Alma Mater Studiorum · Università di Bologna

SCHOOL OF SCIENCE

Department of Computer Science - Science and Engineering - DISI

Pose Signal Filtering for Enhanced 3D Gaussian Splatting Animation

SUPERVISOR:
Prof. Gustavo Marfia

Presented by:
Serina Mato

CO-SUPERVISOR:
Pasquale Cascarano
Jacopo Meglioraldi

Session 3rd

Academic Year 2024/2025

Abstract

This work presents a structured pipeline that leverages Gaussian splatting for precise 3D model reconstruction, utilizes the OpenPose library and the ROMP model for motion animation, and applies signal smoothing techniques to enhance positional consistency. The estrapolation techniques provide the movements of the joint motion data that drive the animation, often resulting in unstable motion trajectories, causing jitter, noise, and abrupt transitions. The proposed approach addresses these challenges by refining the position signal, ensuring smoother and more natural movement.

To achieve this, the system applies motion-aware interpolation to minimize fluctuations in positional data. Using the moving average window filtering the pipeline secures a balance between responsiveness and fluid motion, as mentioned here.

This method significantly improves the motion fluidity of animated Gaussian splatting models, making them ideal for real-time applications, digital avatars, and interactive experiences. Although the smoothing process requires additional computational resources, it effectively reduces motion artifacts, resulting in more stable, realistic, and visually seamless animations.

List of Abbreviations

ROMP Robust Multi-Person Pose Estimation
CAD Computer-Aided Design
DCC Digital Content Creation
VR Virtual Reality
VFX Visual effects
AR Augmented Reality
SDF Signed Distance Function
CMM Coordinate Measurement Machine
CT Computed tomography
TOF Time-of-Flight
SPAD Signal Pass At Danger
RGB Red Green Blue
RGP-D Red Green Blue-Depth
MSHIF Multi-source heterogeneous information fusion
GPS Global Positioning System
IMU Inertial Measurement Unit
V2X Vehicle-to-Everything
SFM Structure from Motion
SIFT Stop Investigate the Source
RANSAC Random-Sample Consensus
SMPL/X Skinned Multi-Person Linear Model eXpressive
SLAM Simultaneous Location and Mapping
GS-SLAM Simultaneous Localization and Mapping with 3D Gaussian Splatting
3D-GS 3D Gaussian Splatting
AGI General Artificial Intelligence
TPV Total Payment Volume
BEV Bird's Eye View
GAN Generative Adversarial Network
NeRF Neural Radiance Field
MLP Multi-Layer Perceptron
IMUs Inertial Measurement Units
SH Spherical Harmonics
SuGaR Surface-aligned Gaussian Reconstruction.
SPM Sparse Parametric Model

ROMP Regression-based Object Motion Prediction

Contents

List of Figures

Introduzione

The digital representation and reconstruction of 3D scenes in computers serve as the foundation for numerous critical applications today. In many cases, 3D reconstruction technology offers a non-invasive alternative to replicating valuable or fragile cultural artifacts, avoiding the potential damage caused by traditional plaster casting techniques. It plays a vital role in the preservation of historical relics and cultural heritage. In the gaming and film industries, dynamic 3D scene reconstruction enhances real-time rendering, improving the visual experience in games and movies. Medical imaging facilitates the creation of patient-specific organ models for surgical planning. For robot navigation, it enables robots to better understand their surroundings, improving both navigation accuracy and safety.

In industrial design, 3D reconstruction assists in generating precise digital models by capturing the geometric details of real objects, helping users analyze dynamic data changes. By capturing a user's body shape, preferences, or needs, designers can create personalized products. In addition, it supports the documentation of equipment and mechanical parts, providing a digital reference for maintenance. 3D avatars play an important role in 3D reconstruction, especially in applications that require realistic human modeling and interaction within digital environments. In recent years, advances in the fields of virtual reality, 3D animation, and digital interactions have led to the creation of increasingly realistic and interactive avatars.

In the following, we examine in detail: 3D avatars play an important role in e-mental Health Interventions. They help in exploring new models of client-therapist interaction. Here, some applications identified (1) in the formation of online peer support communities; (2) replicating traditional modes of psychotherapy by using avatars as a vehicle to communicate within a wholly virtual environment; (3) using avatar technology to facilitate or augment face-to-face treatment; (4) as part of serious games; and (5) communication with an autonomous virtual therapist. [article6 44] The Gaussian splitting is a novel technique, first introduced in 2023. So, there are not many papers and reports about this new technology, researchers are still exploring in depth this new technique.

When it comes to reconstructing 3D avatars or other complex human models, Gaussian

splatting has a few limitations.
It might be difficult to capture intricate details of human anatomy, such as small features in the face, hands, or clothing. Although it works well for smoother surfaces or general scene reconstruction, it may lack the ability to accurately represent highly detailed geometry.

Gaussian splitting shows some challenges with Rigging and Animation: because of its non-rigid transformations, although it can represent static scenes or objects, it becomes tricky to animate complex non-rigid bodies like human avatars. Gaussian splatting does not natively support the types of deformation (e.g. muscle movement, facial expressions) that are essential for creating realistic animations. Moreover, since Gaussian Splatting does not inherently involve vertices or meshes (which are typically used for rigging in animation), it is not easy to integrate with conventional animation tools designed for skeleton-based rigging. So, first, the build pipeline converts frames from a video into 3D Gaussian primitives to reconstruct 3D avatars. Then, it extracts key frames from the video, which represent different views of the avatar. Each pixel in these frames corresponds to a point in 3D space, which is represented as a Gaussian with properties such as position, color, and transparency. These Gaussians are projected from multiple camera angles to form a comprehensive 3D model. Through iterative optimization, the system aligns the Gaussians with the desired views, refining the model to accurately represent the avatar's geometry and appearance. During animating the 3D avatar, there were obvious difficulties with animation. Gaussian splitting has non-rigid deformations, so it struggles with non-rigid body movements like muscle flexing, facial expressions, and clothing movement, which are crucial for realistic avatar animation. Without mesh structures, it is difficult to control and apply these deformations effectively. Since Gaussian Splatting uses 3D Gaussian primitives, it is challenging to integrate into established animation pipelines, such as those used for motion capture or facial animation.
Due to all the above reasons, the 3D avatar animations were a little noisy, it oscillated during movement. So, as an improvement, this work proposes an improvement of the position signal, resulting in a smoother and more natural motion. This is achieved by using temporal filtering and motion-sensitive interpolation to reduce fluctuations in positional data.

Chapter 1

Research Objectives

Creating animated avatars that faithfully reflect users' appearance and movements is a technical challenge that requires the integration of various disciplines, including computer graphics, artificial intelligence, and computer vision.

Among the many techniques, which are explained in detail in **Chapter 2**, which are used to create realistic avatars, "Gaussian Splatting" has gained increasing attention due to its ability to generate photorealistic 3D models with superior visual quality, while maintaining efficient computational resource management.

This thesis explores 3D animation, especially the 3D reconstructed avatar, and the advancement on how to improve the animation motion of the 3D avatar, reducing the time complexity in animation rendering. Using essentially 2 filters, the low-pass band and the moving average window on the animation spectrum, the aim is to efficiently improve the pose signal filtered to render high-quality animations. The proposed method improves performance while maintaining visual fidelity.

In this context, the technique "Gaussian Splatting" that is approached offers an innovative approach to represent 3D surfaces through Gaussian points, allowing for accuracy in texture rendering and fluidity in animated movements. It provides sharper details and better handling of fine textures compared to other representations. Unlike point clouds, this technique utilizes point-based rendering with Gaussian distributions to create smooth blending, avoiding aliasing or jagged edges, and detailed 3D reconstructions.

Applying pose signal filtering to 3D Gaussian splatting animation enhances the stability, realism, and overall quality of animations. It makes for a smoother and more natural motion because, first of all, it reduces noise, helping to eliminate unwanted fluctuations. The 3D avatar studied in this work, after filtering the signal, had a more natural and smooth motion with more fluid and visually appealing movements. Another achievement is that filtering techniques, such as moving average filters, can be efficiently implemented for real-time applications without requiring extensive computational resources. This will

be explained in more detail in **Chapter 4.**

So, in general, the difference between the motion before and after the 3D avatar was quite evident. These avatars, which we worked on mostly as symbols of the 3D animation, had more smooth and natural motion.

Chapter 3 will explain in more detail this real-time 3D scene representation technique.

1.1 Thesis Organization

The organization of the work carried out follows the following structure:

CHAPTER 2 - **Related work** Overview of 3D reconstruction techniques.

CHAPTER 3 - **Methodology** Overview of the steps followed to build the pipeline

CHAPTER 4 - **Experimental setups** here will be explained, some of the experiments conducted to prove that the pipeline works.

CHAPTER 5 - **Conclusion and Future Works** In the final chapter, the conclusions of this study and future works to improve the pipeline.

