

LoDAvatar: Hierarchical Embedding and Adaptive Levels of Detail with Gaussian Splatting for Enhanced Human Avatars

Xiaonuo Dongye

Beijing Institute of Technology

Yihua Bao

Beijing Institute of Technology

Hanzhi Guo

Beijing Institute of Technology

Zeyu Tian

Beijing Institute of Technology

Le Luo*

Peng Cheng Laboratory

Haiyan Jiang

Beijing Institute of Technology

Dongdong Weng†

Beijing Institute of Technology Zhengzhou Research Institute

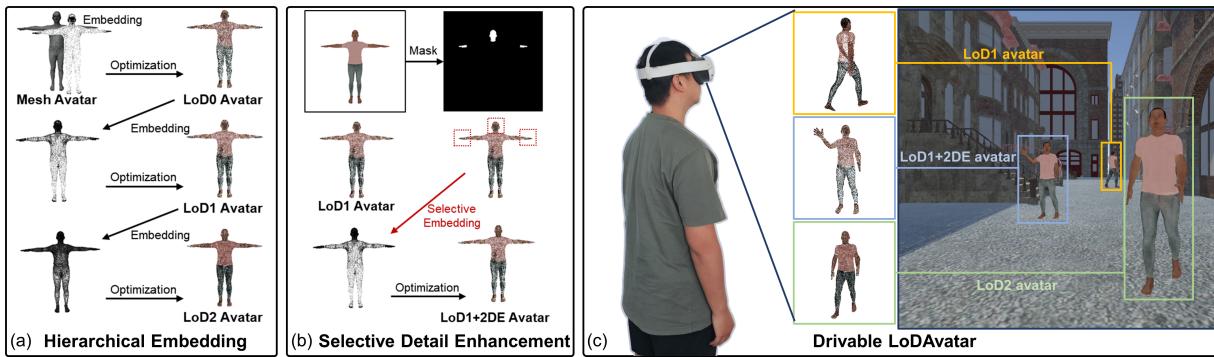


Figure 1: The LoDAvatar introduces levels of detail on Gaussian avatars, achieving a balance between visual quality and computational costs. The implementation of LoDAvatar relies on hierarchical embedding and selective detail enhancement methods. (a) The hierarchical embedding method starts from a base mesh avatar and progressively generates Gaussian avatars, ranging from lower to higher LoD. (b) Selective detail enhancement allows detail to be enhanced only at specific areas using image masks, allowing precise control of the number of Gaussians, thereby reducing computational costs. (c) Avatars generated through LoDAvatar can be driven in real-time and are well-suited for integration into VR.

ABSTRACT

With the advancement of virtual reality, the demand for 3D human avatars is increasing. The emergence of Gaussian Splatting technology has enabled the rendering of Gaussian avatars with superior visual quality and reduced computational costs. Despite numerous methods researchers propose for implementing drivable Gaussian avatars, limited attention has been given to balancing visual quality and computational costs. In this paper, we introduce LoDAvatar, a method that introduces levels of detail into Gaussian avatars through hierarchical embedding and selective detail enhancement methods. The key steps of LoDAvatar encompass data preparation, Gaussian embedding, Gaussian optimization, and selective detail enhancement. We conducted experiments involving Gaussian avatars at various levels of detail, employing both objective assessments and subjective evaluations. The outcomes indicate that incorporating levels of detail into Gaussian avatars can decrease computational costs during rendering while upholding commendable visual quality, thereby enhancing runtime frame rates. We advocate adopting LoDAvatar to render multiple dynamic Gaussian avatars or extensive Gaussian scenes to balance visual quality and computational costs.

Index Terms: Computing methodologies Computer graphics→ Rendering; Point-based models.

1 INTRODUCTION

The progression of virtual reality (VR) technologies has increased the demand for realistic 3D human avatars [1]. Traditional methods of crafting 3D human avatars often involve utilizing scan data

or conducting 3D modeling based on multi-view images, resulting in mesh avatars stored in vertex and face formats [2]. In recent years, the introduction of 3D Gaussian Splatting (3DGS) [3] technology has opened up new avenues for generating human avatars. 3D Gaussian Splatting is an innovative rendering technique designed for real-time rendering of virtual objects and scenes. In contrast to conventional methods that rely on points and meshes for virtual object and scene construction, 3DGS is presented as a flexible and expressive representation [4]. Anisotropic 3D Gaussians can accurately depict high-quality radiation fields, and these Gaussians are explicit and well-suited for rapid GPU-based rasterization [5]. This capability enables rendering high-quality virtual avatars in VR while reducing computational costs and achieving high frame rates during rendering [6]. In creating dynamic Gaussian avatars, researchers have delved into the methodologies of driveable 3D Gaussian Splatting [6]. Driveable 3D Gaussian Splatting embeds Gaussians onto the surface of the corresponding mesh avatar, whereby transforming Gaussians from the world coordinate system to the local coordinate system on the surface of the corresponding mesh triangle [7]. This allows the Gaussians to change along with the mesh model, enabling the dynamic rendering of Gaussian avatars. Due to its reduced data storage requirements and the capacity to guide avatars in performing actions beyond the captured data set, driveable 3DGS is extensively employed in dynamic Gaussian avatars [8]. This method can generate Gaussian avatars based on existing mesh avatars and has been widely used in previous research such as Gaussian avatars [9] and splatting avatars [10].

While significant research efforts have concentrated on achieving dynamic Gaussian avatars, little attention has been paid to balancing the visual quality and computational costs. Increasing the number of Gaussians employed in avatar generation can heighten visual quality but concurrently escalate the computational costs during avatar driving [11]. Real-time interaction with virtual avatars is paramount in VR, underscoring the necessity to render avatars with

*e-mail: leluo1989@gmail.com

†e-mail: crgj@bit.edu.cn

minimal computational costs to attain higher display frame rates. Our motivation lies in utilizing a manageable number of Gaussians to generate avatars and introduce levels of detail (LoD) on Gaussian avatars to better leverage the advantages of high visual quality and low computational costs inherent in Gaussian Splatting.

To strike a balance between superior visual quality and minimized computational costs in Gaussian avatars, this paper introduces LoDAvatar, which generates Gaussian avatars with varying LoD through hierarchical embedding and selective detail enhancement methods, as depicted in Fig. 1. The methodology comprises four phases: data preparation, Gaussian embedding, Gaussian optimization, and selective detail enhancement. In the data preparation phase, a mesh avatar is initially crafted using the mesh and corresponding texture maps as inputs. Key frame animations are generated for the mesh avatar, and a series of multi-view images of the key frames, along with their corresponding camera parameters, are recorded. Subsequently, the Gaussian embedding involves establishing a local coordinate system on each triangle face of the mesh avatar. Gaussians are initialized at the vertices and surface centers of each triangle, with their parameters transformed from the world coordinate system to the local coordinate system. Following Gaussian embedding, we conduct Gaussian optimization by constraining the positions of the Gaussians at the triangle vertices and maintaining a constant number of Gaussians. After optimization, the Gaussians at the original triangle centers are repositioned, and they are connected to the positions of the vertices' Gaussians to form new triangles. In the selective detail enhancement phase, image masks are employed to identify triangles necessitating detail enhancement. New Gaussians are embedded on the corresponding triangle faces to achieve selective detail enhancement. Subsequent optimization procedures maintain the low-detail Gaussians fixed while refining solely the newly introduced Gaussians in each iteration, progressively enhancing the avatar's details and generating Gaussian avatars spanning from low to high levels of detail. The avatars, which can be driven and used for interactions in VR, generated by the hierarchical embedding and selective detail enhancement techniques, demonstrate superior visual quality and reduced computational costs.

In summary, our main contributions are threefold:

1. We introduce a novel method for hierarchical embedding within mesh avatars, facilitating the drivable Gaussian avatars. The hierarchical embedding method enables the generation of human avatars with varying levels of detail by regulating the number of Gaussians, thereby striking a balance between visual quality and computational costs.
2. We use selective detail enhancement for area-controllable levels of detail by incorporating the image mask in conjunction with hierarchical embedding. By enhancing the avatar's facial and manual features, we imbue the avatar with enhanced details, rendering it more suitable for real-time interactions within VR.
3. We evaluate our proposed method through objective and subjective evaluations to showcase the high visual quality and reduced computational costs.

