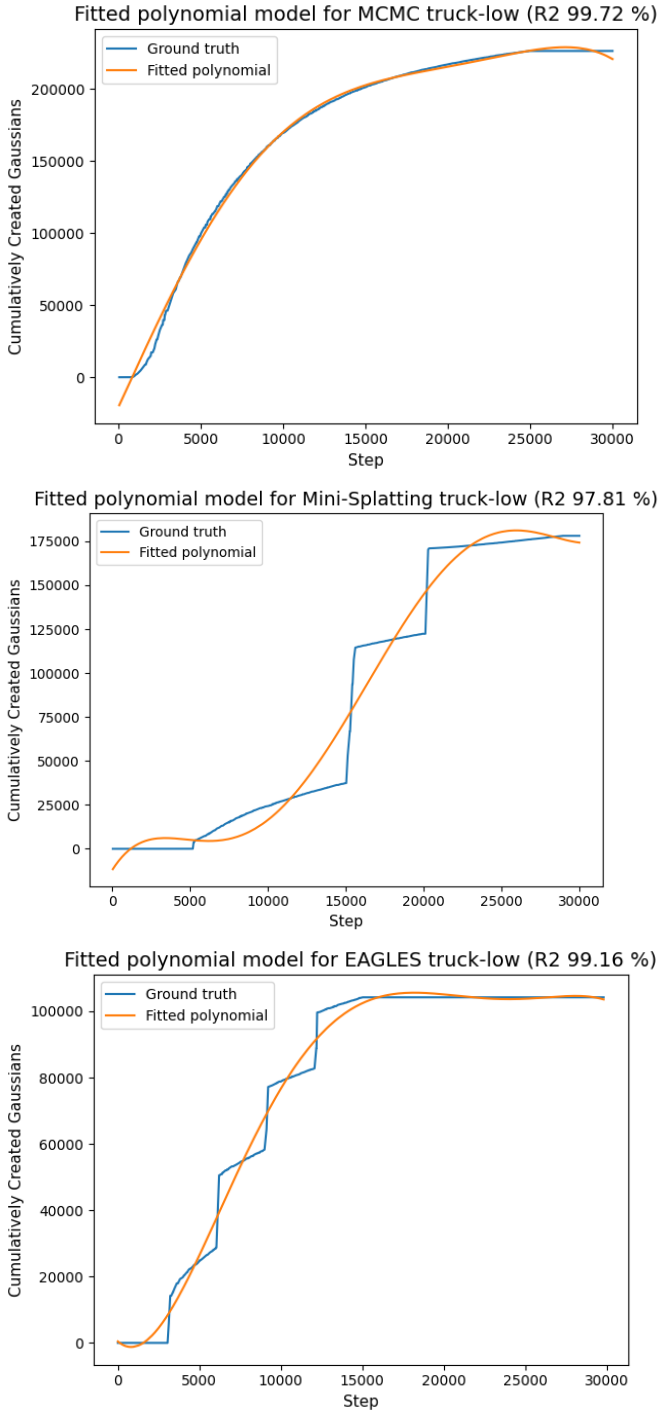

Figure G.1: Examples of the trained 5th-degree polynomial models to meansure the smoothness of curves

Name	Description
Poly-Model R2 Test SSIM	A measure of the predictability of the Test SSIM. A 5th-degree polynomial model is fitted to the SSIM curve. The R^2 represents how much of the variation in the SSIM is explained by the smooth curve (see also G.1)
Entropy of Diff. Created	A measure of the uncertainty of splat creations measured by the entropy of differences [Nar14] in the created splat time series. It represents the complexity of the time series and how predictable its changes are.
Total Creations	The absolute number of splat creations that occurred throughout the training process.
Poly-Model RMSE Created	Similar to the R^2 measure regarding the SSIM. The RMSE is used instead, in order to measure the magnitude of error, rather than just the approximation by the smooth curve.
Test PSNR 2nd Derivative	A measure of smoothness of the test set PSNR curve. The second derivative is approximated for every point in the time series using the formula $x''_t = x_{t-1} - 2x_t + x_{t+1}$. This value is averaged across the timeseries. Lower values indicate a smoother curve.
AUC Num. Gaussians	A measure of the usage of the splat budget. The area under the number of splats curve is estimated and normalized by the total possible area given the splat budget.
Entropy of Diff. PSNR	A measure of the uncertainty of the test PSNR. Similar to the entropy of differences for the created curve.