Chapter 2

Related Work

Based on how the 3D shape and structure are represented and stored in digital format, 3D reconstruction is divided into explicit and implicit expression methods. Explicit expression refers to a method that precisely defines geometric shapes and structures, directly describing an object's external or internal geometry. This approach relies on discrete data, which inherently results in some loss of information, requiring the development of improved processing techniques. In addition, generating images from multiple viewpoints involves a considerable computational cost.[s24072314 47]

In contrast, the implicit expression represents the geometry of an object using a function rather than explicitly defining its shape. Instead of storing geometric details directly, this method encodes the shape through an implicit function or surface equation, which is then used to compute the geometry. By evaluating the function, specific values corresponding to points on the object's surface can be obtained. Here, explained in detail: [s24072314 47]

- **Explicit Expression:** The main methods for displaying data include point clouds, voxels, and meshes. Point clouds consist of discrete data collected from various sensors or scanning devices. It is used to represent the external surface of an object or the spatial structure of a scene. A point cloud is an unordered collection of points in a 3D space. Divide the 3D space into uniform cubic units. Each cubic unit is called a voxel. Each voxel can contain information that represents spatial attributes, such as color, density, or depth. Voxels are commonly used in medical image processing, computational fluid dynamics, and other fields. Voxel storage is used to represent the structure and attributes within a space, but it has high space complexity. The mesh is composed of connected vertices, edges, and faces. The mesh model can be composed of triangles, quadrilaterals, or higher-order polygons and can describe most topological structures. It can accurately represent complex geometric shapes and details. The surface described by each

triangle is planar, which makes it suitable for numerous computer graphics and engineering applications where triangle meshes are commonly used. This ensures that the projection is always convex and easy to rasterize.

- **Implicit Expression:** It does not require explicit storage of geometric data; therefore, it offers advantages in saving storage space and processing complex geometries. However, computing the value of an implicit function can be time consuming, and understanding and manipulating the implicit expression can be challenging. Implicitly represented 3D models can be determined by continuous decision boundaries, enabling shape recovery at any resolution. Commonly used implicit representations include implicit surfaces, Signed Distance Function (SDF), Occupancy Field, Radiance Field, etc.

During 3D reconstruction, factors to pay attention to are the varying nature of the scenes being reconstructed, the desired accuracy, and the technological advancements available.

Let us take the example of an archaeologist working to preserve the intricate details of an ancient temple. The temple is a fixed structure, and no movement or changes occur over time. The goal of the reconstruction is to capture its geometry, textures and fine details for preservation, study, and possible restoration. The methods work by taking multiple photographs of the temple from different angles, allowing the algorithms to analyze and reconstruct a highly detailed 3D model of the building. Since the temple is static and does not change over time, the reconstruction is focused purely on the geometry and textures, without needing to account for any movement or dynamic changes in the environment.

Mostly, the work is done on unstructured, incomplete data (such as images), which causes noisy work. Various solutions grouped into 3 main categories have been explored: **Traditional static 3D Reconstruction Methods**, these methods are designed to reconstruct objects or environments that do not change over time, by using multiple images or viewpoints of an object or environment to generate a highly detailed 3D model; **Dynamic 3D Reconstruction Methods**, which is designed to handle scenes and objects that change over time where the environment or objects are in motion, such as robotics, animation, motion capture, and augmented reality; **3D Reconstruction Methods Based on Machine Learning** are techniques that use machine learning algorithms, particularly deep learning models, to reconstruct three-dimensional (3D) models or structures from two-dimensional (2D) data. They employ large data sets and sophisticated neural networks to infer 3D information from images, video, or point clouds, enabling more accurate and efficient 3D model generation.

In the following, each method will be explained in more detail, emphasizing their main concepts, advantages, and limitations.

2.1 Traditional Static 3D Reconstruction Methods

Most creatures in nature, including humans, rely on vision to perceive and reconstruct. 3D objects in the physical world. 3D reconstruction can be classified into sparse reconstruction and dense reconstruction based on the density of information acquired. Sparse reconstruction focuses on obtaining the accurate 3D positions of a small number of key points or feature points in the scene. Using techniques such as feature point matching and key point extraction, to do this, it uses techniques to represent the geometric shape of the entire scene through these discrete points. The dense reconstruction aims to obtain the accurate 3D coordinates of each pixel in the scene. By estimating the depth of each pixel in the image, the system generates a dense depth map, point cloud, or voxel, allowing high-density reconstruction of the entire scene. Develop a model to create a comprehensive description of the entire scene. The contact method uses specific instruments to quickly and directly measure the 3D information of the scene, which mainly includes trigger measurement and continuous measurement. The contact method can only be used in situations where the instrument can come into contact with the measurement scene, such as coordinate measuring machines (CMMs), etc. The noncontact method utilizes image analysis models to acquire data from the measured object without physically touching it. The noncontact 3D reconstruction process involves capturing an image sequence using visual sensors (one or more cameras). Subsequently, relevant information is extracted and, finally, reverse engineering modeling is conducted using this information to reconstruct the 3D structural model of the object. [47] The non-contact methods are divided into 2 categories: active and passive.

s24072314

2.1.1 Active 3D Reconstruction Methods

Active methods of vision-based 3D reconstruction involve mechanical or radiometric interference with the reconstructed object to acquire depth maps. These methods include structured light, laser rangefinders, and other active sensing technologies. Among them, 3D reconstruction technologies based on active methods mainly include the laser scanning method, industrial computed tomography (CT) scanning, structured light method, time-of-flight (TOF) technology, shadow method, etc. These methods primarily utilize optical instruments to scan the surface of an object and reconstruct the 3D structure by analyzing the scanned data.

Light Detection and Ranging

LiDAR is a laser-based sensing technology that can reconstruct the distances of the object points from the emitting station. It is composed of an active sensor that emits laser waves and a passive sensor that reconstructs distances based on the travel time of the laser beam. Since mechanical laser scanners, were introduced in the 1990s, LiDAR technology has been widely studied to increase the accuracy, reliability, and portability of the technique. Depending on the application, LiDAR technology differs greatly in complexity, cost, and capabilities [3]. article7 Compact time-of-flight (ToF) cameras are used in robot navigation due to their ability to measure the entire scene simultaneously and multiple times per second [27]. article8 Aerial LiDAR is used to map large areas with high-distance laser accuracy, allowing the creation of digital elevation models for geographical information systems [37]. article9 Today, low-energy and fast LiDAR sensors using a single-photon avalanche diode (SPAD) can be found in consumer smartphones, for common user applications such as spatial measurements and improved media recording features. To measure distance, the sensor records the energy of the returning signal from the reflected laser by specialized electric components. Most techniques use the Time of Flight(ToF) to retrieve the distance of surfaces from the emitter. Multiple data are then combined to obtain depth maps and point clouds. Depending on the LiDAR technology used, different properties of the scanned material can be extracted. In particular, lasers are highly susceptible to the reflective properties of the material. Unlike structured light technology, high reflective surfaces mean a clearer reflected signal, allowing for precise measuring even from long distances. The depth information of a scene is usually paired with standard RGB images by mapping the two sensory information into a unified RGB-D image. The cost of acquisition can vary depending on desired accuracy and application, from high-precision high-distance LiDAR sensors mounted on satellites to cheap low-resolution setups mounted in pairs with high-resolution cameras on smartphones. LiDAR sensors are widely distributed in dynamic scene reconstruction applications where a high sampling rate is required for fast-moving objects and with a wide variety of objects with complex material properties appearing in the scene. LiDAR sensors are also used to augment the standard photogrammetry setup, providing additional depth information and easier triangulation at a minimal cost.

Here are some of the advantages and limitations of this technology: LIDAR technology offers several advantages, particularly in gathering terrain data in areas with dense vegetation. LIDAR can accurately capture terrain beneath dense foliage. It is also highly precise, supported by additional sensors like IMUs (Inertial Measurement Units) that track speed, orientation, and gravitational forces. Moreover, the point cloud is generated more quickly than with photogrammetry, requiring less post-processing computational

power.

However, a significant drawback of LIDAR is its high cost of equipment, installation, and setup. Recently, more affordable LIDAR sensors, such as the Livox L1 for unmanned vehicles, have made significant advancements in geospatial data collection.[s101110356 15]

Structured Light

Structured light reconstruction is a technique that uses active light projection of known patterns (usually a grid, parallel stripes, or a dot matrix) over the object to reconstruct its surface based on the deformation of the light spots. Knowing the position of the light source and the position of the sensing camera, a point can be triangulated in the 3D space with the standard camera analytical computation. The setup is visually reported in Figure 5.1. Structured light scanners are standard in 3D reconstruction for industrial settings: Surface reconstruction scanners are integrated with production lines to inspect manufacturing errors [article10 43], portable hand-held scanners are used for custom scans of objects of different sizes, and multicamera setups enhance classical photogrammetry with the additional precision of structured light triangulation [Alton2017Experience6F 2]. This technique is a fast and accurate method due to point triangulation within single images, but it presents some limitations inherent to the material properties of the object scanned. Reflective surfaces, semitransparent surfaces, and bad light conditions distort the projected light, introducing a not negligible error in the reconstruction. The acquisition phase in a controlled environment, as well as the application of opaque coatings to the material prior to scanning [article11 22], drastically reduces the limitations of this technique, making it flexible and reliable. Structured light cameras are still expensive, requiring a calibrated camera and a projector with high manufacturing accuracy, and not affordable for mass users. The procedure of scanning large objects is time-consuming for a single handheld camera sensor, which can take hours to correctly scan all parts of the object while maintaining good overlap between frames for correct positional tracking of the camera.

