Direct Haptic Rendering of 3D Gaussian Splatting Scenes via Ellipsoid Overlap Check

Dominik Pikos and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
dominik.pikos@kit.edu, uwe.hanebeck@kit.edu

***Abstract*—Real-time collision detection and haptic force feedback are essential for dexterous teleoperated manipulation and immersive interaction in complex virtual environments. 3D Gaussian Splatting (3DGS) achieves photorealistic rendering at real-time frame rates from unstructured image captures, enabling immersive applications. However, the use of 3DGS models for haptic interaction remains largely unexplored due to the stringent real-time requirements of haptic rendering. Existing approaches rely on mesh extraction, voxelisation, or signed distance fields, introducing significant preprocessing overhead, loss of visual fidelity, or computational latency incompatible with stable haptic feedback. We address this gap with an end-to-end system that combines 3DGS rendering with collision detection and haptic force feedback. Our approach leverages the native ellipsoidal structure of 3DGS to perform real-time, multi-contact collision detection using an efficient analytical overlap test to receive a set of haptically relevant contact outputs that directly enable force rendering. Validation with real-world datasets on a haptic interface demonstrates low latency, contact accuracy, and haptic fidelity. This enables the first practical system fusing learned Gaussian scene representations with a real-time collision pipeline and consecutive multi-contact force blending, delivering an immersive visuo-haptic experience.**

***Index Terms*—Telepresence, virtual reality, haptics, novel view synthesis, 3D Gaussian Splatting, 3D scene reconstruction, collision detection, haptic rendering**

I. Introduction

Virtual reality (VR) systems are becoming increasingly important for teleoperation tasks, and virtual training scenarios involving complex manipulations. To improve performance and reduce the cognitive load of the user, the VR-System needs to provide an immersive environment and realistic feedback. Such highly immersive telepresence systems are based on realistic multimodal experiences. Apart from visual and acoustic aspects, haptic feedback plays a crucial role, especially in virtual scenarios where manipulation and interactions with the environment take place.

Regarding the visual component, the advent of 3D Gaussian Splatting (3DGS) [1] enables the reconstruction of photorealistic 3D scenes from unstructured 2D images. These scenes can be rendered in real-time, allowing immersive visualisation on VR devices that support free user locomotion. The efficiency and visual quality have also motivated adoption in robotics [2], scene understanding [3], [4], teleoperation [5], surgical training [6], VR-Games, and many other applications that work with reconstruction of real world scenes. However, these photorealistic representations have not been leveraged for haptic interaction—a gap reflecting fundamental technical barriers. Haptic rendering demands efficient collision detection suitable for millions of primitives, high-frequency deterministic update rates, and explicit collision geometry for force computation. Traditional collision detection pipelines rely on explicit geometric representations such as triangle meshes, voxel grids, or signed distance fields (SDF). These approaches struggle to maintain both visual fidelity and stable haptic feedback when interacting with large-scale learned scenes.

Figure 1: Visualising the proposed method for haptic rendering of 3DGS scenes using a haptic interface like HapticGiant [7].

In this work, we address this limitation by enabling direct collision detection and force feedback from large learned 3D scene representations without explicit surface reconstruction. Our key idea is to exploit the native structure of 3DGS models for haptic interaction. As visualised in Fig. 1, each Gaussian can be interpreted as an ellipsoidal primitive, allowing collisions to be formulated as efficient overlap tests between the haptic interaction point and the learned scene representation. This avoids intermediate geometry conversions and preserves consistency between visual rendering and haptic interaction.

The main contributions of this paper are:

This work was supported by the ROBDEKON project (13N16539) of the German Federal Ministry of Research, Technology and Space.

- An efficient ellipsoid overlap test for collision detection in 3DGS scenes, enabling real-time haptic interaction without mesh extraction.
- A multi-contact force blending method that directly operates on 3D Gaussian primitives.
- A unified framework for fusing visual rendering, collision detection, and haptic force rendering into a single learned scene representation.
- Real-time validation on a haptic device using large-scale real-world 3DGS environments.

The paper structure is as follows. After describing related work for our approach in Section II we introduce the three key components of our method: scene representation, collision detection, and haptic rendering in Section III. In Section IV, we propose the architecture to implement our approach in a telepresence setup. After the evaluation in Section V we discuss our study in Section VI and finally conclude our work in Section VII.

II. Related Work

A. 3D Scene Representation & Reconstruction

Photorealistic virtual environments for immersive telepresence require 3D scene representations that support both high visual quality and low-latency rendering. A widely used standard are explicit geometric models such as triangle meshes often obtained via structure-from-motion [8] and multi-view-stereo [9] with subsequent surface reconstruction [10], [11], [12] and texturing. Meshes provide a clear notion of surface geometry and are well supported by real-time graphics and collision pipelines, but they can be brittle in the presence of thin structures, reflections, and incomplete view coverage and often require extensive cleanup, hole filling, and decimation to be usable in interactive systems. Alternative explicit representations, such as point clouds, voxel grids [13], or SDF [14], [15], can improve robustness but typically trade off memory usage, preprocessing effort, visual detail, and photorealistic appearance.

Recent work has shifted towards learned neural scene representations that optimise view synthesis directly from posed images. In particular, neural radiance fields (NeRFs) [16] model scenes as continuous volumetric functions and achieve high-fidelity novel view synthesis, but their rendering relies on expensive ray marching, which makes real-time interaction challenging.