Table G.1: Curve statistics used in the overall behavior summary

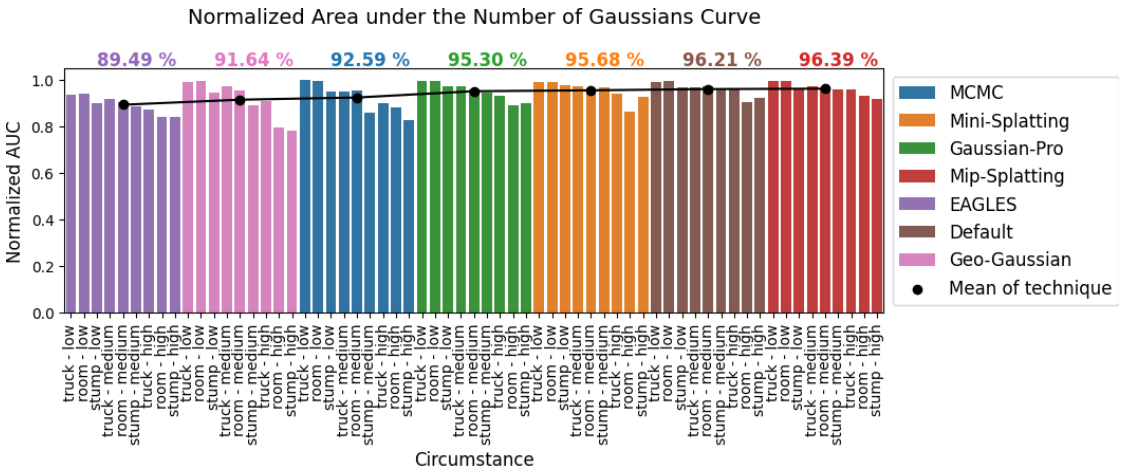

Figure G.2: Area under the number of Gaussians curve

APPENDIX H

COLMAP Scans

Figure H.1: Camera positions of the stump scene's COLMAP scan

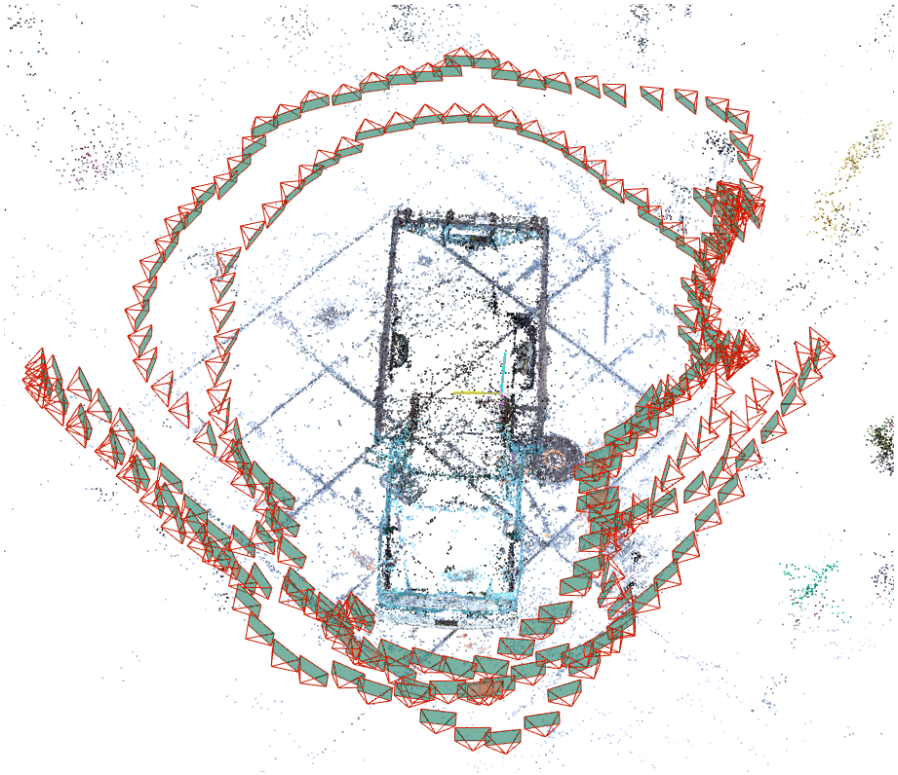

Figure H.2: Camera positions of the truck scene's COLMAP scan

APPENDIX I

Code

This chapter outlines the code that was produced to achieve the goals of this study. It includes the adjusted optimization procedures, the evaluation tool, the data analysis, the training analytics and the demographic analysis.