Here are some of the advantages and limitations of this technology:
Structured light technology is praised for its accuracy and speed. However, like any technology, it has limitations, particularly in terms of the size of objects it can effectively scan.

Photometric Stereo

It utilizes variations in illumination angles from multiple light sources, and it deduces the normality and depth of the surface by analyzing the changes in brightness on the object. It is suitable for objects with complex topological structures, but is sensitive to

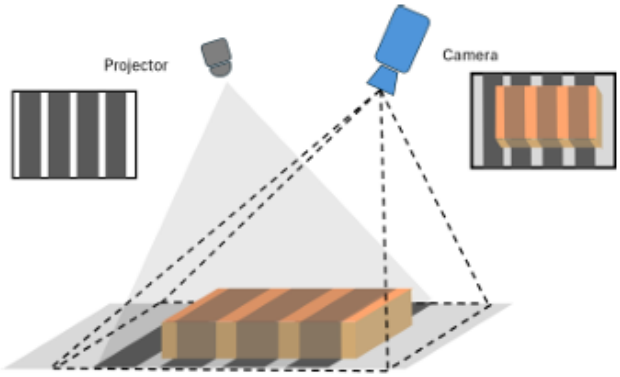

Figure 2.1: Scheme of the structured light reconstruction method. The setup is composed of a light projector and a sensing camera.

lighting conditions. Woodham originally proposed Photometric Stereo in 1980 [article12 40], and the special case where The data is a single image called'shadow shape', which was compared and analyzed by BKPHornin1989 [book 8]. Shadow photogrammetry uses light sources and cameras to deduce the shape and contour of an object by analyzing the shadow cast on its surface [article13 48]. It involves capturing a series of images from a consistent viewpoint of a light source with a known movement pattern. Use the motion of the shadows cast to reconstruct the structure of the scene [article14 21], especially effective for simple topological objects [article15 16].

These are some of the advantages and limitations: [article15 16]
Photogrammetry enables the creation of precise and detailed 3D models, capturing textures and intricate features directly from photographs. It requires only a high-quality camera and software, making it a budget-friendly solution for various applications. Its result is a 3D mesh that is easy to manipulate, such as scaling, rotating, and translating, making it ideal for scanning objects. However, since it focuses on the photographed object rather than the surrounding environment, it is less suitable for capturing entire environments. Photogrammetry suffers from being heavily reliant on the quality of the source images, meaning factors like resolution and lighting conditions significantly impact the final model. Low-quality or poorly lit photos can lead to inaccurate reconstructions. Additionally, photogrammetry often struggles with reflective, transparent, or featureless surfaces, as these may not capture well in photographs. This limitation results in poor performance when scanning objects with these characteristics. Another drawback is the visible artifacts that can occur during the image-stitching process. In addition, gaps in the capture can lead to noticeable flaws, such as holes in the mesh or jagged edges in the scanned object.

Multi-Sensor Fusion

Multi-source heterogeneous information fusion (MSHIF) uses information obtained from different sensors, such as radar, lidar, camera, ultrasound, infrared thermal imager [29], GPS, IMU, and V2X, to overcome the limitations of individual sensors and create a more comprehensive perception of the environment or target, thus improving the accuracy of 3D reconstruction [46]. A multimodal 3D object reconstruction method based on variational autoencoders. This method automatically determines the modality during the training, which includes specific categories of information. Using the transmission elements of the prior distribution, it determines the pattern of latent variables in the latent space, enabling robust implementation of latent vector retrieval and 3D shape reconstruction.

2.1.2 Passive 3D Reconstruction Methods

The passive 3D reconstruction method does not interfere with the object. It solely uses optical sensors to capture the light reflected or emitted from the object's surface and determines its 3D structure based on the image data.

Texture Mapping

For objects with obvious texture features, using texture information on the object surface to map the two-dimensional image to the 3D model can significantly improve the realism of the model's appearance. However, this process requires a higher texture quality [1]. Lee et al. [13] directly associated the vertices of the implicit geometry with a voxel grid having texture coordinates and applied spatially varying perspective mapping to the input image, enabling real-time texture distortion and geometry update. It utilized background noise smoothing technology within a self-supervised framework to perform high-fidelity texture generation in high-resolution scenarios.[47] Here are some advantages and limitations of this technology: Texture mapping enhances 3D models by applying detailed textures, improving their visual appearance and realism. It allows artists to add surface details, colors, patterns, and attributes, making models look more lifelike and natural. It allows artists to efficiently apply premade or custom textures to 3D models, saving time and effort.[39]

Shape from Focus

The focusing method uses the camera focal length adjustment to calculate depth information by observing changes in the focal depth of the object. This is determined by the degree of image blur of the object at various focal lengths. Use a camera to capture

images of the same scene at various focal lengths. In the image, the farther the object is from the focal plane, the blurrier its image will become. Depth estimation is another important aspect to consider. Using the relationship between image blur level and depth, it is possible to estimate the object. The depth value of each part, and finally the 3D reconstruction, convert the depth information into 3D coordinates, thereby obtaining the 3D reconstruction model of the object. The texture method is often used for close-range shooting and is useful when dealing with low-texture or transparent objects.[s24072314 47]

Here are some advantages and limitations of this technology.
Shape from Focus produces high-resolution surface details by capturing focus changes, enabling fine shape representation. Unlike photogrammetry, it does not require special lighting, making it versatile in various environments. In addition, it is cost-effective since it only needs a standard camera system and focus adjustments, making it a more affordable option compared to other advanced 3D scanning methods. However, it is best suited for small objects with well-defined surfaces, as it captures focus depth from multiple perspectives. It relies on sharp focus changes, so objects lacking clear texture or contrast may yield poor depth information. Additionally, it struggles with homogeneous surfaces like shiny, transparent, or featureless ones, as they provide insufficient focus variation for accurate depth estimation.[308479 24]

Structure from Motion (SFM)

Structure-from-motion (SfM) [10.1007/3-540-61123-1_183 33] is a technique used to create a 3D model of an object using an unorganized set of images captured from various unknown cameras. To achieve this, several offline computational steps are needed, such as estimating the camera's intrinsic and extrinsic parameters and triangulating image points into 3D space. These steps are summarized in Figure 5.2.

The process begins by extracting key points from each image. Points are distinctive pixel locations that are easy to recognize in different images and contain valuable information. To enhance their recognition, key points are typically associated with descriptors, which include information about the neighboring pixel structure. Commonly used keypoint detectors, such as SIFT features [790410 18], provide descriptors that are invariant to scale, rotation, and lighting conditions. By matching features, key points between two images are paired by calculating distances in the feature space. Pairs of images with sufficient matching key points are identified and organized into a new structure. These pairs are then used to estimate the intrinsic and extrinsic parameters of the camera using RANSAC-based algorithms [1288525 26]. Once this step is completed, the problem is solved as a triangulation issue, similar to photogrammetry, where the 3D positions of key points are reconstructed for computational efficiency. Triangulated points are optimized through a

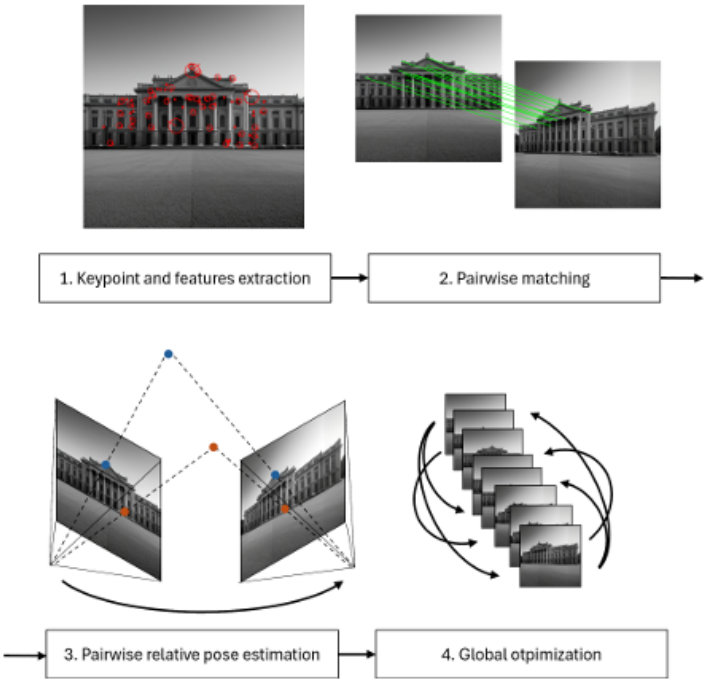

Figure 2.2: Structure from Motion algorithm scheme.

bundle adjustment process to minimize reconstruction errors. SFM is particularly effective in reconstructing large-scale areas, such as city monuments, even when images are captured at different times by different agents. In industrial applications, SfM is often combined with known-calibrated cameras for more precise surface reconstruction. This method is cost-effective, relieving on simple RGB images instead of expensive calibrated cameras, and it does not require synchronized data collection, as different images of the scene taken over time can be used. Although the theoretical foundations of SFM were established at the end of the last century, recent advances in hardware optimization and parallel computing have made its offline computational cost comparable to other methods. One limitation of SfM is its relatively low reconstruction precision as it depends on sparse keypoint triangulation, unlike denser point-based techniques. Additional steps are required to generate a denser 3D model.