3DGS [1] bridges this gap by representing the scene as large set of anisotropic 3D Gaussians with learned appearance and opacity. By rasterising these ellipsoidal primitives with screen-space splatting, 3DGS achieves high-quality novel view synthesis at real-time frame rates on standard GPUs. Compared to mesh-centric pipelines, 3DGS avoids explicit surface extraction and can preserve realistic view-dependent appearance captured by the training images. Recent 3DGS extensions optimise reconstruction fidelity, computational efficiency, model compression, and deal with special input or output formats. An overview is given in [17]. Furthermore, several recent works have attempted to adapt 3DGS scenes for robotics applications, such as visual SLAM or grasping tasks as summarised in [2] or to enable physical simulations for deformable object dynamics as demonstrated in [18].

Beyond visual rendering, the choice of scene representation strongly impacts collision detection: meshes provide explicit surfaces and are therefore the de-facto standard input for established collision pipelines, whereas volumetric neural representations (e.g., NeRFs) require additional proxies (meshes, voxels, SDFs) to enable fast, deterministic contact queries. Existing methods for detecting collisions between introduced representations are addressed in the next subsection.

B. Collision Detection

Real-time collision detection is critical for haptic rendering, enabling stable force feedback in teleoperated manipulation, immersive virtual reality, and robotic interaction. Collision detection methods depend critically on the scene representation, often requiring conversion into proxy-representations. The Voxmap-Point-Shell (VPS) algorithm and its subsequent refinements, introduced by [19], established a foundational paradigm for haptics applications. VPS discretises the environment into a voxel occupancy map and represents the dynamic object as a point shell, enabling fast spatial queries by checking whether points fall within occupied voxels. While efficient and widely adopted, voxelisation inherently sacrifices geometric detail through discretisation artifacts and requires offline preprocessing.

SDF provide continuous implicit surface representations without discretisation, offering superior geometric fidelity compared to voxel-based approaches. SDF-based collision detection queries the distance function to compute overlap and contact geometry [20]. However, SDF-based collision detection can be computationally expensive due to the continuous distance evaluations, and become memory-intensive with increasing model resolution.

For representations that can be naturally decomposed into ellipsoidal primitives, the analytical overlap test, proposed in [21], provides exact collision detection without iterative optimisation. This method formulates ellipsoid intersection as a polynomial root-counting problem, yielding deterministic solutions with explicit contact outputs. Recent applications confirm continued relevance for collision avoidance tasks [22], [23]. Therefore, recent work has already applied this mathematical test to 3DGS scenes [23]. However, this approach focuses on robot navigation planning.

C. Haptic Rendering

Classical haptic rendering methods focus on transforming geometric contact information into stable force feedback at high servo rates. Most approaches follow either impedance or admittance control and compute interaction forces using penalty-based or constraint-based formulations. God-object- [24] and proxy-methods [25] maintain a surface-constrained object whose offset from the device generates restoring forces. Penalty methods like [19] and [26] compute forces

proportional to penetration depth, trading physical exactness for computational simplicity in complex, multi-contact scenarios. Subsequent approaches commonly rely on explicit geometric proxies such as triangle meshes, voxel maps, or SDF [20] to compute penetration depth and surface normals. While effective, these pipelines require preprocessing and assume discrete surface representations, which are not directly compatible with modern learned or volumetric scene models. Several approaches address primitives that lack explicit geometry. For point clouds, [27] proposed a proxy-based method for real-time haptic rendering of streaming point cloud data, extending virtual-coupling proxies to operate without preprocessing and to handle spatial discontinuities in the point set. More recently, [28] introduced haptic rendering of NeRFs, proposing a stochastic proxy method that handles the inherent noise of volumetric NeRF reconstructions.

In contrast, we perform penalty-based force computation directly on the native Gaussian primitives of 3DGS, interpreting their covariances as volumetric contact elements and thereby eliminating intermediate proxy geometry. This enables deterministic multi-contact force rendering from the learned representation.

III. Method

As depicted in Section II, direct haptic force rendering from 3DGS representations remains unexplored. Our work fills this gap by leveraging Gaussian covariances directly as collision ellipsoids, enabling real-time application with explicit contact outputs suitable for haptic feedback control. Our methodology can be split into three parts: *A.* scene representation, *B.* ellipsoid collision check, and *C.* haptic system.

A. Scene Representation

To develop and present our method, we adopt the vanilla 3D Gaussian Splatting [1] approach as a baseline. However, our method can be readily extended to improved or modified variants, provided they employ the same underlying data structure. Depending on the specific output, fine-tuning of the subsequent steps may be required.

We assume that the 3DGS scene has already been trained and the model provides state of the art visual and geometric quality. Consequently, a list of Gaussians $\mathcal{G} = \{G_i\}_{i=1}^{N}$ is obtained, representing the scene. The set contains the 3D covariance matrix $\mathbf{\Sigma}_i$ and the mean position $\underline{\mu}_i \in \mathbb{R}^3$ of each Gaussian G_i, defining

$$G_i(\underline{x}) = e^{-\frac{1}{2}(\underline{x})^\top \mathbf{\Sigma}_i^{-1}(\underline{x})} \quad . \tag{1}$$

Its covariance matrix $\mathbf{\Sigma}_i$ is represented via an equivalent parametrization using

$$\mathbf{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^\top \mathbf{R}_i^\top \quad , \tag{2}$$

where $\mathbf{R}_i$ is the rotation matrix, and $\mathbf{S}_i$ is the scale matrix of the corresponding ellipsoid. This ellipsoid representation, in combination with each given opacity $\alpha_i \in [0, 1]$, is the explicit formulation on which our method is based. In addition, a colour value dependent on the view direction is defined. This plays a decisive role in rendering, but is not taken into account in the proposed haptic pipeline.