Optimization Procedures

The implementations of Default and MCMC are based on the gsplat framework. They are implemented in the same repository, which can be found under the following link: `https://github.com/PaulErpen/gsplat-trainer`. The other methods all have their own repository, which were forked from the original code release:

- Mip-Splatting: `https://github.com/PaulErpen/mip-splatting-capped`
- Mini-Splatting: `https://github.com/PaulErpen/mini-splatting-capped`
- Geo-Gaussian: `https://github.com/PaulErpen/GeoGaussian-capped`
- Gaussian-Pro: `https://github.com/PaulErpen/GaussianPro-capped`
- EAGLES: `https://github.com/PaulErpen/efficientgaussian-capped`

Evaluation Related The evaluation tool is comprised of the frontend-application, the cloud functions and database management scripts. The source code can be found here: `https://github.com/PaulErpen/eval-viewer`.

The training analytics code pulls the statistics from the logging interface that were created during model optimization and visualizes them. It can be found under the following

link: `https://github.com/PaulErpen/wandb-data-analysis`. Its results are displayed in section 4.4 and appendix G.

The demographic analysis is laid out in appendix A. Its visualizations and data are sourced from the following repository: `https://github.com/PaulErpen/user-data-analysis`.

The figures and tables presented in sections 4.2, 4.1 and 4.3 as well as appendices D, C and E are created using the main data analysis codebase. It can be found under the following link: `https://github.com/PaulErpen/final-data-analysis`.

Overview of Generative AI Tools Used

The spellchecking and grammar correction for this work was done with the help of ChatGPT. OpenAI does not directly disclose which model is being used, but it is either GPT-4o or GPT-4o Mini. The entire spellcheck and grammar correction was conducted on the 23rd of March, 2025. Every single paragraph of this work was passed to the chatbot with the following prompt pasted in front of the respective paragraph:

Correct spelling and grammar mistakes in the following excerpt from my thesis. This is a LaTeX document, so treat it as a piece of code. Leave the phrasing and word choice as is, if possible. Do not change any commands or indents.

List of Figures

List of Tables

List of Algorithms

Acronyms

3DGS MCMC 3D Gaussian Splatting as Markov Chain Monte Carlo. 20

ACR Absolute Category Rating. 23, 24

ASAP Active Sampling for Pairwise Comparisons. 30, 42

BRISQUE Blind/Referenceless Image Spatial Quality Evaluator. 22, 29, 39, 56, 59, 65, 72, 80, 95

CNN Convolutional Neural Network. 8, 22

DMOS Differential/Degraded Mean Opinion Score. 24

DSCQS Double Stimulus Quality Scale. 23, 24, 26

DSIS Double Stimulus Impairment Scale. 23–26, 30

EWA Elliptical Weighted Average. 12, 91

GOF Gaussian Opacity Fields. 17, 19

GS Gaussian Splatting. 1–3, 8, 10–21, 25–34, 38, 41, 43, 45, 49, 91

IBR Image-based Rendering. 5, 6, 9

IQA Image Quality Assessment. 3, 20–22, 25, 65

ISI Inter-Stimulus Interval. 23

ITU International Telecommunication Union. 22

JOD just-objectionable-difference. 25, 29, 30, 59, 60, 62–65, 75, 81–83, 92, 95

LDI Layered Depth Images. 6

LOD Level of Detail. 16, 17

LPIPS Learned Perceptual Image Patch Similarity. 22, 29, 45, 56, 59, 65, 72, 79, 95

MLP Multilayer Perceptron. 10, 91

MOS Mean Opinion Score. 24

MSCN Mean-subtraction and Contrast Normalization. 22

MSE Mean Squared Error. 21

NVS Novel View Synthesis. xiii, 1, 3, 5, 8, 9, 11, 14, 16, 20–22, 25, 26, 28, 30, 36

PCA Principal Component Analysis. 53

PLT Page Load Time. 34

PSNR Peak Signal-to-Noise Ratio. 21, 22, 29, 45, 47–49, 54, 56, 59, 64, 65, 72, 76, 78, 91, 92, 95