Here are some advantages and limitations of this technology.[36] article19
It is noninvasive, as it reconstructs 3D models from 2D images without physical contact or specialized equipment. It is cost-effective, using standard cameras instead of expensive scanners, making it an affordable option. SfM is also highly scalable and capable of handling objects of various sizes, from small items to large landscapes, by processing multiple images. However, it heavily depends on the quality of the image, including resolution, overlap, and sharpness. For large-scale reconstructions, the process can become computationally intensive, requiring substantial processing power and time. Additionally, SfM

struggles with textureless surfaces, as it relies on matching features in images, making it difficult to accurately reconstruct smooth or featureless objects.

2.2 Dynamic 3D Reconstruction Methods

Dynamic 3D reconstruction aims to capture and present the 3D structure of objects and environments, as well as their changes in dynamic scenes. It involves effectively handling dynamic factors such as moving objects, lighting changes, and scene evolution to create accurate and up-to-date images that reflect the current state of the scene. The essence of dynamic 3D reconstruction lies in capturing and modeling the 3D structure of an object or scene as it experiences dynamic changes, such as object movement, variations in lighting conditions, or environmental changes. Dynamic 3D reconstruction methods are typically based on techniques such as feature point matching and motion estimation. Feature point matching is used to track key feature points in the scene, while motion estimation is used to estimate camera motion between adjacent frames.

Here are some advantages and limitations of this technology.
Dynamic 3D reconstruction captures moving objects and changing environments, providing realistic models with depth, texture, and motion. It is versatile, applicable to various environments, including indoor and outdoor spaces, and effectively handles both static and dynamic elements. However, it demands high computational power and time, especially for large scenes. Its complex algorithms require advanced hardware and software, and processing multiple views can lead to data overload, creating storage and processing challenges.

2.2.1 Multi-View Dynamic 3D Reconstruction

Multi-view dynamic 3D reconstruction involves capturing a scene from multiple angles using several cameras or video cameras and incorporating temporal data to recreate the 3D structure of a moving scene. To achieve this, all cameras must be synchronized to capture images simultaneously, ensuring that timestamps are consistent across different viewpoints. Maintaining coherence between consecutive frames is crucial for accurate matching and reconstruction.

For each image frame, computer vision techniques identify key feature points or descriptors. By matching these features across images, correspondences between different camera perspectives are established. This process integrates the camera pose information with the scene structure, allowing for simultaneous scene modeling and camera localization [30]. Motion estimation and motion filtering techniques are then applied to handle

shao2023tensor4defficientneural

dynamic elements within the scene. The resulting 3D point cloud or model undergoes optimization and post-processing to improve accuracy and remove noise [9271925 34].

A key application of dynamic 3D reconstruction is the estimation of human poses. By analyzing the captured data, it determines body posture at each time step, including joint angles and body proportions. Compared to other flexible body movements, human motion follows predictable patterns. The shape of the human body adheres to a structured geometric distribution, making parametric models such as SMPL/X [inproceedings2 17] widely used in research. These models, along with extended versions for hands, faces, and other body parts, provide a structured way to describe human geometry in academic studies [Romero_2017 28].

Here are some advantages and limitations of this technology:
Multiview dynamic 3D reconstruction creates highly accurate models by capturing detailed 3D information from various perspectives. It captures dynamic, moving scenes, offering enhanced realism and depth. Versatile, it can be applied to both indoor and outdoor environments, handling static and dynamic elements effectively. However, it requires significant computational resources and time, especially for large-scale or high-resolution models. The use of multiple cameras can lead to data overload, making storage and processing difficult. Occlusions and low-quality input data, such as poor lighting or motion blur, can also hinder accurate reconstruction.

2.2.2 Dynamic 3D Reconstruction Based on RGB-D Camera

In dynamic 3D reconstruction based on RGB-D cameras, depth information and color image data are input. Advanced computer vision algorithms and technologies are utilized to process data gathered by sensors to fulfill requirements such as real-time performance, reconstruction accuracy, and perception of dynamic objects. Dynamic 3D reconstruction algorithms based on binocular cameras generally involve processes such as identifying and tracking objects, estimating camera poses, calculating depth information, and creating 3D models in real-time.

2.2.3 Simultaneous Localization and Mapping (SLAM)

SLAM is used to map unknown environments while simultaneously tracking the position of moving objects. It relies on various sensors, and the choice of sensors influences the specific SLAM algorithm used [MAGNABOSCO2013195 19]. By integrating data from visual and inertial sensors, SLAM enhances the accuracy of motion and orientation estimation in dynamic settings. Inertial data is especially useful for tracking movements that might not be immediately visible over short periods. [li2024ddnslamrealtimedensedynamic 14].

SLAM plays a crucial role in navigation, operating in real-time (online SLAM) or processing recorded data afterward (offline SLAM). In dynamic environments, recognizing previously visited locations is essential. Loop closure detection is a critical component of this process, helping to reduce mapping errors by continually refining and updating the environment map [38].

Several advances have improved the efficiency of SLAM. Yan et al. [45] introduced GS-SLAM, which incorporates a 3D Gaussian representation to enhance SLAM systems. This approach uses a real-time differentiable splatting rendering technique, significantly optimizing mapping and RGB-D re-rendering speeds. GS-SLAM also implements an adaptive 3D Gaussian strategy for efficient reconstruction of newly observed geometries. Matsuki et al. [20] developed a real-time SLAM system that utilizes 3D-GS for incremental 3D reconstruction, adding geometric verification and regularization to resolve ambiguities in dense mapping. This method works well with both mobile monocular cameras and RGB-D cameras.

Here are some advantages and limitations of this technology.

Unlike static 3D reconstruction, dynamic 3D scene reconstruction must account for continuously changing elements such as moving objects, shifting lighting conditions, and evolving structures. This challenge requires advanced techniques for motion estimation, recognition, and analysis. With the increasing demand for real-time, high-precision, and complex 3D scene reconstruction in fields like the Metaverse and General Artificial Intelligence (AGI), there is a noticeable gap between current dynamic reconstruction capabilities and practical application needs.

2.2.4 Gaussian Splatting

3D Gaussian Splatting is a technique for rendering photorealistic scenes in real-time, using a set of images as input. It was introduced in 2023 by Kerbl in the paper titled "3D Gaussian splatting for Real-Time Radiance Field Rendering"[11]. It uses millions of Gaussians to represent the scene during rasterization. Optimize a collection of 3D Gaussians placed in 3D space to create photorealistic reconstructions and enable quick rendering of new viewpoints. Gaussians are selected for their ability to render efficiently through α-blending, without the need to compute normals. A Gaussian $G(x)$ s fully defined by a covariance matrix Σ and is centered at the point μ

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

For each primitive, its projections in camera space can be determined by:[<**empty citation**>]

$$\Sigma' = JW\Sigma W^T J^T$$

Here,Σ' represents the covariance matrix in camera coordinates, W denotes the viewing transformation, and J corresponds to the Jacobian of the affine approximation of the projective transformation. To ensure proper optimization of the covariance matrix Σ - which is of physical significance only when it remains positive definite during training - a new matrix decomposition is introduced to enforce this constraint.

$$\Sigma = RSS^T R^T$$

S and R represent a scaling matrix and a rotation matrix, respectively, and these two components are optimized separately. Color is modeled using spherical harmonic coefficients (SH) to accurately capture color variations based on different viewpoints, while the α parameter regulates Gaussian transparency in a multiplicative manner. Additionally, during training, a process called Adaptive Control of Gaussians is applied to refine the learned representation by increasing density in specific areas. This method targets regions with high view-space positional gradients, which typically correspond to missing geometric details or sparsely represented areas with large Gaussians. To improve the representation, new primitives are introduced by either splitting larger ones or randomly populating the region of interest. Furthermore, elements with an α value below a certain threshold $\alpha\tau$are removed to retain only relevant Gaussian contributions.