Standard real-world 3DGS datasets typically contain millions of Gaussians representing a scene. To enable real-time haptic interaction, the scene representation must be processed efficiently, which requires strategies to reduce data complexity without compromising the accuracy of the resulting haptic representation. Since our objective is to generate a haptic representation of the scene, we focus exclusively on solid objects. Therefore, we apply an opacity-based filtering strategy to the reconstructed Gaussian scene. This approach offers two main advantages: *(1)* it excludes non-relevant floating Gaussians that do not contribute to the perceived geometry, and *(2)* it prevents false-positive collisions in empty space, where sparse or misaligned Gaussians may have been placed by the reconstruction algorithm. Based on preliminary experiments with real-world 3DGS data, we found that applying an opacity threshold of $\alpha_i > (0.1 \dots 0.3)$ provides a suitable trade-off between scene completeness and data reduction. Using this criterion, typical real-world scenes are reduced to approximately one quarter of their original size without losing the geometric representation of the scene. To perceive highest visual quality, the original model is used for rendering, which can be performed on a dedicated machine.

For the registration of the 3DGS model with the haptic workspace, the filtered Gaussians are transformed by a global similarity transform with uniform scale s, rotation $\mathbf{R}_g$, and translation $\underline{t}$. Each Gaussian is mapped to world coordinates via

$$\begin{aligned} \underline{c_i} &= s\,\mathbf{R}_g\ \mu_i + \underline{t}\ , \\ \mathbf{\Sigma}_i^w &= s^2\,\mathbf{R}_g\,\mathbf{\Sigma}_i\,\mathbf{R}_g^\top\ . \end{aligned} \tag{3}$$

These world-space ellipsoids ($\underline{c_i}$,$\mathbf{\Sigma}_i^w$) form the basis for our subsequent collision detection.

B. Ellipsoid Collision Check

Collision detection traditionally employs a hierarchical cascade: a 'broad-phase' rapidly culls candidate pairs using spatial structures, followed by a 'narrow-phase' that performs exact geometric tests on remaining candidates.

1) Broad Phase: The broad phase yields to maintain a small set of nearby Gaussians for the narrow collision test. In the first stage we store all ellipsoid centers $\underline{c_i}$ in a kdtree. For each tool pose, we query the kdtree with a spherical search radius around the tool center $\underline{c}_{\text{tool}}$. In a second stage, we apply a bounding-sphere pre-filter. For each Gaussian from the kdtree subset, we precompute a radius r_i from its log-space scales, and we define a bounding-sphere radius r_{tool} for the tool ellipsoid. A candidate is only kept if

$$\left\|\underline{c_i} - \underline{c}_{\text{tool}}\right\|^2 < (r_i + r_{\text{tool}})^2 \quad , \tag{4}$$

which can be evaluated in a fully vectorised manner for all kdtree candidates. This step removes the majority of non-intersecting Gaussians at negligible cost.

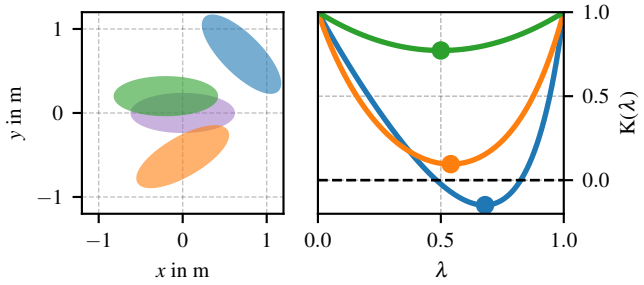

Figure 2: Function $K(\lambda)$ for different positions of ellipses relative to the *tool-ellipse* (purple).

2) Narrow Phase: To facilitate the analytical collision detection with the scene's Gaussian primitives, we model the haptic interaction point as an ellipsoid. In the narrow phase, we perform an exact ellipsoid overlap test for the remaining candidate Gaussians. While the broad phase filters based on bounding spheres, the narrow phase determines if the ellipsoidal volume of the tool actually intersects with the ellipsoidal volume of the 3DGS model.