PWC Pairwise Comparison. 23, 25, 26, 30, 41

RMSE Root Mean Square Error. 76

RTR Austrian Regulatory Authority for Broadcasting and Telecommunications. 33, 34, 95

SDF Signed Distance Function. 19

SGD Stochastic Gradient Descent. 20

SGLD Stochastic Gradient Langevin Dynamics. 20

SIFT Scale Invariant Feature Transform. 6, 8

SMPL Skinned Multi-Person Linear Model. 17

SROCC Spearman's Rank Order Correlation Coefficient. 56, 61, 64, 65

SSIM Structural Similarity Index. 15, 21, 22, 29, 45, 47–50, 54, 56, 59, 64, 65, 72, 76, 77, 91, 92, 95

SVR Support Vector Regressor. 22

Bibliography

[BL05] M. Brown and D.G. Lowe. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 56–63, 2005. Conference Name: Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05) ISBN: 9780769523279 Place: Ottawa, ON, Canada Publisher: IEEE.

[BMV+22] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields, March 2022. arXiv:2111.12077 [cs].

[BPK24] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising Densification in Gaussian Splatting, April 2024. arXiv:2404.06109 [cs.CV].

[Bul14] David R. Bull. Chapter 10 - Measuring and Managing Picture Quality. In David R. Bull, editor, *Communicating Pictures*, pages 317–360. Academic Press, Oxford, January 2014.

[CLI+20] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction, August 2020. arXiv:2003.10983 [cs].

[CLY+24] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. GaussianPro: 3D Gaussian Splatting with Progressive Propagation, February 2024. arXiv:2402.14650.

[CW24] Guikun Chen and Wenguan Wang. A Survey on 3D Gaussian Splatting, July 2024. arXiv:2401.03890 [cs].

[CWL+24] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression, March 2024. arXiv:2403.14530 [cs.CV].

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. arXiv:2010.11929 [cs].

[Deu96] P. Deutsch. *RFC1951: DEFLATE Compressed Data Format Specification version 1.3*. RFC Editor, USA, April 1996. https://doi.org/10.17487/RFC1951.

[DHR+24] Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lučić, Richard Szeliski, and Jonathan T. Barron. SMERF: Streamable Memory Efficient Radiance Fields for Real-Time Large-Scene Exploration, July 2024. arXiv:2312.07541 [cs].

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description, 2018. arXiv:1712.07629 [cs.CV].

[Fau94] Olivier Faugeras. 3-D scene representation as a collection of images. *Proceedings of 12th International Conference on Pattern Recognition*, 1994. https://doi.org/10.1109/ICPR.1994.576404.

[FFS+24] Yutao Feng, Xiang Feng, Yintong Shang, Ying Jiang, Chang Yu, Zeshun Zong, Tianjia Shao, Hongzhi Wu, Kun Zhou, Chenfanfu Jiang, and Yin Yang. Gaussian Splashing: Unified Particles for Versatile Motion Synthesis and Rendering, July 2024. arXiv:2401.15318 [cs].

[FLK+23] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. COLMAP-Free 3D Gaussian Splatting, December 2023. arXiv:2312.07504 [cs.CV].

[FW24] Guangchi Fang and Bing Wang. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians, October 2024. arXiv:2403.14166 [cs].

[GD98] J. P. Grossman and William J. Dally. Point Sample Rendering. pages 181–192, Vienna, 1998. Springer Vienna. Book Title: Rendering Techniques '98 Series Title: Eurographics, https://www.doi.org/10.1007/978-3-7091-6453-2_17.

[GGS23] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. EAGLES: Efficient Accelerated 3D Gaussians with Lightweight EncodingS, December 2023. arXiv:2312.04564 [cs.CV].

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, August 1996. Association for Computing Machinery.

[GL24] Antoine Guédon and Vincent Lepetit. Gaussian Frosting: Editable Complex Radiance Fields with Real-Time Rendering, March 2024. arXiv:2403.14554 [cs.CV].

[HBZN24] Lukas Höllein, Aljaž Božič, Michael Zollhöfer, and Matthias Nießner. 3DGS-LM: Faster Gaussian-Splatting Optimization with Levenberg-Marquardt, September 2024. arXiv:2409.12892 [cs.CV].