Optimization is carried out by repeatedly projecting Gaussians onto target views, calculating the loss, and back-propagating the gradient to accurately determine their position, rotation, color, transparency, and quantity in the representation. This entire process is illustrated in Figure 3.3. The Gaussian Splatting technique starts with an initial candidate derived from a Structure-from-Motion (SfM) process applied to the input images. During this stage, camera parameters for each view, which are initially unknown, are estimated. The extracted point cloud provides the centers μ for the initial Gaussians, while the covariance matrices Σ are randomly assigned. This step requires a large number of overlapping images to accurately determine camera parameters and properly align input images.

Gaussian positioning enables fast novel view rendering due to the straightforward computation of each Gaussian projection matrix. The visualization process can be performed in real time. However, like other implicit representation methods, editing the representation for applications such as animation remains challenging.

Despite this, Gaussians rely on geometric primitives that have clear spatial positioning, rotation, and scale, making them more interpretable compared to other implicit

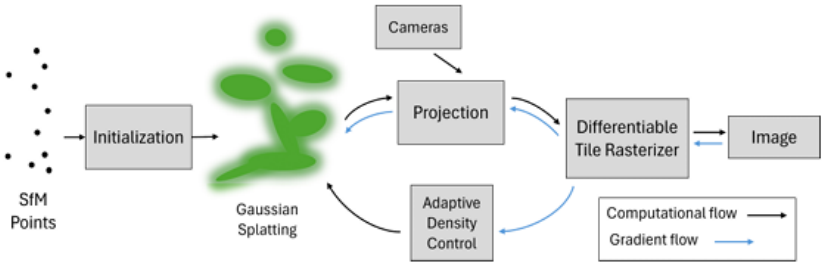

Figure 2.3: The Gaussian pipeline scheme [kerbl20233dgaussiansplattingrealtime] begins with a point cloud obtained through Structure from Motion. This initializes the Gaussian splatting model, which is then optimized by repeatedly projecting Gaussians based on camera poses and comparing them to ground truth views to backpropagate the gradient.

representations. Some research has explored animating Gaussian splatting using video data by learning the motion of Gaussians over time and adjusting lighting conditions. Other studies have aimed to convert Gaussian representations into explicit formats for easier reconstruction and industrial applications.

For example, SuGaR [guÃľdon2023sugarsurfacealignedgaussiansplatting] introduced additional loss functions and a refinement step to better align Gaussians with the ground truth surface, followed by Poisson surface reconstruction. Meanwhile, 2D Gaussian Splatting (2DGS) [13] proposed using 2D Gaussians instead of 3D ones, aligning them with object surfaces to improve geometric accuracy. The marching cube approach was then used to extract a mesh representation of the scene.

Here are some advantages and limitations of this technology.
Gaussian splatting produces exceptionally detailed images by utilizing multiple scans of an object, allowing for real-time exploration from any perspective. This technology can be trained quickly and generates smaller files compared to conventional 3D scene formats used in the metaverse, digital twins, spatial computing, and virtual reality (VR). Although its file sizes are slightly larger than those of neural radiance fields (NeRF), Gaussian splats deliver superior quality and enable a more immersive, interactive experience

2.2.5 Image Segmentation

Image segmentation is crucial in 3D reconstruction as it helps divide objects or scenes within an image into distinct regions, offering more precise and valuable data for the subsequent 3D modeling process. This technique has several applications, such as object segmentation [6618864], background removal[7743326], contour extraction [inproceedings19], semantic segmentation, and dynamic scene segmentation [7378903]. By utilizing effective image segmentation, the

accuracy and stability of 3D reconstruction can be enhanced, resulting in a 3D model that contains richer semantic details. Image segmentation algorithms can be chosen according to the particular scenario and needs. Traditional techniques remain effective in certain situations, while deep learning methods often provide more precise segmentation when trained on extensive datasets. The choice of algorithm generally depends on factors such as the specific application requirements, available computing resources, and the amount of data at hand.

Here are some advantages and limitations of this technology.

Image segmentation enhances analysis by dividing images into meaningful parts, improving object detection, 3D reconstruction, and object tracking. It helps isolate features in complex scenes, allowing applications such as medical imaging, autonomous vehicles, robotics, surveillance, and motion detection to be incorporated. This technology, especially deep learning-based, is computationally intensive and requires large datasets for training. It can also be sensitive to variations in data, such as changes in lighting, scale, or occlusions, leading to potential inaccuracies in segmentation.

2.3 3D Reconstruction Methods Based on Machine Learning

2.3.1 Deep Learning Methods

Deep learning techniques surpass many traditional machine learning approaches in several fields, particularly in computer vision. As technology continues to evolve, neural network-based methods for dynamic 3D scene reconstruction have gained significant attention from researchers. Neural networks are capable of uncovering feature information that may be difficult for humans to interpret and can extract complex, high-dimensional features.

Point Cloud

3D point cloud processing algorithms include: **voxel-based algorithms**, **view-based algorithms**, and **point-based algorithms**. The point-based algorithm directly uses point coordinates as input and can learn directly from the original data in an end-to-end manner, simplifying feature engineering and rule design in the traditional process. It has strong generalization ability and robustness and is suitable for scenarios of all types and sizes.

Point-BLS, which extracts feature sets for points using a feature extraction network

based on deep learning, followed by a comprehensive classification learning system.

Recent research has focused on methods that generate point cloud objects during training, addressing the challenging and time-consuming task of data annotation [42]. These point cloud generation methods include self-reconstruction, point cloud GAN [41], upsampling, and completion, depending on the specific pretask involved.
Here are some advantages and limitations of this technology:[25] Point clouds offer detailed and precise 3D representations of objects and environments, capturing fine surface details. They are versatile for various applications like 3D modeling, object recognition, and robotics, and provide a direct representation of real-world environments from measurements like LiDAR. Point clouds require significant storage and computational power, especially for large or real-time data. They may have sparse or incomplete data, lack semantic information for higher-level tasks, and contain noise or errors from sensor limitations or environmental factors that affect quality. However, it requires significant storage and computational power, especially for large or real-time data. They may have sparse or incomplete data, lack semantic information for higher-level tasks, and contain noise or errors from sensor limitations or environmental factors that affect quality.

Neural Radiance Fields

NeRFs (Neural Radiance Fields) are trained using images of an object or scene captured from various viewpoints. The training algorithm calculates the relative positions of the images and adjusts the neural network's weights to match the data with the images. Here is how the process works in detail: Training begins with a set of images of an object or scene taken from different angles, ideally using the same camera. In the first step, a computational photography algorithm determines the camera's position and orientation for each photo in the collection. This information, along with the images, is then used to train the neural network. The weights of the neural network are updated based on the differences between the pixels in the images and the expected results. This process is repeated around 200,000 times until the network achieves a satisfactory NeRF. Initially, this process took days, but with recent optimizations by Nvidia, the entire procedure can now occur in parallel in just a few seconds.

Here, each step of the Figure 3.4 above: a) **X, Y, and Z** denote the 3D coordinates of a point along the ray. There is not just a ray created, but also defining points along it. For each point in the 3D space, the objective is to determine its color and density. θ **and** φ represent the azimuthal and polar angles, which define the viewing direction. Every point matters, and an object may appear differently from various perspectives, which is how reflections and lighting effects are achieved. b) In this scenario, the Neural Network is a simple multi-layer perceptron (MLP), which aims to predict the color and density

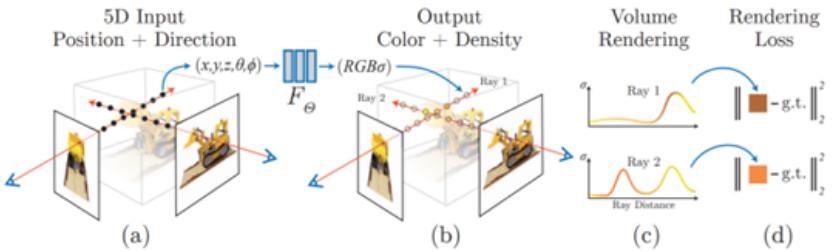

Figure 2.4: Schema from [23] for NeRF pipeline

of each point along every ray. The process involves a technique called "Ray Marching," which defines points along a ray, and then queries the neural network at each point to predict the radiance (color and density).

Here is the process:

- A ray is created.
- The ray is divided into multiple points.
- For each point, a linear model is queried.

Ultimately, we predict four values: R, G, B, and sigma (density). At this stage, rays are created for every pixel, and for each point on the ray, color and density are predicted. c),d) To render a 3D scene accurately, first, there must be eliminated points that are in the "space." For each point, we will check if it intersects with an object based on the predicted density. An advantage of NeRF is that it can generate detailed and accurate 3D models of complex scenes, capturing fine surface features and reflections.[23] It offers a continuous scene representation that can be queried at any point, supporting applications like object manipulation and rendering.