2 RELATED WORK

2.1 Gaussian Splatting for Virtual Objects Generation

Various approaches are employed in VR to represent virtual objects [12], including meshes [13], point clouds [14], grids [15], and neural radiance fields (NeRF) [16]. The recent emergence of 3D Gaussian Splatting technology has introduced a novel approach to generating high-quality virtual objects [17]. This technology utilizes multi-view images to initialize a Structure from Motion (SfM) point cloud, which is subsequently transformed into 3D Gaussians [3]. These points are projected onto the image plane using aligned cameras and rendered through differentiable rasterization. Following this, the rendered image is compared to the input multi-view images

to compute loss, update the parameters within the 3D Gaussians, and adjust the Gaussians through adaptive density control. We summarize the notable features of Gaussian Splatting as follows: (1) The image-based 3D reconstruction approach expedites rapid data acquisition and virtual object generation. (2) It ensures the visual quality of virtual objects. (3) It incurs lower computational costs.

Gaussian Splatting technology demonstrates strong performance in scene generation [18–20], real-time rendering [21–23], and digital avatar creation [24, 25]. In avatar creation, researchers have focused on rendering dynamic face [26] and full-body avatars [27]. For instance, Saito *et al.* introduce relightable Gaussian codec avatars to create high-fidelity relightable head avatars with animated expressions [28]. Zheng *et al.* present GPS-Gaussian, a method for synthesizing novel views of characters in real time [29]. Given the impressive attributes, 3DGS has emerged as a significant method for generating virtual objects and avatars in VR [6].

2.2 Dynamic Gaussian Avatar and Gaussian Embedding

While 3DGS has demonstrated exceptional rendering capabilities for static objects and environments, integrating dynamic objects and avatars into static scenes presents significant challenges [30]. Researchers have explored two primary approaches for dynamic Gaussians, *i.e.* 4D Gaussian Splatting and drivable 3DGS [6]. The 4D Gaussian Splatting introduces a temporal dimension to the 3DGS, capturing Gaussian objects at each frame through 4D data acquisition and rendering them frame by frame [31]. A notable example of this approach is HiFi4G [32], which combines 3D Gaussian representation with non-rigid tracking to achieve a compact and compression-friendly representation. Drivable 3DGS involves dynamically adjusting the parameters of individual Gaussians within the Gaussian objects during rendering to achieve dynamic effects [7]. This approach, requiring less data storage, has found wide application in dynamic scenes [33], facial expressions [34], hair rendering [35], body avatars [36, 37], *etc.* Gaussian embedding is a crucial method in the implementation of drivable 3DGS. The fundamental concept of Gaussian embedding involves rigging the Gaussian representation to a parametrically deformable mesh [38]. As the mesh deforms, the embedded Gaussians on its surface adjust accordingly. For instance, Qian *et al.* introduced Gaussian Avatars [9], which utilize Gaussian initialization at the surface center of each mesh to create a series of dynamically bound Gaussians through adaptive density control. Shao *et al.* propose Splatting Avatar [10], employing trainable Gaussian embedding on a standard mesh. This method generates a set of Gaussians randomly on the mesh surface and refines them using the walking on triangles [10] technique. Furthermore, Jiang *et al.* developed the VR-GS system [39], a highly efficient two-level embedding strategy that enables the interactive representation of physical dynamics-aware Gaussians in VR. Despite the various Gaussian embedding methods researchers have proposed, the challenge of efficiently implementing Gaussian embedding remains unresolved [30]. Our research, inspired by the principles of Gaussian embedding, aims to devise low computational loss Gaussian embedding.

2.3 Levels of Detail

Levels of detail play a crucial role in managing the intricacy of virtual environments to strike a balance between complexity and performance in the realm of computer graphics [40]. The LoD can be adjusted based on the object's distance from the viewer, object significance, or position [41, 42]. By incorporating LoD techniques, rendering efficiency is bolstered through a reduction in the workload on graphics pipelines. The slight decrease in visual fidelity of distant or less-significant objects is often imperceptible due to its minimal impact on object appearance [43]. In recent years, LoD techniques have garnered significant interest in the domain of the neural radiance field. There are two primary pathways for realiz-

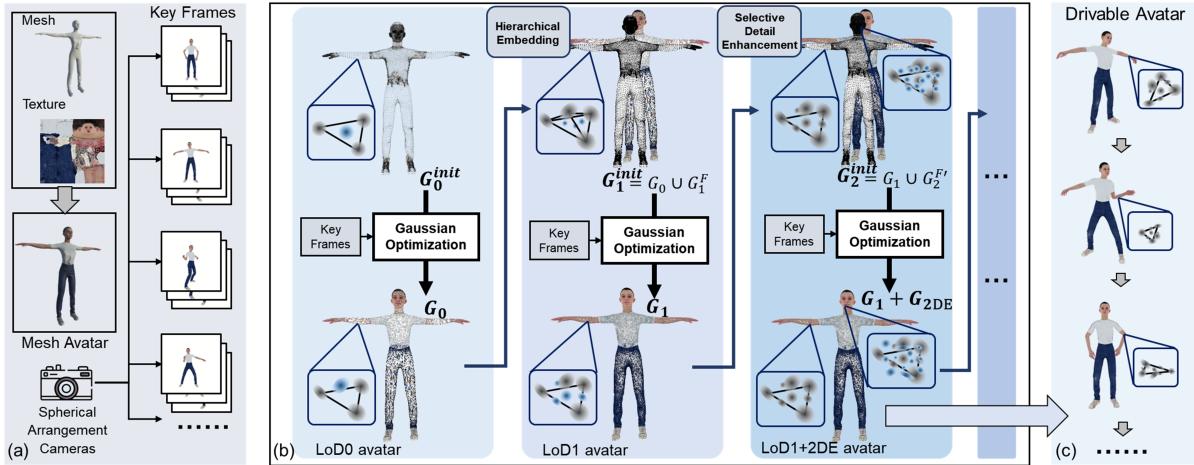


Figure 2: The pipeline of LoDAvatar. (a) In the data preparation phase, using mesh and texture as input, multi-view key frame images are captured by cameras arranged in a spherical space, and the corresponding camera parameters are recorded. (b) Local coordinate systems are established on the mesh’s triangle faces, followed by the embedding of Gaussians and subsequent Gaussian optimization. Various LoDAvatars are generated via hierarchical embedding and selective detail enhancement. (c) The Gaussians are embedded in the local coordinate systems of the triangle faces, synchronizing with the movements of the mesh to create drivable Gaussian avatars.

ing objects with varying LoD: transitioning from low detail to high detail and vice versa [44]. Noteworthy works in enhancing detail from low levels include BungeeNeRF [45], Mip-NeRF [46], and LoD-NeuS [47]. BungeeNeRF enhances NeRF by progressively activating high-frequency channels in NeRF’s positional encoding inputs, gradually unveiling more intricate details as training progresses. Notable works in reducing detail from high to low include NGLoD [48], which represents implicit surfaces using an octree-based feature volume that adeptly fits shapes with multiple discrete LoDs. With advancements in 3DGS techniques, researchers have started exploring modeling different LoD in explicit 3D Gaussian scenes. For instance, Fischer *et al.* [49] integrated 3D Gaussians as an efficient geometry scaffold while utilizing neural fields as a compact and flexible appearance model. CityGaussian [11] generates varying detail levels through the progressive compression strategy LightGaussian [50], operating directly on trained Gaussians to achieve substantial compression rates with minimal performance degradation.

In contrast to studies focusing on modeling large scenes, our work centers on implementing LoD on dynamic avatars. Our method generates drivable Gaussian avatars with a controllable number of Gaussians, transitioning from low to high detail using hierarchical embedding and selective detail enhancement. This method reduces computational overhead while upholding exceptional visual quality.

3 METHOD

3.1 Overview

In Section 3, we elaborate primarily on utilizing two methods, hierarchical embedding, and selective detail enhancement, to produce Gaussian avatars characterized by high visual quality and minimal computational costs. Our methodology is delineated into four primary phases: data preparation, Gaussian embedding, Gaussian optimization, and selective detail enhancement. The data preparation phase is instrumental in generating the requisite model and training data for Gaussian embedding, as depicted in Fig. 2(a). Subsequently, the Gaussian embedding, Gaussian optimization, and selective detail enhancement phases are instrumental in crafting Gaussian avatars with varying LoD, as illustrated in Fig. 2(b). These resultant Gaussian avatars are capable of real-time driving and application in interactive VR settings, as demonstrated in Fig. 2(c). Given the advancements over traditional 3DGS offered, we initially expound upon the fundamentals of 3DGS in Section 3.2, followed

by a detailed exposition of the four sequential phases of our methods in Sections 3.3- 3.6, respectively.

3.2 Preliminary

The 3DGS technique is employed to depict virtual objects or scenes using anisotropic 3D Gaussians, which are determined by the image and camera parameters. The virtual entities are represented by a 3D Gaussian series with 3D covariance matrix Σ and mean μ .