[HCP+24] Xianglong He, Junyi Chen, Sida Peng, Di Huang, Yangguang Li, Xiaoshui Huang, Chun Yuan, Wanli Ouyang, and Tong He. GVGEN: Text-to-3D Generation with Volumetric Representation, July 2024. arXiv:2403.12957 [cs].

[HSM+21] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis, 2021.

[IR12] ITU-R. Methodology for the subjective assessment of the quality of television pictures. Technical report, ITU-R Rec., January 2012. https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-11-200206-S!!PDF-E.pdf.

[IT08] ITU-T. Subjective video quality assessment methods for multimedia applications. Technical report, ITU-T Rec., April 2008. https://www.itu.int/rec/T-REC-P.910-202207-S/en.

[IT13] ITU-T. Mean opinion score interpretation and reporting. Technical report, ITU-T Rec., May 2013. https://www.itu.int/rec/T-REC-P.800.2-201607-I/en.

[JSM+20] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local Implicit Grid Representations for 3D Scenes, March 2020. arXiv:2003.08981 [cs].

[Kan98] Sing Bing Kang. Survey of image-based rendering techniques. In Sabry F. El-Hakim and Armin Gruen, editors, *Videometrics VI*, volume 3641, pages 2 – 16. International Society for Optics and Photonics, SPIE, 1998.

[KCG+24] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. HUGS: Human Gaussian Splats, 2024. arXiv:2311.17910 [cs.CV].

[Kel25] Mark Kellogg. mkkellogg/GaussianSplats3D. https://github.com/mkkellogg/GaussianSplats3D, March 2025. original-date: 2023-09-26T01:15:29Z.

[KKLD23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering, August 2023. arXiv:2308.04079 [cs].

[KMK+24] Bernhard Kerbl, Andréas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets, June 2024. arXiv:2406.12080 [cs].

[KRBD17] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R. Das. Improving user perceived page load time using gaze. In *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*, NSDI'17, pages 545–559, USA, March 2017. USENIX Association.

[KRS+24] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Jeff Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian Splatting as Markov Chain Monte Carlo, June 2024. arXiv:2404.09591 [cs].

[Lev44] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[LGL+21] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields, January 2021. arXiv:2007.11571 [cs].

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, August 1996. Association for Computing Machinery.

[LLD+24] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. GeoGaussian: Geometry-aware Gaussian Splatting for Scene Rendering, March 2024. arXiv:2403.11324 [cs.CV].

[LLF+25] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. CityGaussian: Real-Time High-Quality Large-Scale Scene Rendering with Gaussians. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 265–282, Cham, 2025. Springer Nature Switzerland.

[LM91] Michael S. Landy and J. Anthony Movshon. *Computational Models of Visual Processing*. MIT Press, 1991. https://doi.org/10.7551/mitpress/2002.001.0001.

[LMR+15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, October 2015.

[LMTL21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields, 2021. arXiv:2104.06405 [cs.CV].

[Low04] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[LRS+24] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D Gaussian Representation for Radiance Field, February 2024. arXiv:2311.13681 [cs].

[LWH+24] H. Liang, T. Wu, P. Hanji, F. Banterle, H. Gao, R. Mantiuk, and C. Öztireli. Perceptual Quality Assessment of NeRF and Neural View Synthesis Methods for Front-Facing Views. *Computer Graphics Forum*, 43(2):e15036, 2024. https://doi.org/10.1111/cgf.15036.

[LYX+23] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering, November 2023. arXiv:2312.00109 [cs.CV].

[LZH+24] Zhihao Liang, Qi Zhang, Wenbo Hu, Ying Feng, Lei Zhu, and Kui Jia. Analytic-Splatting: Anti-Aliased 3D Gaussian Splatting via Analytic Integration, March 2024. https://doi.org/10.1007/978-3-031-72643-9_17.

[MESK22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4):1–15, July 2022. arXiv:2201.05989 [cs].

[MF21] Kede Ma and Yuming Fang. Image Quality Assessment in the Modern Age, October 2021. arXiv:2110.09699 [cs, eess].

[MGK+24] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando De La Torre. Taming 3DGS: High-Quality Radiance Fields with Limited Resources, June 2024. arXiv:2406.15643 [cs].

[MMB12] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, December 2012.

[MRAQ24] Pedro Martin, Antonio Rodrigues, Joao Ascenso, and Maria Paula Queluz. NeRF View Synthesis: Subjective Quality Assessment and Objective Metrics Evaluation, May 2024. arXiv:2405.20078 [cs].