Here are some advantages and limitations of this technology.

The representation of NeRF suffers from vulnerability to sampling and aliasing problems, which can lead to significant artifacts in the synthesized images. One limitation of NeRF is that it requires a large dataset of high-quality images to train the network. This data set needs to capture the scene from various viewpoints, which can be challenging and time-consuming. The quality of the data set also plays a critical role in the performance of the network. If the data set is noisy or contains artifacts, it can significantly impact the quality of the synthesized views.

Chapter 3

Methodology

In this chapter, we discuss all the steps of building the 3D reconstruction avatar pipeline. An important place has Gaussian splatting, which is a highly effective technique for creating photorealistic 3D avatars while ensuring efficient rendering. It is a method first introduced in 2023, so its full potential is still being studied. This technique begins by capturing images from multiple angles, using SfM techniques to generate a 3D point cloud. Each point is converted into Gaussian splats that represent position and color. Then a training process refines details such as scale, covariance, and transparency, creating millions of particles. During rendering, Gaussian rasterization transforms these splats into colored pixels, replacing traditional triangle-based meshes with Gaussian blurs that directly capture texture. It combines sophisticated mask filtering for accurate segmentation, enhancing geometric precision, and optimization effectiveness to produce a high-quality, accessible 3D avatar. In the following, each step of the pipeline-building process is explained.

3.1 Data Acquisition and Preprocessing

First, 6 videos in total are used as input data for the study. Then the Gaussian splatting extracts frames from the input video. These frames contain the 2D images of the scene from different angles or viewpoints, which are crucial for constructing accurate 3D avatars. 2 of these videos are recovered with the smartphone and 4 of them were provided by this paper [shao2024splattingavatarrealisticrealtimehuman]. From each video, the extracted frames are used to estimate avatar keypoint poses using OpenPose [cao2019openposerealtimemultiperson2d] and additionally before the images are fed to the pipeline, the person images are segmented using Segment Anything (SAM) [kirillov2023segment].

3.1.1 OpenPOSE

OpenPose is applied to the extracted frames to detect key body joints (head, shoulders, elbows, hands, hips, knees, feet, and facial landmarks) from 2D images or frames.

OpenPose is an open source, real-time, multi-person pose estimation library developed by Carnegie Mellon's Perceptual Computing Lab.[4] Detects and tracks human body, hand, face, and foot key points from images or videos using deep learning. Using convolutional neural networks (CNNs), OpenPose identifies body parts and connects them to form a structured pose. It supports 2D and 3D pose estimation and is capable of tracking multiple individuals in a single frame. The method is built upon Part Affinity Fields (PAFs), which encode both location and orientation information of body parts to facilitate pose estimation in an end-to-end manner.

Given an input image $I : \Omega \to \mathbb{R}^3$ defined over a spatial domain $\Omega \subset \mathbb{R}^2$, OpenPose first computes body part confidence maps $S^j : \Omega \to [0, 1]$ for each keypoint j. These confidence maps are obtained by applying a convolutional neural network $\mathcal{F}_S$ parameterized by θ_S:

$$S = \mathcal{F}_S(I; \theta_S), \tag{3.1}$$

where $S = \{S^j\}_{j=1}^{J}$ represents the set of confidence maps for J body keypoints. Each map S^j estimates the probability of keypoint j occurring at each pixel location.

In addition to keypoint detection, OpenPose models pairwise associations between body parts using Part Affinity Fields (PAFs), represented as vector fields $F^c : \Omega \to \mathbb{R}^2$. Each F^c encodes the orientation and position of a limb c, facilitating the grouping of key points into coherent poses. The PAFs are computed as follows:

$$F = \mathcal{F}_F(I; \theta_F), \tag{3.2}$$

where $F = \{F^c\}_{c=1}^{C}$ is the set of vector fields for C limb connections, and $\mathcal{F}_F$ is another CNN branch parameterized by θ_F.

Each PAF F^c at location $x \in \Omega$ is defined as a unit vector along the direction of the limb connecting two keypoints (j_1, j_2):

$$F^c(x) = \begin{cases} v_{j_1 j_2}, & x \text{ lies on the limb segment} \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

where $v_{j_1 j_2} = (x_{j_2} - x_{j_1})/\|x_{j_2} - x_{j_1}\|$ is the unit vector from keypoint j_1 to keypoint j_2.

The training objective consists of two loss functions: a confidence map loss L_S and a PAF loss L_F, both formulated as mean squared errors (MSE) against ground truth annotations $\hat{S}$ and $\hat{F}$:

$$L_S = \sum_{j=1}^{J} \sum_{x \in \Omega} \left\| S^j(x) - \hat{S}^j(x) \right\|^2, \tag{3.4}$$

$$L_F = \sum_{c=1}^{C} \sum_{x \in \Omega} \left\| F^c(x) - \hat{F}^c(x) \right\|^2. \tag{3.5}$$

The total loss is thus given by:

$$L = L_S + L_F, \tag{3.6}$$

This is minimized via backpropagation using gradient-based optimization techniques.

During inference, key points are extracted as local maxima in the confidence maps. The optimal assignment of key points to the skeletal structures is achieved by solving a bipartite graph matching problem. Let $K = \{k_i\}_{i=1}^{N}$ be the set of key points detected for a given part and let $E = \{(k_i, k_j)\}$ be potential limb connections weighted by PAF confidence scores. The goal is to maximize the total association score A:

$$A = \sum_{(k_i,k_j) \in E} \sum_{x \in L(k_i,k_j)} F^c(x) \cdot v_{k_i k_j}, \tag{3.7}$$

where $L(k_i, k_j)$ denotes the set of points sampled along the segment connecting k_i and k_j. This optimization problem is solved using greedy matching or graph-based optimization techniques.

3.1.2 Segment Anything (SAM)

Almost in parallel with OpenPose activity, SAM runs in parallel to create a segmentation mask to separate the foreground object - the person - from the background.

The Segment Anything Model (SAM) is a deep learning-based segmentation framework designed to generalize across diverse image segmentation tasks without requiring task-specific fine-tuning. Unlike conventional segmentation models that are trained on specific datasets and struggle with unseen objects or domains, SAM introduces a prompt-based segmentation approach, allowing it to segment any object given an appropriate user-specified cue.

That is how it works:

The key innovation of SAM lies in its ability to operate in a zero-shot manner, meaning it can accurately segment objects in new images without additional training. This

is achieved by combining a powerful vision transformer-based feature extraction network with an interactive prompt mechanism that guides the segmentation process. Using different types of input prompts, the target object is defined in an image. These prompts can include point annotations, bounding boxes, or rough segmentation masks. The model then refines the segmentation based on these cues, producing an accurate object mask. This prompt-based mechanism makes SAM adaptable to a wide range of applications, from medical imaging to autonomous navigation, where precise object delineation is required. Given an input image, SAM first encodes its visual features using a pre-trained transformer model. The encoded image representation is then processed alongside the provided prompts. The model interprets the prompts in the context of the image features, identifying the most relevant regions, and generating a segmentation mask that best matches the given input. This process allows for interactive and adaptive segmentation, meaning users can refine or modify the segmentation by providing additional prompts. Following the methodology proposed in SplattingAvatar, we estimated different segmentation masks using SAM and used only the one with the biggest connected-area component. This is reasonable since the input video should be in controlled setting where only the desired person to be reconstructed is visible without any major obstruction and any flat-like background.

3.2 Reconstruction 3D Avatar

In this stage of the pipeline, ROMP aligns the key points of the skeleton avatar with the SMPL mesh, tracks movement, and allows 3D avatar reconstruction. ROMP provides a base mesh that serves as the foundation for initializing Gaussians. Each frame is mapped to a standardized pose using a transformation into a canonical pose space, allowing consistent training across frames.

Additionally, the base mesh plays a crucial role in binding Gaussians to the rigging system. This is accomplished through the triangle-walking method from the SplattingAvatar paper, which enables the Gaussians to follow the deformation of the mesh. As a result, the Gaussians dynamically adjust to different poses, ensuring accurate movement and realistic rendering of the 3D avatar across various animations. For each technology, there is a brief explanation of the technique and the use and observation of particles in the execution of the pipeline.

3.2.1 ROMP

ROMP is a deep learning model designed for 3D [11]human pose and shape estimation from a single image or video. Predicts SMPL parameters that define a human body model using a skeletal structure, pose, and shape. This model enhances Gaussian splatting by providing accurate 3D body pose and shape estimations. While Gaussian Splatting uses Gaussian primitives for efficient 3D scene representation, it lacks explicit skeletal control, making it challenging for animation. ROMP fills this gap by predicting SMPL-based parameters, including pose, shape, and translation, from images or video. These estimations help initialize Gaussians based on the detected human pose and shape, ensuring a more realistic avatar representation. ROMP also refines motion tracking for smoother animations and integrates skeletal data, improving the control and accuracy of animated avatars created through Gaussian Splatting.