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (1)$$

To optimize these Gaussians efficiently through gradient descent, Kerbl *et al.* introduce parametric ellipses by utilizing a scaling matrix S and a rotation matrix R to construct the covariance matrix.

$$\Sigma = RSS^T R^T \quad (2)$$

During the rendering process, the projection of Gaussians from 3D space to a 2D image plane is executed via a view transformation W and the Jacobian of the affine approximation of the projective transformation J . The covariance matrix Σ' in the 2D image plane can be calculated as

$$\Sigma' = JW\Sigma W^T J^T \quad (3)$$

The color C of a pixel is determined by blending all overlapping 3D Gaussians with

$$C = \sum_{i=1}^N c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (4)$$

where c_i represents the color of each point and α_i is derived by evaluating a 2D Gaussian with covariance Σ multiplied by a learned per-point opacity.

In summary, as a representation of a virtual object or scene, the 3D Gaussians encompass the following parameters: (1) World Position $X \in R^3$, (2) World Rotation $r \in R^4$ expressed as quaternion, (3) Scaling Factor $s \in R^3$, (4) Spherical Harmonics Coefficients for color information $h \in R^{48}$, and (5) Opacity $\alpha \in [0, 1]$.

3.3 Data Preparation

Data preparation serves as the foundational phase within our methodology. During this stage, the preparation involves creating a triangle-based mesh avatar for Gaussian embedding and compiling multi-view key frame images and corresponding camera parameters for Gaussian optimization. In our methodology, Gaussian avatars are generated from mesh avatars and textures rather than directly from multi-view images. This decision is based on the fact that avatars generated directly from multi-view images lack topological consistency. This inconsistency prevents the establishment of a consistent

local coordinate system on the same triangle faces of the mesh under different key frames, rendering the avatar unable to be driven dynamically. In our methodology, we illustrate this process using the Skinned Multi-Person Linear Model (SMPL) [51] as a primary example. The SMPL avatar, a vertex-based 3D human representation, encompasses 6890 vertices and 13776 triangular faces, enabling the depiction of diverse human body shapes and poses through 10 shape parameters and 23 joint points) [51]. The SMPL can be parameterized by fitting from multi-view real-world images and has been extensively employed as a standard avatar in prior research [52]. The mesh avatar and textures can also be sourced from platforms such as Mixamo [53]. Key frame animations are generated from this avatar along with the associated texture maps. Subsequently, 42 virtual cameras are strategically positioned around the SMPL avatar, each with distinct angles and identical internal parameters, forming a spherical arrangement. The SMPL avatar is centrally located amidst these cameras within a spherical configuration with a radius of 2 meters, ensuring comprehensive avatar coverage during the filming process. Multi-view images of the avatar are captured at each key frame, maintaining a resolution of 1080×1080 , while the corresponding camera parameters are recorded to generate training data for Gaussian optimization. During the forthcoming Gaussian optimization phase, these key frames will be leveraged to optimize identical Gaussians.

3.4 Gaussian Embedding

Gaussian embedding is conducted on the mesh avatar as outlined in Section 3.3. The primary objective of Gaussian embedding is to establish a linkage between the avatar represented with mesh and the avatar represented with Gaussians. Initially, the world coordinates of the vertices on each triangle surface of the mesh avatar are acquired. For each triangle, a local coordinate system is established with the center position of the triangle serving as the origin. The z-axis is aligned with the normal direction of the triangle face, the x-axis points towards the first vertex, and the y-axis is determined as the cross of z and x directions. Subsequently, we select four locations, the three vertices and the triangle’s center, and initialize the Gaussians, as shown in Fig. 3(a). Within the local coordinate system of the triangle, each Gaussian is endowed with four new variables: idx , X^l , r^l , and s^l . Here, idx denotes the triangle face number to which the Gaussians are embedded; X^l signifies the Gaussian’s local position on the triangle face; r^l represents the local rotation; and s^l denotes the local scale. The embedded Gaussian’s local positions, rotations, and scales can be defined by:

$$G = \begin{cases} X^w = k \cdot R \cdot X^l + T \\ r^w = R \cdot r^l \\ s^w = k \cdot s^l \end{cases} \quad (5)$$

where R denotes the rotation matrix from local to the world coordinate system; T signifies the translation matrix; k represents the scaling factor for the triangle face area variation upon mesh movement; the color parameter h and transparency α remain constant. The Gaussians embedded in the triangles adjust accordingly as the mesh avatar moves. Consequently, the mesh-Gaussian embedding on triangles are established, yielding the initialized Gaussian parameter G_0^{init} .

3.5 Gaussian Optimization

The original 3DGS encompasses two primary optimization steps: Gaussian parameter optimization and adaptive density control. To attain Gaussian avatars characterized by distinct LoD, our method omits the utilization of adaptive density control within the Gaussian optimization phase. Specifically, we avoid the splitting and pruning sessions to maintain a constant number of Gaussians. This decision is motivated by the aim to ensure that each Gaussian, at varying LoD,

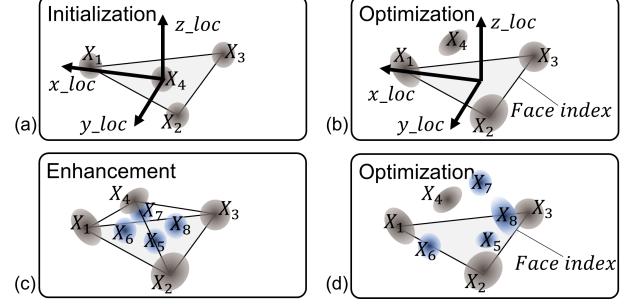


Figure 3: Hierarchical embedding on the triangle faces. (a) Establishing local coordinate systems and initializing Gaussians. (b) Fixing the position of the Gaussians at the vertices and performing Gaussian optimization. (c) Connecting the positions of optimized Gaussians, forming new triangle faces, and initializing new Gaussians. d) Fixing existing Gaussians and optimizing new Gaussians to achieve enhanced LoD.

effectively retains the essential characteristics of the human avatar throughout the optimization process. The Gaussian parameters at lower LoD are held constant in each successive iteration. Adding details are incorporated into the avatar at lower LoD by introducing and optimizing new Gaussians, thereby establishing a hierarchical embedding framework for Gaussian avatars.

We extract the camera parameters and project the initialized G_0^{init} onto the image planes, comparing them with the multi-view key frame images to compute the loss function $L = 0.8 \times L_I + 0.2 \times L_{DSSIM}$ for optimizing the position, rotation, scaling, opacity, and color coefficient. Here L_I is the L_1 loss between the rendered image and the original image. L_{DSSIM} represents the structural similarity index measure loss between the images. The initialized Gaussians G_0^{init} is divided into two components: Gaussians at the vertices of the triangle faces G_0^V and Gaussians on the triangle faces G_0^F , represented as

$$G_0^{init} = G_0^V \cup G_0^F = \{X_0^V, r_0^V, s_0^V, h_0^V, \alpha_0^V\} \cup \{X_0^F, r_0^F, s_0^F, h_0^F, \alpha_0^F\} \quad (6)$$

During the optimization process, we maintain the position parameter X_0^V fixed, and solely optimize the parameters $r_0^V, s_0^V, h_0^V, \alpha_0^V$ of G_0^V and G_0^F , as depicted in Fig. 3(b). Following optimization, we obtain the Gaussian avatar G_0 at the lowest level of detail, denoted as

$$\begin{aligned} G_0 &= G_0^V \cup G_0^F \\ &= \{X_0^V, r_0^V, s_0^V, h_0^V, \alpha_0^V\} \cup \{X_0^F, r_0^F, s_0^F, h_0^F, \alpha_0^F\} \\ &= G_1^V \end{aligned} \quad (7)$$

When G_0 is driven, the Gaussians on each face adjust with their corresponding triangles.

Subsequently, additional details are incorporated into G_0 to achieve a Gaussian avatar with a higher LoD. Post the initial optimization, the Gaussians at the center of the triangle faces transition to new local positions $X_0^{F'}$. For each triangle in the mesh avatar, the position of each $X_0^{F'}$ is linked to the vertices of its respective triangle face, forming three new triangle faces, as illustrated in Fig. 3(c). Four new Gaussians are initialized at the center positions of the newly formed triangle faces and the original triangle face. At this stage, G_0 can be viewed as the Gaussian at the vertex positions, while the newly embedded Gaussians can be seen as the Gaussians on the faces, represented as

$$G_1^{init} = G_1^V \cup G_1^F = G_0 \cup \{X_1^F, r_1^F, s_1^F, h_1^F, \alpha_1^F\} \quad (8)$$

In the subsequent optimization step, we maintain the parameters of G_0 fixed, reintroduce the multi-view key frame images and camera parameters, and optimize solely the newly introduced G_1^F to capture additional details in the avatar. This iterative process enables the

gradual augmentation of details while managing the number of Gaussians through hierarchical embedding. The Gaussian avatar at a higher level of detail is denoted as

$$G_1 = G_1^V \cup G_1^{F'} = G_0 \cup \{X_1^{F'}, r_1^{F'}, s_1^{F'}, h_1^{F'}, \alpha_1^{F'}\} = G_2^V \quad (9)$$

At this juncture, five Gaussians are embedded in each triangle face of the original mesh, all adjusting with their corresponding triangle faces when manipulated, as depicted in Fig. 3(d).