[MST+20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, August 2020. arXiv:2003.08934 [cs].

[MWPO+21] Aliaksei Mikhailiuk, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, and Rafał K. Mantiuk. Active Sampling for Pairwise Comparisons via Approximate Message Passing and Information Gain Maximization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2559–2566, January 2021. ISSN: 1051-4651.

[Nar14] Pasquale Nardone. Entropy of Difference, November 2014. arXiv:1411.0506 [physics].

[NMR+24] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. RadSplat: Radiance Field-Informed Gaussian Splatting for Robust Real-Time Rendering with 900+ FPS, March 2024. arXiv:2403.13806 [cs].

[NNS72] M. E. Newell, R. G. Newell, and T. L. Sancha. A solution to the hidden surface problem. In *Proceedings of the ACM annual conference - Volume 1*, volume 1 of *ACM '72*, pages 443–450, New York, NY, USA, August 1972. Association for Computing Machinery.

[NSW24] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis, January 2024. arXiv:2401.02436 [cs].

[POM17] Maria Perez-Ortiz and Rafal K. Mantiuk. A practical guide and software for analysing pairwise comparison experiments, December 2017. arXiv:1712.03686 [stat].

[POMZ+20] María Pérez-Ortiz, Aliaksei Mikhailiuk, Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, and Rafał K. Mantiuk. From Pairwise Comparisons and Rating to a Unified Quality Scale. *IEEE Transactions on Image Processing*, 29:1139–1151, 2020. Conference Name: IEEE Transactions on Image Processing.

[QLC+24] Qiang Qu, Hanxue Liang, Xiaoming Chen, Yuk Ying Chung, and Yiran Shen. NeRF-NQA: No-Reference Quality Assessment for Scenes Generated by NeRF and Neural View Synthesis Methods. *IEEE Transactions on Visualization and Computer Graphics*, 30(5):2129–2139, May 2024. arXiv:2412.08029 [cs].

[RSV+23] Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P. Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T. Barron, and Peter Hedman.

MERF: Memory-Efficient Radiance Fields for Real-time View Synthesis in Unbounded Scenes, February 2023. arXiv:2302.12249 [cs].

[Sca24] Scaniverse. Open-sourcing .SPZ: it's .JPG for 3D Gaussian splats. https://scaniverse.com/news/spz-gaussian-splat-open-source-file-format, 2024.

[Sch99] Daniel Scharstein. *View Synthesis Using Stereo Vision*. Springer Science & Business Media, June 1999. https://doi.org/10.1007/3-540-48725-5.

[SF16] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, June 2016. ISSN: 1063-6919.

[SGHS98] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, July 1998. Association for Computing Machinery.

[Sna25] Noah Snavely. Bundler structure from motion toolkit. https://github.com/snavely/bundler_sfm, February 2025. original-date: 2013-03-10.

[Spe04] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[SSS06] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3):835–846, July 2006.

[SSW+21] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching With Transformers, 2021. arXiv:2104.00680 [cs.CV].

[SVDV96] Steven M. Seitz, View Profile, Charles R. Dyer, and View Profile. View morphing. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30, August 1996.

[SZY+24] Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. SA-GS: Scale-Adaptive Gaussian Splatting for Training-Free Anti-Aliasing, March 2024. arXiv:2403.19615 [cs.CV].

[TAA24] Shaira Tabassum and Seyed Ali Amirshahi. Quality of NeRF Changes with the Viewing Path an Observer Takes: A Subjective Quality Assessment of Real-time NeRF Model. In *2024 16th International Conference on Quality of Multimedia Experience (QoMEX)*, pages 88–91, June 2024. ISSN: 2472-7814.

[TFT+20] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B. Goldman, and Michael Zollhöfer. State of the Art on Neural Rendering, April 2020. arXiv:2004.03805 [cs].

[Thu27] L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927. Place: US Publisher: Psychological Review Company.

[TLY+21] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes, January 2021. arXiv:2101.10994 [cs].

[TS17] Akamai Technologies and SOASTA. The state of online retail performance. Technical report, 2017. https://s3.amazonaws.com/sofist-marketing/State+of+Online+Retail+Performance+Spring+2017+-+Akamai+and+SOASTA+2017.pdf.

[TTM+22] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhoefer, and Vladislav Golyanik. Advances in Neural Rendering, March 2022. arXiv:2111.05849 [cs].