We utilise the algebraic separating point criterion based on the method described by Gilitschenski et al. [21]. The tool ellipsoid and the Gaussian ellipsoid candidate are each defined by their centers $(\underline{c}_i, \underline{c}_{\text{tool}})$ and shape matrices $(\mathbf{A}_i, \mathbf{B}_{\text{tool}})$. The shape matrices correspond to the inverse of the covariance matrices. The intersection test is formulated as a minimisation problem of a scalar function $K(\lambda)$ over the interval $\lambda \in [0, 1]$. We define a convex combination of the shape matrices $\mathbf{E}(\lambda)$ and a weighted center vector $\underline{w}(\lambda)$:

$$\begin{aligned} \mathbf{E}(\lambda) &= \lambda \mathbf{B}_{\text{tool}} + (1-\lambda)\mathbf{A}_i \ , \\ \underline{w}(\lambda) &= \lambda \mathbf{B}_{\text{tool}} \underline{c}_{\text{tool}} + (1-\lambda)\mathbf{A}_i \underline{c}_i \ , \\ m(\lambda) &= \mathbf{E}(\lambda)^{-1} \underline{w}(\lambda) \ . \end{aligned} \tag{5}$$

The value $K(\lambda)$ is then given by

$$\begin{aligned} K(\lambda) = 1 - \lambda \underline{c}_{\text{tool}}^{\top} \mathbf{B}_{\text{tool}} \underline{c}_{\text{tool}} - (1-\lambda) \underline{c}_i^{\top} \mathbf{A}_i \underline{c}_i \\ + \underline{w}(\lambda)^{\top} m(\lambda) \ . \end{aligned} \tag{6}$$

The minimum $K_{\min} = \min K(\lambda)$ is found numerically using a fixed-iteration golden-section search, which avoids eigenvalue decompositions and remains computationally lightweight. Two ellipsoids overlap if and only if the minimum value of this function is positive. Beyond overlap detection, this property of the $K(\lambda)$-function also allows an estimation of the penetration depth. Due to its convex nature, as proven in [21], the $K_{\min}$-value scales with the degree of overlap as shown in Fig. 2 for the projection in 2D.

If there is any overlap, we get a set of corresponding overlapping ellipsoids including their opacities, mean positions, and the calculated $K_{\min}$-values. This list is then transferred to the haptic system, where it can be used to counteract by a controller (e.g., an impedance controller).

C. Haptic System

To translate the detected collisions into haptic feedback, we employ a direct volumetric force model as general rendering formulation. We model the interaction between the haptic tool and the virtual 3DGS environment as a virtual spring-damper system. Unlike position-based rendering with fixed spring constant, which yields noisy haptic signals due to the discrete and stochastic nature of the scene representation, our formulation incorporates collision detection outputs for enhanced haptic feedback.

Let $\mathcal{C}_t$ be the set of N intersected Gaussian ellipsoids at time step t. As described in Section III-A, each primitive $i \in \mathcal{C}_t$ is characterised by its center μ_i, opacity α_i and additionally the calculated intersection metric $K_{\min,i}$ derived from the narrow phase (Section III-B2).

a) Weighted Collision Centroid: To determine the effective point of interaction within the volumetric cloud, we employ an importance sampling strategy based on penetration depth. We define a weight

$$w_i = \frac{\exp(K_{\min,i} - \max_K)}{\sum_{j \in \mathcal{C}_t} \exp(K_{\min,j} - \max_K)} \tag{7}$$

for each colliding splat using a softmax function over the intersection metrics $K_{\min,i}$. This ensures that primitives with deeper overlaps contribute significantly more to the force direction than collisions with only marginal contact. The effective collision centroid $\underline{p}_{\text{eff}}$ is then computed as the weighted average of the primitive centers with

$$\underline{p}_{\text{eff}}(t) = \sum_{i \in \mathcal{C}_t} w_i \underline{c}_i \ . \tag{8}$$

The primary direction of the repulsive force, denoted $n(t)$, is the normalised vector from the tool center $\underline{c}_{\text{tool}}$ to this effective centroid. To mitigate high-frequency jitter caused by splats entering and leaving the collision set, we apply an exponential moving average to the force direction.

b) Opacity-Modulated Stiffness: A key challenge in the *haptification* of 3DGS scenes is that the primitives lack explicit physical properties. According to the *Bouguer-Beer-Lambert*-law, the attenuation of light is a direct function of the material's density along the viewing path. We leverage this relationship by assuming that regions of high particle density and thus high opacity correlate with increased mechanical stiffness. Therefore, we modulate the system's stiffness based on the aggregated opacity of the contact region. This provides a pragmatic baseline for translating visual representation into haptic resistance in the absence of semantic material data. The instantaneous weighted opacity $\alpha_{\text{raw}} = \sum w_i \alpha_i$ is smoothed by exponential moving average to produce the continuous opacity signal $\alpha(t)$. The effective stiffness coefficient k_{eff} is then derived as

$$k_{\text{eff}}(t) = k_{\text{base}} \cdot \alpha(t) \ , \tag{9}$$

where k_{base} is a user-defined base stiffness.