To generate Gaussian avatars with increased LoD, connecting $\{X_5, X_1\}$, $\{X_5, X_2\}$, $\{X_5, X_3\}$, $\{X_6, X_1\}$, $\{X_6, X_2\}$, $\{X_6, X_4\}$, $\{X_7, X_1\}$, $\{X_7, X_3\}$, $\{X_7, X_4\}$, $\{X_8, X_2\}$, $\{X_8, X_3\}$, $\{X_8, X_4\}$ results in creating 12 new triangles, initializing the embedding of $(4 + 12 = 16)$ additional Gaussians on the surface. By repeating this process iteratively and continuously optimizing the newly generated Gaussians, we can progressively enhance the existing Gaussian avatar. In scenarios where the mesh avatar employed for embedding is an SMPL avatar, following 3 to 4 iterations of Gaussian embedding and optimization, the formation of G_2 and G_3 avatars, each comprising $296k - 1.17M$ Gaussians, can yield a visually superior Gaussian avatar observable from diverse viewpoints.

3.6 Selective Detail Enhancement

In Section 3.5, we employ a hierarchical embedding method in iterative cycles to transform a low-detail Gaussian avatar into a high-detail representation. Within each iteration, every triangle face of the original mesh avatar undergoes subdivision into four new triangle faces, accompanied by the initialization of new Gaussians on these faces. On top of this, the selective detail enhancement method is introduced to regulate the division of triangle faces and the initialization of new Gaussians. The core concept behind selective detail enhancement lies in achieving targeted enhancement by selectively addressing specific triangle faces, embedding, and initializing new Gaussians while creating a more detailed Gaussian avatar. Past studies have underscored the substantial impact of the quality on avatars' faces [54, 55] and hands [56] on interactions within VR [57]. Hence, enhancing designated details can be accomplished by selectively implementing Gaussian embedding on the facial and manual triangles of the mesh avatar. This method offers the advantage of enabling additional control over the number of Gaussians, facilitating the acquisition of crucial details for the face and hands of the Gaussian avatar while minimizing additional computational costs. Consequently, this strategy enhances the user's visual subjective experience within the VR.

Given the challenge of directly obtaining the indices of face and hands triangles on the mesh avatar, we employ image masks to identify the triangles corresponding to the avatar's head and hands. By selecting one or more images from the key frame images and performing image segmentation, we can derive the image masks delineating the avatar's head and hands. In the scenario illustrated in Fig. 4, we opt for a frontal view image captured when the avatar assumes a T-pose, along with the associated internal and external camera parameters, denoted as K and $[R|T]$ for this specific image. Leveraging the avatar, we can derive the world coordinates $[x, y, z]$ of all vertex positions within the mesh avatar. Subsequently, employing the prevalent image segmentation algorithm SAM [58], we extract the masks representing the avatar's head and hands. Following this, we project all vertex positions within the mesh avatar onto the image plane based on the internal and external camera parameters corresponding to the image, denoted as $[u, v, w] = K[R|T][x, y, z, 1]^T$. This projection process assists in determining whether a vertex falls within the designated mask. Upon confirming that all three vertices of a triangle lie within the mask, we record the indices of these vertices, forming the triangle face. During a specific iteration transitioning from the low-detail to high-detail avatar, these indices can be selectively utilized for generating new triangle faces with Gaussian embedding. This targeted approach aims to enhance the facial and manual areas of the Gaussian avatar, imbuing the avatar with crucial

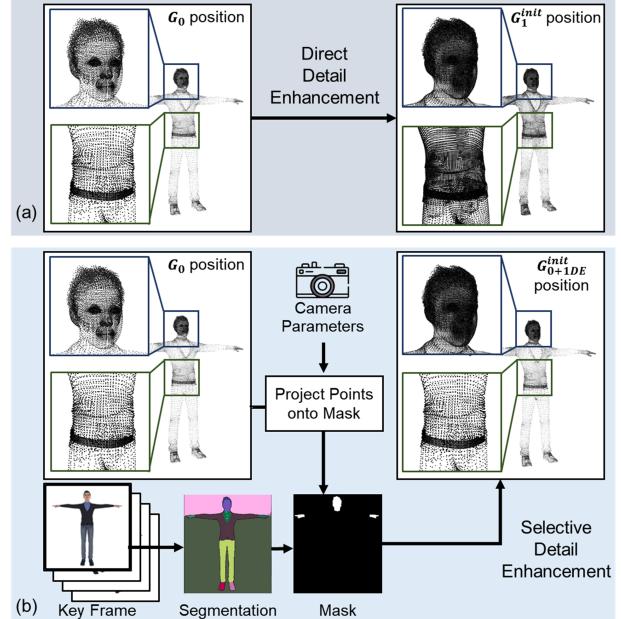


Figure 4: (a) Direct detail enhancement embeds new Gaussians on all triangle faces. (b) Selective detail enhancement embeds new Gaussians solely on the triangle faces selected by a mask, allowing for targeted enhancement in specific areas to regulate the number of Gaussians.

details essential for optimal performance in real-time interactive VR environments.

4 EVALUATION

Section 4 assesses our proposed methods through two experiments. Experiment 1 aims to determine the capability to produce high visual quality Gaussian avatars incorporating hierarchical embedding and selective detail enhancement methods. Experiment 2 investigates the frame rate fluctuations associated with varying numbers of Gaussian avatars when displayed statically and dynamically in real-time, to evaluate the computational costs incurred.

4.1 Gaussian Avatars and Baselines

In Section 4, our evaluation encompasses eight Gaussian avatars: (1) LoD1, (2) LoD2, (3) LoD3, (4) LoD1 with LoD2 detail enhancement (LoD1+2DE), (5) LoD1 with LoD3 detail enhancement (LoD1+3DE), (6) LoD2 with LoD3 detail enhancement (LoD2+3DE), (7) Gaussian Avatars, and (8) Splatting Avatars. Within each experiment, these avatars are derived from the same mesh (SMPL) and its corresponding textures as initial inputs. Specifically, the LoD1, LoD2, and LoD3 are formulated through the hierarchical embedding twice, thrice, and four times, respectively, as delineated in Sections 3.3 to 3.5. The absence of the LoD0 is due to the SMPL avatar featuring only 6890 vertices and 13776 faces. The LoD0 initialized with merely 20666 Gaussians, leading to visible transparent body parts during motion [59]. Expanding upon the LoD1 and LoD2, we establish the (4) LoD1+2DE, (5) LoD1+3DE, and (6) LoD2+3DE using the selective detail enhancement method described in Section 3.6.

Regarding the baselines, we chose Gaussian Avatars [9] and Splatting Avatars [10], which closely align with our method and yield top-quality generated avatars. In Gaussian Avatars, a Gaussian is initialized at the center of each triangle face in the mesh avatar and subsequently subdivided through adaptive density control. This method is initially applied to facial datasets, and we tested it on full-body avatars in this experiment. Splatting Avatars is the current state-

of-the-art method that initializes Gaussians randomly and embeds each Gaussian to the corresponding faces through the ‘walking on triangle’ during optimization. While these two methods share similarities with our embedding method, they initialize and optimize Gaussians on mesh avatars in a single step, unlike our hierarchical embedding method. By comparing these two methods with ours, we aim to assess the hierarchical embedding method. Additionally, comparing different LoDAvatars can validate the effects of selective detail enhancement.

4.2 Experiment 1: Visual Quality - Objective Assessment

4.2.1 Dataset

In objective assessments, we employed the People Snapshot Dataset [60] to train the 8 different avatars. This dataset comprises standard SMPL [51] avatars, along with corresponding textures, commonly utilized in previous studies for evaluating the quality of Gaussian avatars [12]. Specifically, we chose the female-3-casual and male-2-casual mesh avatars, characterized by intricate high-frequency and low-frequency details, along with their associated textures and key frame animations, as the dataset. In the data preparation phase, following the method described in Section 3.3, we positioned 42 virtual cameras within a spherical space with a radius of 2 meters around each of these avatars. We selected 40 key frames from the animations, capturing 40 sets of key frames with 42 different perspective images for each key frame. Each image was standardized to a resolution of 1080×1080 , with the corresponding camera parameters recorded. Subsequently, leveraging this dataset, we utilized this data to undergo processes such as Gaussian embedding, Gaussian optimization, and selective detail enhancement, creating 8 distinct Gaussian avatars.