In this step of the pipeline, in the execution of the ROMP, the match of the keypoint joins of the skeleton avatar with the mesh obtained from the SMPL parameters is observed. They are very important because they help us determine the avatar poses. After frame-by-frame ROMP tracking and then comparing the poses with each video, it is observed that all the original avatar poses were captured. It is a very positive result because not only does the system have image segmentation and pose for each of the avatars of the videos, but also a sample with the pose and trajectories that can be applied to other avatars, of course, of other videos, and see how their motion reacts. Let us explain in detail the benefits of this approach in the pipeline.

- In the output from the ROMP model, we have the trajectories and poses of our avatar, and we can apply filters to smooth the pose signal. In this step of the pipeline, we are going to apply signal filtering to remove noise from the signal and make the animation more fluent and realistic. This will be explained more in **section 3.2.2**

- Moreover, we have our 3D avatar with the segmented images and poses. What we can do is match the trajectories and poses of one specific avatar with another 3D avatar, to reconstruct the final video. We did some experiments applying one avatar to the other avatar posing, and it moved when doing the avatar poses.

Here is the final pipeline figure:

This design is highly flexible, allowing individual stages to be replaced in a modular way without affecting the entire process. Such adaptability is especially beneficial for different applications; for instance, offloading certain computational stages to external machines can enable the pipeline to function on devices with limited processing power. In the above section, a detailed explanation of each stage is provided.

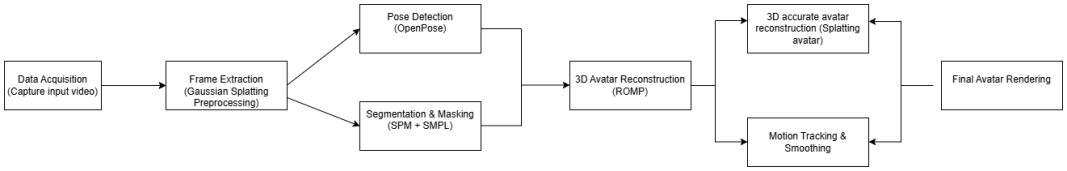

Figure 3.1: 3D Avatar Reconstruction pipeline

3.3 Signal Filtering

We took an input video of average length, trying to reconstruct the 3D avatar video. Plotting and analyzing the pose signal it was observed that the signal had a lot of noise, the high-frequency signal part, meaning that the avatar did not move in a good normal way. To remove the noise signal and smooth the motion, we applied the filtering pose signal. As we shall see, this made the final 3D reconstructed avatar more smooth in motion. Let's see in detail the steps we made. In the process, we had to achieve the pose signal from the Gaussians. The Average signal approach was applied, which will be explained in the following. The average value of the signal refers to the mean or average value of a signal over a certain period or a set of samples. In signal processing, it is commonly used to characterize the central tendency or overall behavior of a signal.

To calculate the signal average, you typically sum the values of the signal over the desired time interval or number of samples and then divide by the total number of samples or the length of the interval. Mathematically, for a discrete signal x(t) with N samples: Signal Average =

$$\text{Signal Average} = \frac{1}{N}\sum_{i=1}^{N} x(t_i)$$

Where:

- $x(t_i)$ represents the signal value at sample t_i,
- N is the total number of samples.

In continuous signals, the signal average can be represented as:

$$\text{Signal Average} = \frac{1}{T}\int_{0}^{T} x(t)\,dt$$

.

After explaining how to obtain the signal, specific work has been done on the Gaussians. At the start, the system can capture the position of each joint in 3D space, represented as (x, y, z) coordinates of the key points, and a θ representing the rotation of each of the joints in every motion. So in the paper [31] it is intended that for each keypoint a new vector is achieved as (x, y, z, θ). This work has been done for all 73 channels. To find

the new coordinates of the key points for every position, the vector has been normalized and extracted: coordinates (x, y, z) and θ to represent the rotation angle of the joint. This process has been performed every time a motion is made to determine the new pose position for every joint involved. This is the previous step of the average signal.

So in this way, we gained the pose signal for each of the 6 videos studied in the pipeline. So let us better explore the signal filtering, giving at the beginning some general information of how they work, and then let us see how they are applied in our pose signal. As signal filtering in the pipeline, for the pose signal, it is chosen: the moving average window and the low-pass band. Let's see in detail the two techniques:

- **Moving Average Window:** moving average window is a commonly used [5]technique for smoothing pose data. It helps reduce noise and create smoother and more stable animations by averaging the values of a signal over a moving window of data points. This technique is especially useful for filtering erratic pose data, such as those obtained from motion capture systems. That is how it works: A moving average filter operates by taking the average of a set of data points within a "window" that moves over time. For a given time t, the window includes data points from t - n to t + n, where n is the window size. The filter then computes the average of all the data points within this window and replaces the value at time t with the computed average. The moving average at time t is given by:

$$\mathrm{MA}_t = \frac{1}{N} \sum_{i=t-N+1}^{t} x_i$$

  Where:

  - MA_t is the moving average at time t,
  - x_i are the data points in the window,
  - N is the window size (the number of points in the moving average),
  - t is the current time or data point.

  Explanation: The moving average at time t is the average of the last N data points (from $t - N + 1$ to t). The window slides as you move through the data, averaging over the next N points.

  - **Choose a Window Size:** The size of the window (often denoted as n) determines how many neighboring data points are used to calculate the average. A smaller window results in less smoothing but better responsiveness to changes, while a larger window smooths more, but reduces sensitivity to rapid changes.

- **Sliding the Window:** The window slides through the data sequence, processing one frame (pose signal) at a time. At each position, the filter calculates the average of the data points within the window.
- **Compute the Average:** For each frame, the filter computes the average value of the pose signals within the window and replaces the original value with this average.
- **Output the Smoothed Signal:** The resulting output is a smoother version of the pose signal, with reduced noise and irregular fluctuations.
- So in the pipeline, the Moving Average Window with these filter parameters: **Sampling rate** is 30; **Window size for the moving average** is 10.

In this case, we could sample each 30HZ, we saw it was more productive using this sampling rate of the pose signal, because trying other sampling rates and then reconstructing the 3d avatar video, it has been noticed that the rendering was not so good in the motion, the avatar sometimes did unnatural movement and it did look granted. The other filter parameter, the moving window, was set to be 10 frames because it was ideal to ensure normal and smooth motion. To improve filtering and try to eliminate the noise in the pose signal, a low-band filter has been applied, which is explained below.

- Low-Pass Filter: A low-pass filter allows lower-frequency signals [32] (unknown representing actual smooth movements) to pass through while attenuating or removing higher-frequency signals (representing noise or rapid unintended fluctuations).
  The transfer function $H(s)$ for a simple first-order low-pass filter is given by:

$$H(s) = \frac{1}{1 + \frac{s}{\omega_c}}$$

  Where:

  - s is the complex frequency variable ($s = \sigma + j\omega$),
  - ω_c is the cutoff frequency (in radians per second).

  In the time domain, the equation for a first-order low-pass filter is:

$$y(t) = \frac{1}{\tau} \int_0^t x(\tau) e^{\frac{t-\tau}{\tau}} d\tau$$

  Where:

  - $x(t)$ is the input signal,

- $y(t)$ is the output signal,
- τ is the time constant of the filter, related to the cutoff frequency by $\tau = \frac{1}{\omega_c}$.

- **Cut-off frequency:** The cutoff frequency is the threshold at which the filter starts attenuating higher-frequency signals. It is crucial to choose an appropriate value for this frequency to ensure that important movement details are preserved while reducing unwanted noise.
- **Working principles:** At frequencies below the cutoff frequency, the output signal is strong and the filter allows most of the signal to pass through. At frequencies above the cutoff frequency, the signal starts to attenuate, with a steeper roll-off compared to passive filters.
- In the pipeline **the cutoff frequency for the low-pass filter (Hz)** is 5; When applying this filter, all the high-frequency parts of the signal are not accepted, causing a smoother filtered signal and no noise (which we know noise is in the high frequencies). After trying and applying different cutoff rates, we saw that the best rate was 5. So immediately every part of the signal higher than 5 Hz is not extracted. The signal is cleaner and smoother, and after reconstructing the avatar video, we saw that his motion was much more natural and smoother than before, and the image during the motion was clearer and better than before.