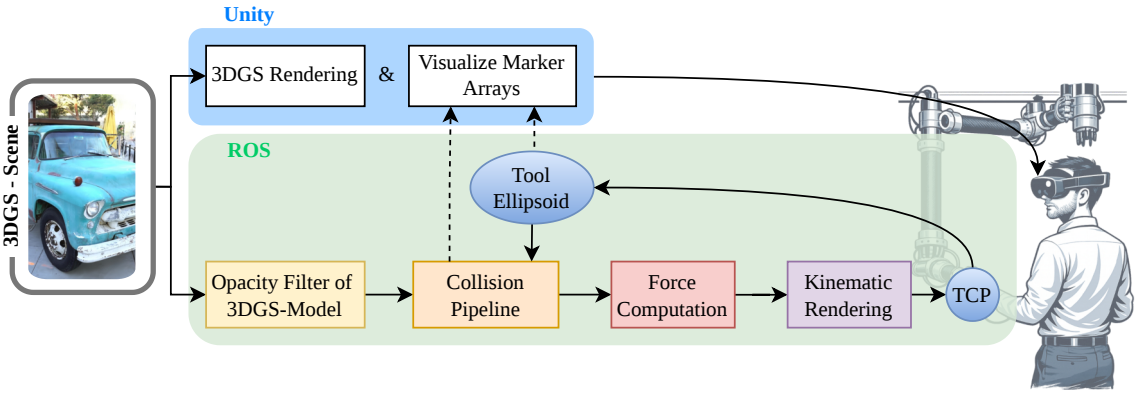

Figure 3: System architecture of the proposed framework integrating 3DGS with ROS-based collision and force computation and *Unity*-based VR visualisation. The opacity-filtered 3DGS model and the tool ellipsoid, controlled by the TCP, feed the collision pipeline. Collisions are visualised in VR and translated into forces driving kinematic rendering on the haptic device.

c) Constitutive Force Equation: The total haptic feedback wrench is defined as a viscoelastic coupling. The elastic magnitude $\underline{F}_{\text{elastic}}$ scales with the volumetric density of the collision, approximated here by the number of active collisions $N = |\mathcal{C}_t|$ and the effective stiffness k_{eff}. A damping term $\underline{F}_{\text{damp}}$ is introduced proportional to the tool's filtered velocity $\underline{v}_{\text{tool}}$ to prevent energy accumulation.

$$\begin{aligned} \underline{F}_{\text{elastic}}(t) &= k_{\text{eff}}(t) \cdot N \\ \underline{F}_{\text{damp}}(t) &= -D \cdot \underline{v}_{\text{tool}}(t) \end{aligned} \tag{10}$$

The final output force $\underline{F}_{\text{out}}$ combines these components, projected along the smoothed normal vector $\underline{n}_{\text{smooth}}$, and is clamped to the maximum capability of the device $F_{\max}$.

$$\begin{aligned} \underline{F}_{\text{raw}} &= \left(\underline{F}_{\text{elastic}}(t) + \underline{F}_{\text{damp}}(t)\right) \cdot \underline{n}_{\text{smooth}}(t) \\ \underline{F}_{\text{out}} &= \text{clamp}\left(\underline{F}_{\text{raw}}, 0, F_{\max}\right) \end{aligned} \tag{11}$$

This density-based volumetric penalty method allows for stiff contacts in dense regions of Gaussian reconstruction, effectively translating visual appearance into haptic texture.

IV. System Architecture

The proposed system integrates 3D Gaussian Splatting, ellipsoid-based collision detection, and impedance-based haptic rendering into a unified framework for interactive exploration of virtual scenes. An overview of the complete architecture is shown in Fig. 3. The design follows a modular separation between *(i)* a visualisation subsystem based on *Unity*, *(ii)* a back end for collision detection and force computation based on Robot Operating System (ROS), and *(iii)* the haptic device that integrates the hardware loop enabling haptic interaction for a user.

The haptic interface provides Cartesian pose measurements of the haptic interaction point (HIP), hereinafter referred to as tool center point (TCP), and accepts Cartesian wrench commands for force feedback. Our experimental setup uses the admittance-type HapticGiant platform. For stable interaction, the real-time ROS nodes for the haptic controller and the proposed feedback pipeline run on a dedicated control PC. The computed wrench setpoint from the collision pipeline is subscribed by the serial kinematic rendering. Given the delta between this impedance-type wrench setpoint and the current external force at the end-effector, the admittance control, defined in [7], converts it into Cartesian acceleration. This Cartesian acceleration target is then mapped to joint commands that respect the kinematic configurations and result in a counter-movement at the end-effector representing collision feedback for the user.

A separate visualisation PC hosts a *Unity* application that renders the original, unfiltered 3DGS scene in virtual reality. The user perceives the virtual environment through a head-mounted display and manipulates the TCP via the haptic device. *Unity* additionally visualises the tool ellipsoid and colliding Gaussian ellipsoids from marker arrays received directly from the ROS collision node. Both machines are connected via a standard Ethernet network and communicate through ROS messages. This decoupled architecture allows high-frequency force rendering on the haptic device, while the computationally more demanding 3DGS rendering and VR visualisation can run at the lower frame rates typical for graphics pipelines.

To enable consistent interaction between the physical workspace and the virtual 3DGS scene, all components share a common world coordinate frame. The filtered Gaussian ellipsoids are first transformed from the 3DGS reconstruction frame into this world frame using the global similarity transform introduced in Section III-A. The same world frame is used as the reference frame of the HapticGiant TCP and is exported via ROS. *Unity* subscribes to the world-frame TCP pose and applies a known rigid transform to place the virtual tool handle and tool ellipsoid at the correct location inside the rendered 3DGS scene. The calibration between the *Unity* camera rig and the world frame ensures visual-proprioceptive alignment: the apparent position of the virtual handle in VR coincides with the physical handle in the user's hand. This alignment is crucial to avoid spatial conflicts and to maintain a coherent multisensory experience.