4.2.2 Evaluation

Building upon evaluation from existing research [9], we employed two setups to assess the visual quality of different Gaussian avatars: (1) novel-view (employing key frame poses from training sequences to animate Gaussian avatars and rendering from novel viewpoints) and (2) reenactment (animating the avatars with different poses and rendering all 42 camera views). To assess the quality of the avatars generated, we employed established image similarity metrics *PSNR*, *SSIM*, and *LPIPS*. In this evaluation, we refrained from conducting assessments of avatars at distinct LoD across varying observation distances. This choice was deliberate, as we utilized a uniform white background during Gaussian optimization, rendering image similarity metrics unsuitable for evaluating visual quality across diverse observation distances.

4.2.3 Results

In the contexts of novel-view and reenactment, Tables 1 and 2 present the image similarity metrics for the two distinct models among the eight Gaussian avatars, we highlight the top three digits of each metric in bold. The objective assessment of image similarity reveals that when the Gaussian counts are limited and the avatar is at a lower LoD, the image similarity tends to be low due to the number of Gaussians is insufficient to represent all the details of the original mesh avatar. As the number of Gaussians increases, enabling a more comprehensive representation and increasing the avatar’s details, higher image similarity metrics are attained. Within the settings of novel-view and reenactment, the LoD2+3DE avatars and LoD3 avatars generated through our methods, demonstrate commendable image similarity metrics, nearly reaching or even surpassing the image similarity of the GA and SA groups. This observation suggests that via iterative hierarchical embedding, we can progressively generate Gaussian avatars with higher LoD from low-detail Gaussian avatars, thereby consistently enhancing the visual quality of the Gaussian avatars.

Table 1: Novel-View Image Similarity Metrics

Avatar	female-3-casual			male-2-casual		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
LoD1	14.87	0.55	0.31	13.96	0.51	0.33
LoD1+2DE	16.23	0.67	0.26	14.22	0.65	0.28
LoD1+3DE	18.58	0.72	0.24	16.93	0.73	0.22
LoD2	24.11	0.83	0.12	22.83	0.91	0.11
LoD2+3DE	28.61	0.91	0.06	27.74	0.93	0.08
LoD3	30.34	0.98	0.04	30.14	0.96	0.04
GA	27.88	0.92	0.06	25.29	0.89	0.10
SA	30.20	0.98	0.03	30.99	0.97	0.03

Table 2: Reenactment Image Similarity Metrics

Model	female-3-casual			male-2-casual		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
LoD1	11.12	0.42	0.38	10.98	0.48	0.37
LoD1+2DE	14.54	0.62	0.29	12.87	0.59	0.31
LoD1+3DE	14.76	0.69	0.26	14.82	0.62	0.25
LoD2	20.17	0.79	0.15	20.13	0.78	0.13
LoD2+3DE	22.48	0.88	0.08	23.17	0.84	0.12
LoD3	26.42	0.92	0.07	26.13	0.93	0.08
GA	22.42	0.87	0.10	23.47	0.86	0.11
SA	27.82	0.93	0.05	26.12	0.93	0.07

4.3 Experiment 1: Visual Quality - Subjective Evaluation

4.3.1 Dataset and Setup

In the subjective evaluation, we aim to assess participants’ subjective perception of avatars’ visual quality across varying viewing distances. Diverging from objective assessments, we did not utilize the People Snapshot dataset due to its limited texture resolution of 256×256 . In a VR context, where users engage with 3D avatars while immersed in a head-mounted display (HMD), higher-resolution textures are imperative for 3D avatars compared to evaluating 2D avatars on a screen [61]. This adjustment ensures that participants receive a more effective subjective visual quality evaluation. In the subjective evaluation, we leveraged the SMPL and SMPL-X [62] Unity project [63]. The SMPL-X Unity project offered textures with a resolution of 4096×4096 . We converted the UV maps and applied them to the SMPL model for our evaluation.

For various Gaussian avatars, during the data preparation phase, our camera configuration method remains consistent with the description in Section 4.2, entailing the placement of 42 cameras within a spherical space encompassing a two-meter radius, without setting cameras at different interaction distances. Consequently, with different observation distances, distinct Gaussian avatars share the same images and camera parameters as inputs, differing solely in their embedding and optimization methods. This method is motivated by two primary considerations: firstly, we maintain a similar setup method as in Section 4.2, which helps in better evaluating the visual quality of avatars through a combination of objective metrics and subjective evaluation; secondly, the eight avatars are all generated from the same mesh avatar and camera setup, with only differences in the Gaussian embedding, Gaussian optimization, and selective detail enhancement phases, facilitating a more effective evaluation of the proposed method.

During the subjective evaluation, we utilized the mesh avatar as the baseline. Participants were presented with a random order of eight Gaussian avatars and one mesh avatar at varying distances, and the subjective evaluation encompassed assessments for both static and dynamic avatars. In the static scenario, avatars maintained an

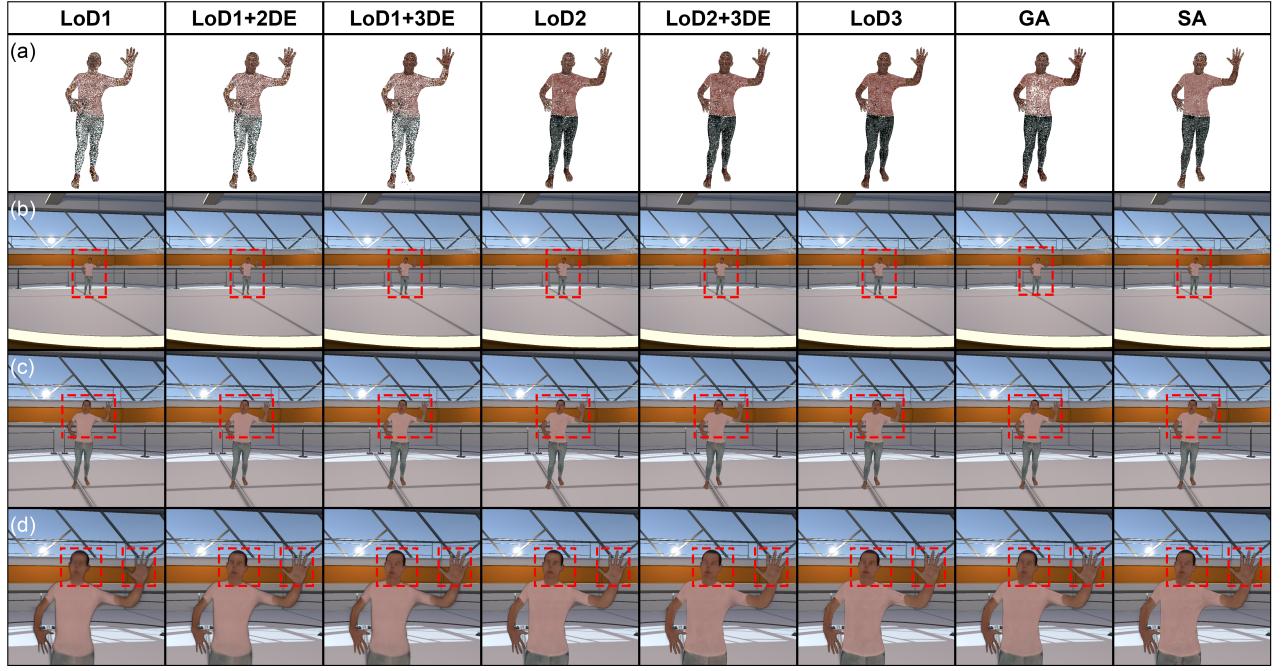


Figure 5: Gaussian avatars in the subjective evaluation. (a) The Gaussian point cloud positions. (b)(c)(d) Gaussian avatars and background observed at 1m, 5m and 9m distances.

initial T-pose; while in the dynamic scenario, avatars engaged in a 30-second animation sequence. The different Gaussian avatars were integrated into the Unity Gaussian Splatting project [64]. The Gaussian position point cloud maps corresponding to the eight Gaussian avatars and schematic diagrams illustrating observations at different distances are shown in Fig. 5.