[Tur94] Greg Turk. The PLY Polygon File Format. https://web.archive.org/web/20020124203029/http://www.dcs.ed.ac.uk/teaching/cs4/www/graphics/Web/ply.html, 1994.

[VPS+24] Evangelos Ververas, Rolandos Alexandros Potamias, Jifei Song, Jiankang Deng, and Stefanos Zafeiriou. SAGS: Structure-Aware 3D Gaussian Splatting, April 2024. arXiv:2404.19149 [cs].

[WB06a] Zhou Wang and Alan C. Bovik. *Modern Image Quality Assessment*. Synthesis Lectures on Image, Video, and Multimedia Processing. Springer International Publishing, Cham, 2006. https://www.doi.org/10.1007/978-3-031-02238-8.

[WB06b] Zhou Wang and Alan C. Bovik. *Modern Image Quality Assessment*. Synthesis Lectures on Image, Video, and Multimedia Processing. Springer International Publishing, Cham, 2006.

[WB09] Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*,

26(1):98–117, January 2009. Conference Name: IEEE Signal Processing Magazine.

[WBSS04] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. Conference Name: IEEE Transactions on Image Processing.

[WGSJ20] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end View Synthesis from a Single Image, April 2020. arXiv:1912.08804 [cs].

[WJ63] Joe H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, March 1963. https://doi.org/10.1080/01621459.1963.10500845.

[WLW+23] Yushuang Wu, Xiao Li, Jinglu Wang, Xiaoguang Han, Shuguang Cui, and Yan Lu. Efficient View Synthesis with Neural Radiance Distribution Field, August 2023. arXiv:2308.11130 [cs].

[WT11] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, Madison, WI, USA, June 2011. Omnipress.

[WYF+24] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering, 2024. arXiv:2310.08528v3 [cs.CV].

[WYZ+24] 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. arXiv:2403.11134 [cs].

[XLL+24] Haodong Xiang, Xinghui Li, Xiansong Lai, Wanting Zhang, Zhichao Liao, Kai Cheng, and Xueping Liu. GaussianRoom: Improving 3D Gaussian Splatting with SDF Guidance and Monocular Cues for Indoor Scene Reconstruction, May 2024. arXiv:2405.19671 [cs].

[XYY+24] Yuke Xing, Qi Yang, Kaifa Yang, Yilin Xu, and Zhu Li. Explicit-NeRF-QA: A Quality Assessment Database for Explicit NeRF Model Compression, September 2024. arXiv:2407.08165 [cs, eess].

[YCH+23] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian Splatting, November 2023. arXiv:2311.16493.

[YFKT+21] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks, December 2021. arXiv:2112.05131 [cs].

[YLCL24] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-Scale 3D Gaussian Splatting for Anti-Aliased Rendering. 2024. arXiv:2311.17089 [cs.CV].

[YLK+24] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An Open-Source Library for Gaussian Splatting, September 2024. arXiv:2409.06765 [cs].

[YLT+21] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for Real-Time Rendering of Neural Radiance Fields. pages 5752–5761, 2021.

[YLX+24] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. GSDF: 3DGS Meets SDF for Improved Rendering and Reconstruction, March 2024. arXiv:2403.16964 [cs.CV].

[YSG24] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes, September 2024. arXiv:2404.10772 [cs].

[YSW+19] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable Surface Splatting for Point-based Geometry Processing. *ACM Transactions on Graphics*, 38(6):1–14, December 2019. arXiv:1906.04173 [cs].

[YYX+24] Qi Yang, Kaifa Yang, Yuke Xing, Yiling Xu, and Zhu Li. A Benchmark for Gaussian Splatting Compression and Quality Assessment Study, July 2024. arXiv:2407.14197 [cs].

[ZGO+19] Emin Zerman, Pan Gao, Cagri Ozcinar, Aljosa Smolic, Pan Gao, Cagri Ozcinar, and Aljosa Smolic. Subjective and Objective Quality Assessment for Volumetric Video Compression. *Electronic Imaging*, 31:1–7, January 2019. Publisher: Society for Imaging Science and Technology.

[ZIE+18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric, April 2018. arXiv:1801.03924 [cs].

[ZPvBG02] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, July 2002. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[ZZX+24] Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. FreGS: 3D Gaussian Splatting with Progressive Frequency Regularization, April 2024. arXiv:2403.06908 [cs.CV].