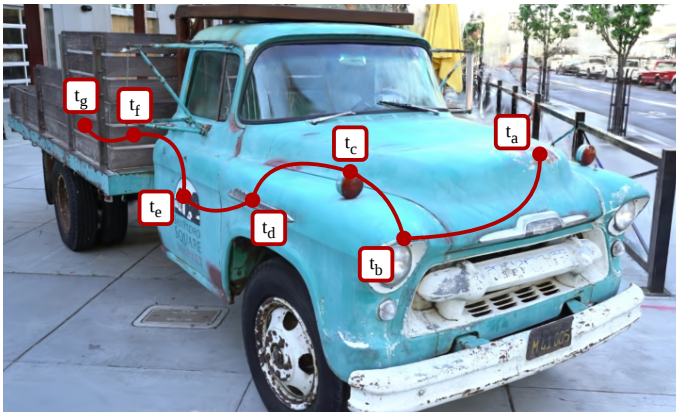

Figure 4: TCP path and contact points in the virtual scene.

V. Evaluation

For validation, we implemented our approach on the HapticGiant, a six degree-of-freedom room-scale kinesthetic haptic interface [7]. This system provides full haptic feedback and enables natural user locomotion within an area of $4.7\,\mathrm{m}$ times $5.5\,\mathrm{m}$, allowing intuitive interactions with large virtual environments. The HapticGiant is designed as an admittance-type haptic interface, while the method proposed in this paper is implemented using impedance control. The impedance formulation offers a clear physical interpretation of how counter-forces are generated and represents the state-of-the-art baseline for haptic interaction. Moreover, as most commercial haptic devices follow the impedance principle, our method demonstrates broad applicability. Extending the proposed approach to admittance control is planned for future work and is expected to further reduce oscillations and improve system stability.

The entire algorithm is executed on a computer with an *Intel Core i5-8600* CPU simultaneously with the control algorithm of the haptic interface. This computer communicates with a dedicated machine for VR-visualisation and 3DGS rendering. The control frequency of the haptic device and for the haptic feedback from 3DGS is both set to $1\,\mathrm{kHz}$.

A. Performance

We evaluate quality and stability of the haptic interaction with a real-world 3DGS dataset. We use a pre-trained vanilla 3DGS model of the Truck-Example from tanks and temples-Dataset [29]. The original model contains 2.5 million Gaussians, which is reduced down to 1 million Gaussians after opacity filtering.

The tool is represented by a generic 3D ellipsoid. Given these models and the system architecture, outlined in Section IV, a user manipulates the TCP through the 3DGS scene with the randomised trajectory shown in Fig. 4 and touches the object with the haptic end effector. In the area of $2.0\,\mathrm{m}$ times $3.0\,\mathrm{m}$, seven contact points between the tool-ellipsoid and the object, captured in the 3DGS scene, are created. In these contact scenarios, both the number of colliding Gaussians and the direction of the resulting normal vector differ. Fig. 5 shows the statistics measured during this interaction. The moments of contact are highlighted by vertical lines.

a) Collision Count: The first row shows the number of kdtree-candidates, filtered and colliding Gaussians. It shows the beneficial usage of the filter strategy. First, the kdtree identifies sets of candidates that contain less than 500 ellipsoids. The prefilter then keeps a subset near to the TCP on which the overlap check is executed. Consequently, the broad phase is critical for performance: the kdtree and bounding-sphere pre-filter effectively reduce the millions of scene primitives to a fraction of relevant candidates, making the computationally demanding analytical narrow phase viable for real-time execution times in most scenarios. It can be seen that each contact with the object can contains up to 100 colliding ellipsoids. The overlap check also demonstrates very high contact accuracy.

b) Repulsive Force: The second row shows the calculated repulsive forces divided in x, y and z-components in world frame. The accuracy of the overlap check consequently affects the detailed force computation. Already single collisions result in low repulsive forces, enabling fast and precise response in terms of gentle haptic feedback. However, no repulsion is calculated without contact. With increasing collision count, the repulsive force also increases, enabling rigid haptic feedback when touching solid objects in the scene. It can also be shown that, due to the deterministic nature of the method, the effective direction of the repulsive force depends on the geometry of the object and the specific contact points. An apparent example can be observed at and immediately after the contact point t_c. Here, the exposed turn signal is first touched from above and then from front. The direction of the repulsive force changes accordingly.

c) User Force: The third row reports the measured user force at the end effector of the HapticGiant interface. The contact points are characterised by significant spikes that indicate repulsion to the users movement. The magnitude depends on how strongly the user pushes against the object. At the contact points t_d and t_g a small oscillatory behaviour can be observed. After the initial contact, the applied user force slightly yields, leading to a minor counter movement of the user and decreasing collision count before reapplying force toward the object. Such effects are commonly observed in impedance-controlled systems and result from the dynamic coupling between human motion and virtual environment feedback. The force level between the contact points corresponds to free locomotion and can be explained by the capability of the haptic device itself.