4.3.2 Participants and Experiment Procedure

A total of 15 participants took part in the subjective evaluation, comprising 6 females and 9 males, aged between 22 and 36 years ($M = 27.2$). Eight of them had prior experience with VR devices. The testing was set up using Unity 2022.3.24f1. Each evaluation session featured only one type of avatar and background within the scene. The inclusion of the background aided participants in perceiving the current observation distance. Participants utilized Meta Quest 3 HMD and conducted the experiments while seated. Avatars appeared directly in front of participants at horizontal distances of 1m, 5m, and 9m, corresponding to near, medium, and far interaction distances in human-avatar interaction [65]. The eight Gaussian avatars and a baseline mesh avatar were presented to participants in a random order. Furthermore, the observation distances between participants and avatars were randomized, with different types of avatars appearing at varying observation distances. After observing each avatar for a minimum of 30 seconds, participants were tasked with rating the visual quality of the avatars they observed using a 7-point Likert Scale. Participants evaluated each avatar until they rated all static and dynamic avatars at different observation distances. Subsequently, all rating data were recorded for analysis.

4.3.3 Results

The subjective evaluations of the Gaussian avatars at various LoD and baselines when observed from different distances in both static and dynamic scenarios are depicted in Fig. 6. Each participant completed the entire evaluation, and the scores from 15 participants exhibited a normal distribution. The comparison of the subjective evaluations encompassed the following aspects:

The same LoDAvatar at different observation distances. The data was analyzed using the Repeated Measures ANOVA analysis.

In the static setting, significant differences were found for LoD1, LoD2 and LoD3 avatars ($F_{(2,42)} = 131.20, p < 0.001$). In the dynamic setting, significant differences were also found for LoD1, LoD2 and LoD3 avatars ($F_{(2,42)} = 192.96, p < 0.001$). The results suggest that high LoD avatars demonstrate consistent subjective visual quality across multiple observation distances, whereas low LoD avatars exhibit differences. Consequently, the avatar’s LoD can be dynamically adjusted based on the human-avatar distance to mitigate visual quality variations.

Different LoDAvatar at the same observation distances. The data was analyzed using paired t-tests. At a 1m observation distance, significant differences were observed between LoD1 and LoD2 (static(S) $t_{14} = -9.00, p < 0.001$; dynamic(D) $t_{14} = -14.00, p < 0.001$); significant differences were observed between LoD2 and LoD3 (S: $t_{14} = -7.25, p < 0.001$; D: $t_{14} = -6.59, p < 0.001$). At a 5m observation distance, significant differences were observed between LoD1 and LoD2 (S: $t_{14} = -6.58, p < 0.001$; D: $t_{14} = -11.52, p < 0.001$); significant differences were observed between LoD2 and LoD3 (S: $t_{14} = -3.61, p = 0.003$; D: $t_{14} = -6.87, p < 0.001$). At a 9m observation distance, significant differences were observed between LoD1 and LoD2 (S: $t_{14} = -4.29, p = 0.001$; D: $t_{14} = -9.03, p < 0.001$); significant differences were observed between LoD2 and LoD3 (S: $t_{14} = -2.26, p = 0.041$; D: $t_{14} = -4.52, p < 0.001$). As the human-avatar distance increases, significant differences between low LoD and high LoD avatars persist. However, the decrease in mean scores and t-values suggests a reduction in these discrepancies. This implies that with an increasing observation distance, low LoD Gaussian avatars can still maintain a certain level of visual quality and receive subjective evaluations that progressively approach those of high LoD avatars.

The subjective influence of selective detail enhancement. The data was analyzed using paired t-tests. At a 1m observation distance, significant differences were observed between LoD1 and LoD1+2DE (S: $t_{14} = -4.58, p < 0.001$; D: $t_{14} = -3.57, p = 0.003$); significant differences were also found between LoD1 and LoD1+3DE (S: $t_{14} = -7.90, p < 0.001$; D: $t_{14} = -10.64, p < 0.001$); there were significant differences between LoD2 and LoD2+3DE (S: $t_{14} = -5.28, p < 0.001$; D: $t_{14} = -8.57, p < 0.001$). At a 5m observation distance,

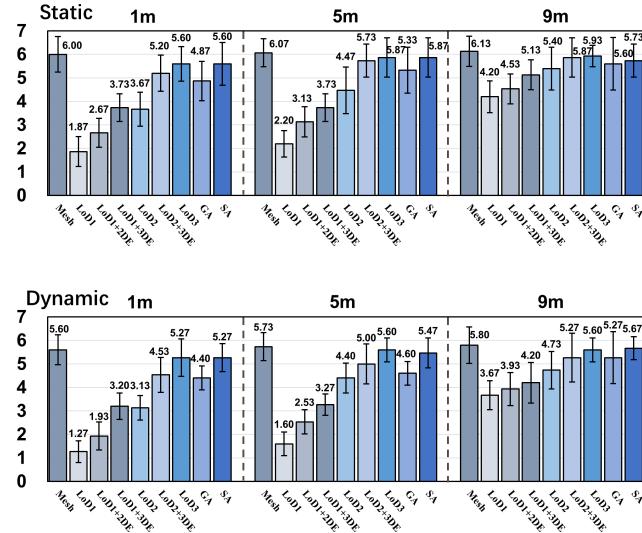


Figure 6: The subjective score of mesh avatar and different Gaussian avatars in static and dynamic settings. The error bar indicates the standard deviation.

significant differences were observed between LoD1 and LoD1+2DE (S: $t_{14} = -4.09, p = 0.001$; D: $t_{14} = -6.09, p < 0.001$); significant differences were also found between LoD1 and LoD1+3DE (S: $t_{14} = -9.28, p < 0.001$; D: $t_{14} = -7.17, p < 0.001$); significant differences were observed between LoD2 and LoD2+3DE (S: $t_{14} = -4.46, p = 0.001$; D: $t_{14} = -3.15, p = 0.007$). At a 9m observation distance, there were no significant differences between static LoD1 and LoD1+2DE ($t_{14} = -1.78, p = 0.096$), but significant were found in dynamic LoD1 and LoD1+2DE ($t_{14} = -2.26, p = 0.041$). there were significant differences between LoD1 and LoD1+3DE (S: $t_{14} = -4.09, p = 0.001$; D: $t_{14} = -3.23, p = 0.006$); significant differences were also observed between LoD2 and LoD2+3DE (S: $t_{14} = -2.17, p = 0.048$; D: $t_{14} = -4.00, p = 0.001$). Furthermore, at 1m distance, there were no significant differences between LoD2+3DE and LoD3 (S: $t_{14} = -1.47, p = 0.164$; D: $t_{14} = -2.05, p = 0.060$); at 5m distance, there were no significant differences between LoD2+3DE and LoD3 ($t_{14} = -0.69, p = 0.499$), but there was a significant difference dynamically ($t_{14} = -3.15, p = 0.007$); at 9m distance, there were no significant differences between LoD2+3DE and LoD3 (S: $t_{14} = -0.367, p = 0.719$; D: $t_{14} = -1.58, p = 0.136$). These results indicate that through facial and manual detail enhancement, avatars can enhance subjective visual quality evaluations while only slightly increasing the number of Gaussians, thus achieving a balance between visual quality and computational costs. These results highlight the importance of selective detail enhancement.

Comparison with baseline and existing methods. The LoD2+3DE received favorable evaluations in both static and dynamic at all distances. We conducted paired t-tests to compare LoD2+3DE with the baseline mesh avatar, GA, and SA. At a 1m observation distance, there were no significant differences between LoD2+3DE and GA (S: $t_{14} = 1.16, p = 0.265$; D: $t_{14} = 0.62, p = 0.546$); there were no significant differences between LoD2+3DE and SA (S: $t_{14} = -1.47, p = 0.164$; D: $t_{14} = -1.78, p = 0.096$); significant differences were observed between LoD2+3DE and the baseline (S: $t_{14} = -2.45, p = 0.028$; D: $t_{14} = -4.68, p < 0.001$). At a 5m observation distance, there were no significant differences between LoD2+3DE and GA in the static settings ($t_{14} = 1.57, p = 0.138$), but significance was found in the dynamic settings ($t_{14} = 2.45, p = 0.028$); no significant differences were found between LoD2+3DE and SA (S: $t_{14} = -0.52, p = 0.610$; D: $t_{14} = -1.71, p = 0.110$); significant differences were observed between LoD2+3DE and the baseline (S: $t_{14} = -2.45, p = 0.028$; D: $t_{14} = -3.56, p = 0.003$). At 9m

observation distance, there were no significant differences between LoD2+3DE and GA (S: $t_{14} = 0.94, p = 0.364$; D: $t_{14} = 0.69, p = 0.499$); there were no significant differences between LoD2+3DE and SA (S: $t_{14} = 0.435, p = 0.670$; D: $t_{14} = -1.19, p = 0.253$); no significant differences were found between LoD2+3DE and the baseline (S: $t_{14} = -1.00, p = 0.334$; D: $t_{14} = -1.95, p = 0.072$). The comparison results between the baseline and the LoD2+3DE suggest that although variances still exist between LoD2+3DE and the baseline at close and medium observation distances, the LoD2+3DE avatar can attain outcomes akin to the mesh avatar at far observation distances. Additionally, the LoD2+3DE avatar can produce visual quality comparable to the top existing Gaussian avatars while utilizing fewer Gaussians for avatar generation.

In summary, the subjective experiments have led to the following results: (1) Gaussian avatars at different LoD show differences in subjective visual quality evaluations. Nevertheless, with increasing observation distance, lower LoD Gaussian avatars can maintain some visual quality, approaching subjective evaluations similar to those of higher LoDAvatars. (2) The application of selective detail enhancement, which includes adding specific details to the face and hands, results in enhanced subjective evaluations.

4.4 Experiment 2: Computational Costs

4.4.1 Setup

Experiment 2 primarily assesses the average frame rates of LoDAvatars when operating within VR, to evaluate the computational costs. Within VR environments, various scenarios involve user interactions with virtual avatars. Real-time responsiveness in these interactions is paramount, necessitating reduced computational costs during rendering to ensure that avatars can sustain higher frame rates throughout VR engagements [66]. As described in Section 3.2, Gaussian splatting diverges from conventional rendering pipelines by splatting each Gaussian onto the camera’s image plane. Prior research has illustrated that when rendering the same virtual entities, avatars constructed based on Gaussians exhibit higher frame rates during runtime compared to mesh avatars [67], with computational costs directly correlated to the number of Gaussians [11]. In this experiment, we utilize the avatars and textures provided by the SMPL in Section 4.4 as input. Among the varied Gaussian avatars generated from this mesh avatar, the number of Gaussians is as follows: LoD1 consists of 75,770 Gaussians, LoD1+2DE consists of 96,470 Gaussians, LoD1+3DE consists of 179,270 Gaussians, LoD2 consists of 296,186 Gaussians, LoD2+3DE consists of 378,986 Gaussians, LoD3 consists of 1,177,850 Gaussians, GA consists of 182,391 Gaussians, and SA consists of 412,338 Gaussians.

4.4.2 Apparatus and Materials

We employed a computer equipped with an Intel Core i7 processor and an NVIDIA RTX 3080 graphics card for frame rate assessments. The testing environment was set up using Unity 2022.3.24f1, with no extraneous objects present in the scene, solely Gaussian avatars. Individual tests were conducted with one, two, and three avatars, encompassing static and dynamic avatar scenarios. Each scene ran for one minute, during which we recorded the average frame rates exhibited by these avatars within Unity.

4.4.3 Results

The results depicted in Fig. 7 illustrate the fluctuation in running frame rates of different Gaussian avatars, both static and dynamic, as the number of avatars increases. Increasing the avatar’s LoD will also increase the computational costs as the number of Gaussians increases. Notably, for dynamic avatars, the frame rates during rendering lag behind those of static avatars. Furthermore, as the number of Gaussians rises, the decline in frame rates becomes more pronounced, highlighting the challenge posed by increased computational costs. While LoD3 avatars and SA showcased superior

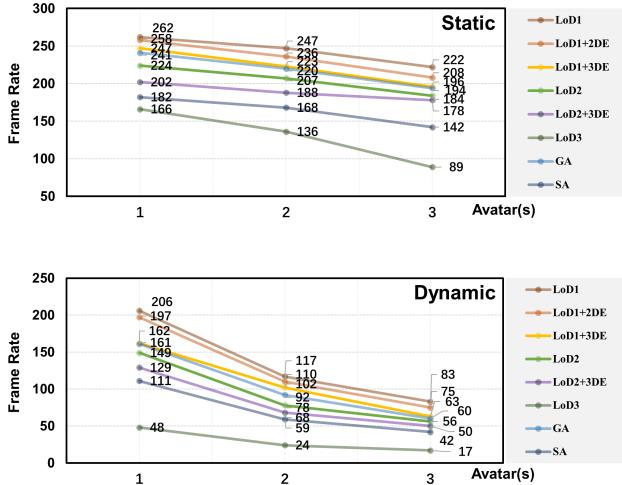


Figure 7: The average frame rate variation with the number of avatars for different Gaussian avatars in static and dynamic settings.

visual quality in Experiment 1, these avatars exhibited higher computational costs. Conversely, avatars with lower detail levels were demonstrated to achieve higher frame rates during rendering. Moreover, employing the selective detail enhancement method to enhance the head and hands of avatars with more interactive functionalities can further control the number of Gaussians, thereby reducing computational costs to a certain extent.

In summary, as the LoD of avatars increases and the quantity of avatars increases, a higher number of Gaussians results in increased computational costs. Thus, we advocate for the integration of LoD into Gaussian avatars to align with real-time operational requisites for observing or engaging with avatars in VR. Designers can use Gaussian avatars with different LoD while maintaining visual quality in VR, which can help reduce rendering computational costs.

5 DISCUSSION

5.1 Further Analysis

As a novel rendering technique, Gaussian Splatting presents advantages in both high visual quality and low computational costs. The drivable 3D Gaussian avatar method has gained significant attention due to the need for real-time interactions with avatars in VR. This method inputs a base mesh avatar and transforms the 3D Gaussians' world position, rotation, and scaling into local coordinates on triangle faces. The embedded 3D Gaussians move correspondingly with mesh changes, enabling dynamic Gaussian avatars.

Prior research has primarily concentrated on implementing drivable Gaussian avatars without extensively addressing the balance between visual quality and computational costs in dynamic Gaussian avatars. As the number of Gaussians increases, avatars exhibit enhanced visual quality but incur higher computational costs, resulting in a trade-off. We present LoDAvatar, which introduces levels of detail into Gaussian avatars to better leverage the advantages of Gaussian Splatting, which are high visual quality and low computational costs. The core methods of LoDAvatar involve hierarchical embedding and selective detail enhancement. Hierarchical embedding generates various LoD Gaussian avatars from low to high detail levels while controlling the number of Gaussians, yielding a range of avatars with diverse LoDs. Selective detail enhancement reinforces specific areas of avatars with strong interactive attributes, such as the head and hands, further controlling the number of Gaussians to reduce computational costs while minimizing the impact on visual quality. Employing these methods, we created avatars with varying LoD starting from a mesh avatar, and assessed the visual quality of these avatars alongside existing methods through objective and

subjective evaluations. Additionally, we conducted runtime frame rate tests on different LoD avatars and existing methods to evaluate computational costs during rendering. In terms of visual quality, objective and subjective evaluations indicate that our proposed hierarchical embedding and selective detail enhancement methods can increase the number of Gaussians, resulting in the generation of diverse LoD avatars. The high LoD avatars can achieve a visual quality comparable to that of established high-quality Gaussian avatar generation methods. With an increase in the distance between participants and avatars, the subjective evaluation gap between low LoD and high LoD avatars decreases, indicating the feasibility of incorporating LoD concepts in Gaussian avatars. Furthermore, the frame rate tests for various Gaussian avatars in Experiment 2 reveal that the computational costs of rendering Gaussian avatars are linked to the number of Gaussians, with a more pronounced frame rate decrease as the number of Gaussians rises, especially noticeable in dynamic avatars. Additionally, the outcomes of the static avatar experiments suggest the potential for generating different LoD Gaussian objects and scenes. Our proposed method holds promise for future deployment of Gaussian avatars on mobile devices, suitable for real-time rendering of multiple Gaussian avatars or extensive Gaussian scenes.

Based on the outcomes of the two experiments, our hierarchical embedding and selective detail enhancement methods contribute to achieving Gaussian avatars with LoD. We advocate for the adoption of LoDAvatar in dynamic Gaussian avatars to strike a balance between visual quality and computational costs.

5.2 Limitations and Future Work

In this study, we introduced LoD for dynamic Gaussian avatars using hierarchical embedding and selective detail enhancement methods. However, our current work has certain limitations that suggest promising avenues for future research.

Our method utilizes a mesh avatar as input to generate LoDAvatars, rather than employing real images. This decision stems from the lack of topological consistency in mesh avatars generated from real images, necessitating re-topologizing each key frame mesh for Gaussian embedding. When generating LoDAvatars from real-world images, it is necessary to first fit them to a standard SMPL model with multi-view images. Future research could explore the automated creation of LoDAvatars directly from real-world images to enhance the efficient generation. Furthermore, we implemented LoD in Gaussian avatars through hierarchical embedding without initially generating distinct LoD mesh avatars for embedding. This is due to the intensive workload involved in texture resetting and skeleton re-binding on mesh avatars. Conversely, the hierarchical embedding and selective detail enhancement methods rely primarily on code execution. Future research could investigate the visual quality disparities using Gaussian embedding on various LoD mesh avatars and hierarchical Gaussian embedding.