B. Runtime

Computation time plays a crucial role in collision detection pipelines, especially for haptic applications to enable real-time haptic feedback. We evaluate the runtime performance of the proposed collision detection pipeline by tracking the computation time of the ellipsoid overlap check (see the fourth row of Fig. 5). As expected, the runtime increases with the number of pre-filtered candidates. In dense regions

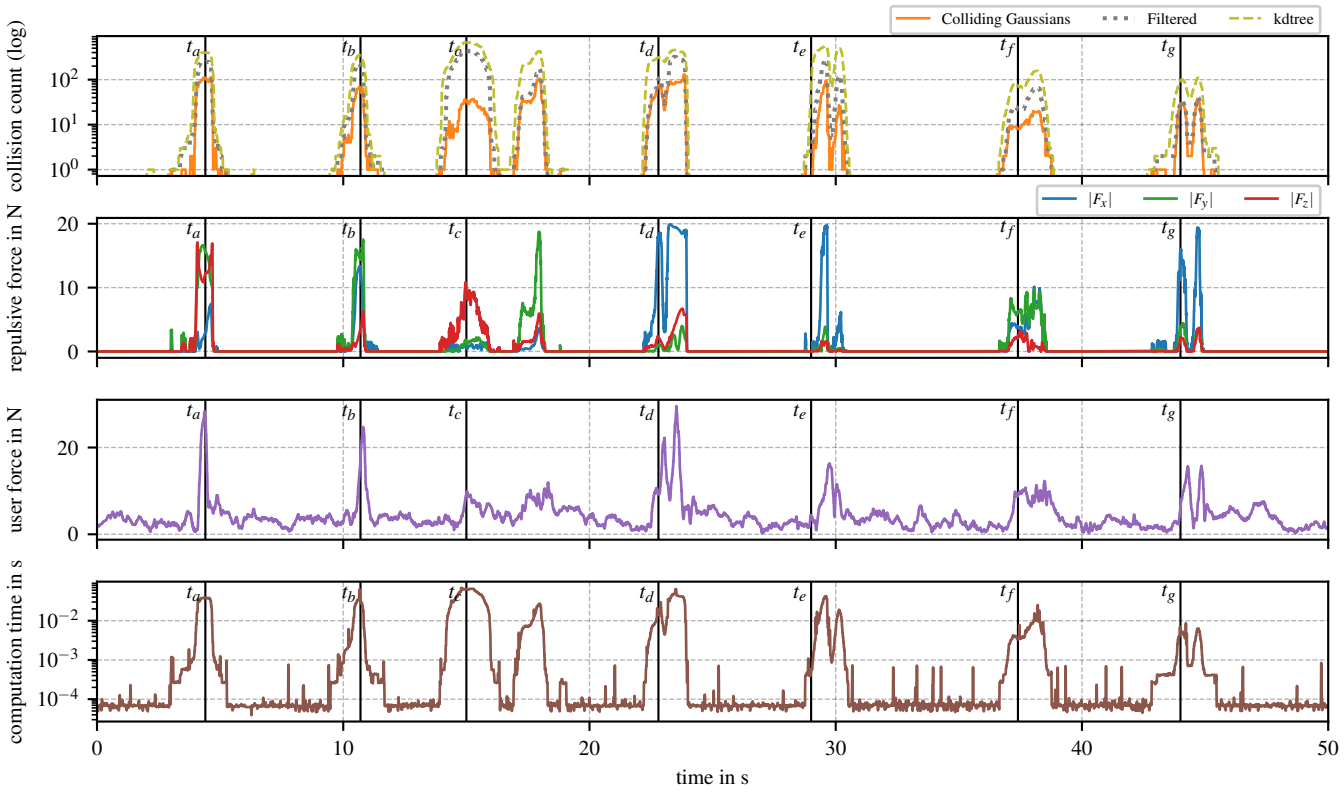

Figure 5: Performance of our approach in a scenario where the user touches various points of the virtual object (Truck) in the 3DGS scene. It shows the collision count including the performance of the filter strategy, calculated repulsive forces, the user force measured at the end effector and the computation time of the collision detection. The moments of contact are highlighted by vertical lines t_a to t_g. All plots are temporally aligned

containing more than 100 possible intersections, the overlap computation requires up to $50\,\mathrm{ms}$. To maintain real-time stability despite these latency peaks, the system relies on its decoupled asynchronous architecture. The haptic impedance controller strictly operates at $1\,\mathrm{kHz}$, temporarily sustaining the most recently computed valid force output until the next collision update from the ROS node arrives. However, a fast calculation of the initial contact and the first sequence of contact points is essential. Here, computation time remains mainly below $1\,\mathrm{ms}$. This performance enables synchronisation with the haptic control loop running at $1\,\mathrm{kHz}$, thus ensuring smooth force rendering and high system responsiveness.

Overall, the evaluation demonstrates that the proposed 3DGS-based haptic rendering pipeline achieves accurate and stable haptic feedback with real-time performance, effectively handling dense real-world environments, while preserving precise contact dynamics and responsive force rendering.

VI. Discussion

While the proposed framework enables real-time, direct haptic rendering from 3D Gaussian Splatting scenes, several limitations remain. First, the approach is currently restricted to static environments, as the underlying 3DGS representation does not support dynamic or deformable geometry. Consequently, time-varying objects or physically evolving scenes cannot be modelled consistently. Furthermore, haptic fidelity depends on reconstruction quality: sparse regions, occlusions, or holes in the Gaussian model may lead to missing or inconsistent contacts and thus absent force feedback in poorly observed or non-visible areas.

The present system entirely relies on impedance-based force rendering. More advanced strategies, such as admittance control, or energy-consistent coupling schemes, may further improve stability and realism and are therefore the subject of ongoing developments. In addition, the current force model considers only normal volumetric interactions. Friction, surface texture, and material-dependent properties are not yet represented, which limits perceived realism. The interaction tool is approximated by a single ellipsoid, which restricts the accurate modelling of complex tool geometries and multi-point contacts. Richer contact models incorporating texture and material parameters as well as more expressive tool and multi-body representations will therefore be investigated in future research.

Finally, the evaluation focuses primarily on runtime performance and qualitative interaction behavior. Quantitative validation against geometric or force ground truth, as well as controlled user studies assessing perceptual realism and usability, remains future work.

VII. Conclusion

This paper presented a method for direct haptic rendering of 3DGS scenes, enabling real-time collision detection and force feedback from photorealistic scene reconstructions obtained purely from unstructured 2D images. The core idea is to interpret each Gaussian as an ellipsoidal collision primitive and to apply an analytical overlap test within a hierarchical pipeline. The resulting contact parameters—penetration depth, primitive positions, and opacities—feed into a multi-contact force blending scheme that derives both force direction from a weighted collision centroid, and force magnitude from opacity-modulated stiffness, effectively translating visual density into haptic resistance. Validation on the room-scale HapticGiant interface using a real-world 3DGS model confirmed geometry-dependent force directions and stable haptic feedback at a $1\,\mathrm{kHz}$ control rate on a standard CPU. By operating directly on the learned scene representation, the framework maintains full consistency between what the user sees and what the user feels. This work demonstrates that a set of 2D images is sufficient to create an immersive, haptically interactive 3D environment, establishing 3DGS as a unified representation for visual and haptic rendering in telepresence systems.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[2] S. Zhu, G. Wang, X. Kong, D. Kong, and H. Wang, "3d Gaussian splatting in robotics: A survey," *arXiv:2410.12262*, 2024.

[3] H. Zhou, J. Shao, L. Xu, D. Bai, W. Qiu, B. Liu, Y. Wang, A. Geiger, and Y. Liao, "Hugs: Holistic urban 3d scene understanding via gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 336–21 345.

[4] D. Yang, Y. Gao, X. Wang, Y. Yue, Y. Yang, and M. Fu, "Opengs-slam: Open-set dense semantic slam with 3d gaussian splatting for object-level scene understanding," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8486–8492.

[5] Y. Lee, H. Kim, H. Ji, J. Heo, Y. Lee, J. Kang, J. Lee, and D. Lee, "Human-in-the-loop gaussian splatting for robotic teleoperation," *IEEE Robotics and Automation Letters*, vol. 11, no. 1, pp. 105–112, 2025.

[6] S. Yang, Q. Li, D. Shen, B. Gong, Q. Dou, and Y. Jin, "Deform3dgs: Flexible deformation for fast surgical scene reconstruction with gaussian splatting," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2024, pp. 132–142.

[7] M. Fennel, M. Walker, D. Pikos, and U. D. Hanebeck, "HapticGiant: A novel very large kinesthetic haptic interface with hierarchical force control," *IEEE Transactions on Haptics*, pp. 1–14, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/11125831

[8] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM siggraph papers*, 2006, pp. 835–846.

[9] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *11th international conference on computer vision*. IEEE, 2007, pp. 1–8.

[10] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, no. 4, 2006.

[11] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 2002.

[12] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Trans. Graph.*, vol. 13, no. 1, p. 43–72, Jan. 1994.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[14] S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones, "Adaptively sampled distance fields: A general representation of shape for computer graphics," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 249–254.

[15] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *10th international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.

[16] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[17] Y. Bao, T. Ding, J. Huo, Y. Liu, Y. Li, W. Li, Y. Gao, and J. Luo, "3d gaussian splatting: Survey, technologies, challenges, and opportunities," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 7, pp. 6832–6852, 2025.

[18] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, "Physgaussian: Physics-integrated 3d gaussians for generative dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 4389–4398.

[19] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99, USA, 1999, p. 401–408.

[20] H. Xu and J. Barbič, "6-dof haptic rendering using continuous collision detection between points and signed distance fields," *IEEE Transactions on Haptics*, vol. 10, no. 2, pp. 151–161, 2017.

[21] I. Gilitschenski and U. D. Hanebeck, "A direct method for checking overlap of two hyperellipsoids," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2014)*, Bonn, Germany, Oct. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6954724/

[22] D. Leprich, M. Rosenfelder, M. Herrmann-Wicklmayr, K. Flaßkamp, P. Eberhard, and H. Ebel, "Efficient collision-avoidance constraints for ellipsoidal obstacles in optimal control: Application to path-following mpc and uavs," *arXiv:2510.26531*, 2025.

[23] T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager, "Splat-nav: Safe real-time robot navigation in gaussian splatting maps," *IEEE Transactions on Robotics*, 2025.

[24] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *Proceedings ieee/rsj international conference on intelligent robots and systems. Human robot interaction and cooperative robots*, vol. 3. IEEE, 1995, pp. 146–151.

[25] D. C. Ruspini, K. Kolarov, and O. Khatib, "Haptic interaction in virtual environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97*, vol. 1. IEEE, 1997, pp. 128–133.

[26] M. Otaduy and M. Lin, "Stable and responsive six-degree-of-freedom haptic manipulation using implicit integration," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, 2005, pp. 247–256.

[27] F. Ryden and H. J. Chizeck, "A proxy method for real-time 3-dof haptic rendering of streaming point cloud data," *IEEE transactions on Haptics*, vol. 6, no. 3, pp. 257–267, 2013.

[28] H. Zhang, L. Zhu, Y. Xiang, J. Zheng, and A. Song, "Haptic rendering of neural radiance fields," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–10.

[29] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.