In our experiments, we utilized the SMPL model as the base mesh avatar, with LoD ranging from LoD1 to LoD3. Subsequent research could explore different iterations to produce high-quality Gaussian avatars from mesh avatars with varying numbers of vertices and faces to exert precise control over the number of Gaussians.

Within LoDAvatar, all Gaussians are embedded in local triangle face coordinate systems, with all Gaussians moving with the mesh. Future research could explore driving different components of Gaussian avatars, such as achieving more realistic movement of hair and clothing during avatar driving.

In LoDAvatar, our emphasis is solely on implementing LoD on Gaussian avatars by adjusting the position, rotation, and scale of Gaussians. Subsequent research could delve into modifying color parameters and transparency on dynamic Gaussian avatars to create color-variable LoDAvatar for consistent lighting between avatars and backgrounds.

6 CONCLUSION

In this paper, we introduce LoDAvatar, a method utilizing hierarchical embedding and selective detail enhancement to generate Gaussian avatars with different LoD. Our method takes existing mesh avatars as input, involving data preparation, Gaussian embedding, Gaussian optimization, and selective detail enhancement. We conduct two experiments to evaluate our proposed methods. Experiment 1 assesses the visual quality through both objective and subjective evaluations, demonstrating that the hierarchical embedding and selective detail enhancement methods can produce LoDAvatars with commendable visual quality. In Experiment 2, we examine the average frame rates of LoDAvatars during runtime to analyze computational costs, further emphasizing the importance of integrating LoD in Gaussian avatars. LoDAvatar showcases the potential to reduce the computational costs required for rendering when observing avatars from a distance. We suggest that the hierarchical embedding and selective detail enhancement methods can be effectively employed for LoD generation in dynamic Gaussian avatars, striking a balance between visual quality and computational efficiency.

REFERENCES

- [1] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15893–15903, 2022.
- [2] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16922–16932, 2023.
- [3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [4] Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 3d gaussian as a new vision era: A survey. *arXiv preprint arXiv:2402.07181*, 2024.
- [5] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.
- [6] Guikun Chen and Wenguang Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024.
- [7] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.
- [8] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. *arXiv preprint arXiv:2311.08581*, 2023.
- [9] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024.
- [10] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1606–1616, 2024.
- [11] Yang Liu, He Guan, Chuanchen Luo, Lue Fan, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. *arXiv preprint arXiv:2404.01133*, 2024.
- [12] 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. *Computational Visual Media*, pages 1–30, 2024.
- [13] Olga Sorkine and Daniel Cohen-Or. Least-squares meshes. In *Proceedings Shape Modeling Applications, 2004.*, pages 191–199. IEEE, 2004.
- [14] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [15] Yongning Zhu and Robert Bridson. Animating sand as a fluid. *ACM Transactions on Graphics (TOG)*, 24(3):965–972, 2005.
- [16] 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. *Communications of the ACM*, 65(1):99–106, 2021.
- [17] Anurag Dalal, Daniel Hagen, Kjell G Robbersmyr, and Kristian Muri Knausgård. Gaussian splatting: 3d reconstruction and novel view synthesis, a review. *IEEE Access*, 2024.
- [18] Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. An efficient 3d gaussian representation for monocular/multi-view dynamic scenes. *arXiv preprint arXiv:2311.12897*, 2023.
- [19] Xinhai Li, Huaibin Wang, and Kuo-Kun Tseng. Gaussiandiffusion: 3d gaussian splatting for denoising diffusion probabilistic models with structured noise. *arXiv preprint arXiv:2311.11221*, 2023.
- [20] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20923–20931, 2024.
- [21] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024.
- [22] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024.
- [23] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv preprint arXiv:2311.16043*, 2023.
- [24] Shoukang Hu, Tao Hu, and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20418–20431, 2024.
- [25] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6657, 2024.
- [26] Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: Impressive head avatars with learnable gaussian diffusion. *arXiv preprint arXiv:2312.01632*, 2023.
- [27] Panwang Pan, Zhuo Su, Chenguo Lin, Zhen Fan, Yongjie Zhang, Zeming Li, Tingting Shen, Yadong Mu, and Yebin Liu. Humansplat: Generalizable single-image human gaussian splatting with structure priors. *arXiv preprint arXiv:2406.12459*, 2024.
- [28] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2024.
- [29] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19680–19690, 2024.
- [30] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eun-byung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21719–21728, 2024.
- [31] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023.
- [32] Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. Hifi4g: High-fidelity human

- performance rendering via compact gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19734–19745, 2024.
- [33] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.
- [34] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024.
- [35] Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. Gaussianhair: Hair modeling and rendering with light-aware gaussians. *arXiv preprint arXiv:2402.10483*, 2024.
- [36] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024.
- [37] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 505–515, 2024.
- [38] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 634–644, 2024.
- [39] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: a physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024.
- [40] David Luebke, Martin Reddy, Jonathan D Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. *Level of detail for 3D graphics*. Elsevier, 2002.
- [41] Filip Biljecki, Hugo Ledoux, Jantien Stoter, and Junqiao Zhao. Formalisation of the level of detail in 3d city modelling. *Computers, environment and urban systems*, 48:1–15, 2014.
- [42] Jimmy Abualdenien and André Borrman. Levels of detail, development, definition, and information need: a critical literature review. *Journal of Information Technology in Construction*, 27, 2022.
- [43] James H Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, 1976.
- [44] Tan Kim Heok and Daut Daman. A review on level of detail. In *Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004.*, pages 70–75. IEEE, 2004.
- [45] Yuanbo Xiangli, Lining Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision*, pages 106–122. Springer, 2022.
- [46] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multi-scale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.
- [47] Yiyu Zhuang, Qi Zhang, Ying Feng, Hao Zhu, Yao Yao, Xiaoyu Li, Yan-Pei Cao, Ying Shan, and Xun Cao. Anti-aliased neural implicit surfaces with encoding level of detail. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023.
- [48] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021.
- [49] Tobias Fischer, Jonas Kulhanek, Samuel Rota Bulò, Lorenzo Porzi, Marc Pollefeys, and Peter Kortschieder. Dynamic 3d gaussian fields for urban areas. *arXiv preprint arXiv:2406.03175*, 2024.
- [50] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *arXiv preprint arXiv:2311.17245*, 2023.
- [51] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.
- [52] D. Bojanic. Smpl-fitting. <https://github.com/DavidBoja/SMPL-Fitting>, 2024.
- [53] Mixamo. <https://www.mixamo.com/>. Accessed: July 31, 2024.
- [54] Haejung Suk and Teemu H Laine. Influence of avatar facial appearance on users’ perceived embodiment and presence in immersive virtual reality. *Electronics*, 12(3):583, 2023.
- [55] Marc Erich Latoschik, Daniel Roth, Dominik Gall, Jascha Achenbach, Thomas Waltemate, and Mario Botsch. The effect of avatar realism in immersive social virtual realities. In *Proceedings of the 23rd ACM symposium on virtual reality software and technology*, pages 1–10, 2017.
- [56] Haiyan Jiang, Dongdong Weng, Zhen Song, Xiaonuo Dongye, and Zhenliang Zhang. Dexhand: dexterous hand manipulation motion synthesis for virtual reality. *Virtual Reality*, 27(3):2341–2356, 2023.
- [57] Daniel Roth, Jean-Luc Lugrin, Dmitri Galakhov, Arvid Hofmann, Gary Bente, Marc Erich Latoschik, and Arnulf Fuhrmann. Avatar realism and social interaction quality in virtual reality. In *2016 IEEE virtual reality (VR)*, pages 277–278. IEEE, 2016.
- [58] Alexander Kirillov, Eric Mintun, Nikhil Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [59] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.
- [60] Thiendo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8387–8397, Jun 2018. CVPR Spotlight Paper.
- [61] Mel Slater. Immersion and the illusion of presence in virtual reality. *British journal of psychology*, 109(3):431–433, 2018.
- [62] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.
- [63] Smpl unity. <https://smpl.is.tue.mpg.de/download.php/>. Accessed: July 31, 2024.
- [64] Aras Pranckevičius. Unity gaussian splatting. <https://github.com/aras-p/UnityGaussianSplatting>, 2024.
- [65] Shane L Rogers, Rebecca Broadbent, Jemma Brown, Alan Fraser, and Craig P Speelman. Realistic motion avatars are the future for social interaction in virtual reality. *Frontiers in Virtual Reality*, 2:750729, 2022.
- [66] Xiaonuo Dongye, Dongdong Weng, Haiyan Jiang, and Pukun Chen. Learning personalized agent for real-time face-to-face interaction in vr. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 759–760, 2024.
- [67] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *arXiv preprint arXiv:2407.17418*, 2024.