Chapter 4

Experimental Setups

The builder pipeline is been observed and evaluated in terms of the quantity and quality of the results achieved. Various videos of different lengths, from short about 30 seconds to long about 2 - 2.5 minutes. This proposed technique has been observed in interaction with those input videos to reconstruct the final 3d avatar video. In this chapter, first, the experiment setups have been explained, and then an analysis of the performance matrices and their possible application. The experiments we did with six input videos, where 4 of them were taken from the project represented in the paper [31] and the other 2 videos were done with a smartphone in VarLab, to be included in the experiments. The first 4 videos there demonstrate some easy and simple movements forward, and then the person has moved the arms in a circle movement and even opened and closed their arms. In the other 2 videos recorded in the Varlab, there are some principal movements like moving forward, putting the arms up and then down and then rotating on itself, then going backward. All of these videos were at a normal frame rate, 30 FPS. All filming was done in natural light. The machine properties for the purpose and on which the whole pipeline are in 6 Giga of CPU. The pose sign extracted from these videos has noise because some movements were not very fluid, or in the recorded video some movements were done very fast or very slow. We preferred to record videos in natural light and with moving speed and gestural movement, to do some realistic videos to demonstrate that every ordinary video can be useful for this purpose. We will see then that very long videos do have a long waiting time. For each of the 6 input videos with a length of 30 seconds, in the prepossessing 319 frames were achieved per each video. For each of the videos, OpenPose found the key points of the joint of the person, and all the outputs were processed with SAM and SPML parameters to match the key points inside the segmentation masking. For the system, it took 1 minute and 13 seconds to process the key points for each of the inputs. Afterward, it took about 50 seconds to mask the 3D avatars. The evaluation process and thus reconstruction of the avatars took about 3 hours or even

more depending on the avatar. The server that hosted this pipeline has 2 GPUs with 32 Giga per each.

4.1 Results

The proposed pipeline produces some final 3D reconstructed avatars that have smoother and more realistic motion. It is important to note that the results presented in this report are an improvement for smoothing motion and clear rendering avatar, other words, the original 3DGS paper. It is considered to be an advancement of the original pipeline.

Here is the final pipeline figure:

Figure 4.1: Some of the frames

Both trainings reached the last steps in less than 3 hours, but with a reconstructing pipeline time cost of 3 hours for each of the 3D avatars. Each of the 6 videos was long around 30 seconds and we did get 319 frames per each. In the following to understand better the evaluation of our 3D avatar, we represent an image of the original pose input video, then we have the key points of the joints for the avatar, and at the end the mask achieved from this avatar and applied to another avatar.

This work has been done for every original pose of the input video and other avatars.

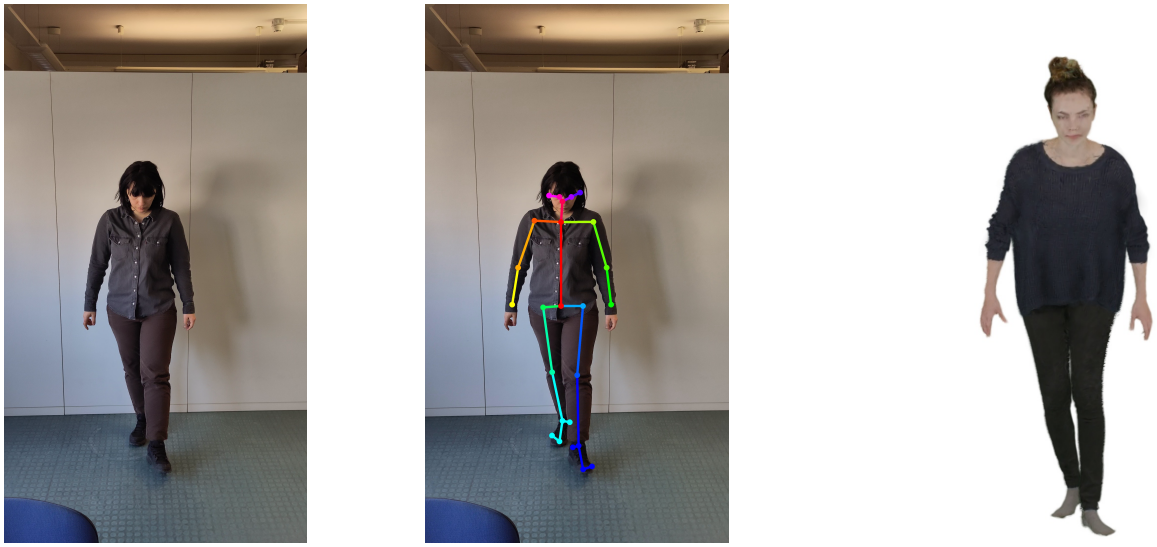

Figure 4.2: Here is the evaluation of the pipeline from the original pose video in the right, then key points for every joint, and the pose applied to the another avatar

We can see that the original pose is identical to the final reconstructed pose. The reconstructed avatar presents some reconstruction that is not corrected around the figure. This is due to the noise in the pose signal which we did remove applying the filtering. In the final avatar, we observed that there was a reconstruction that was not corrected for on the surface around the armpit or ankle. This is because Splatting Avatar can not do a realistic reconstruction to this area where the initial frames are not so clear, or they do not exist at all.

Another important optimization for the pipeline is the filtering of pose signal to eliminate noise and secure smoother motion for the avatar. In the following, an image of the signal with noise is presented, and then the image of the signal is filtered.
Notice that the signal at the beginning had a lot of noise, which is represented by the many variations. After applying the low-pass filter with cutoff 3, all of the frequencies greater than 3 are cut off they are not part of the new signal. Moreover, to reduce noise we apply the moving average window. In this case, we apply another filter to the signal, which is the moving average window of window size 10. In this case, the filter for every

point does the average of the 10 data points near and so on for each data in the signal. It is observed that the noise signal has improved a lot and that is why the avatar motion is more natural.

It would be better if more filters were applied at the same time, depending on the pose signal, to see more improvements in removing the noise.

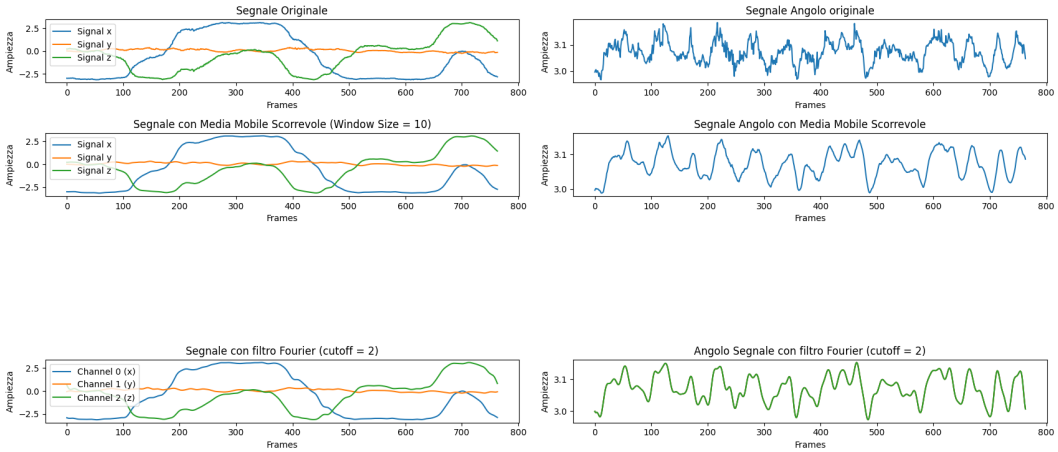

Figure 4.3: Deleting noise from original pose signal

The work done to eliminate noise in pose signals is essential for accurate and realistic 3D avatar reconstruction. Noisy data can cause jerky, unstable, or incorrect movements, making the avatar look unnatural. Smoothing the signal ensures fluid and lifelike animations by eliminating abrupt jumps in joint positions. It was important for the final animation avatar to have natural, realistic movement, to be as realistic as possible.

Conclusion and Future Works

- This proposal is based on the results of the 3d avatar pipeline and filtering poses, to reconstruct a good 3D avatar. Because of this pipeline, people can achieve a 3D reconstructed motion-optimized avatar with just any length of the video in input. This proposed pipeline is very flexible and modular, which allows for future improvement at every stage of it. This method might not be affordable for anyone, because of the high computational power of the machine for preprocessing. This technology is thought to be in all domains. Due to the segmentation step, the pipeline created masks that are ready for any user to use in another 3D avatar video. Filtering the pose signal allowed for a clearer reconstruction, removing the noise signal, and achieving more natural reconstructed avatars. The filtering of the pose signal helped to produce cleaner pose signals and ensure smoother transitions between movements. This is especially important in real-time applications, where any disruption can negatively impact the experience. These improvements are essential for creating visually appealing and stable animations in entertainment, education, and simulation, contributing to more immersive experiences. The usage of Gaussian splitting technology is very important in the transition from explicit mesh representation to implicit ones in science and some producing chains.

- **Future research** In the course of this work, some areas of improvement are detected. Some research work might be necessary to achieve this in single steps of the pipeline, further improving the stability, visual quality, and efficiency of 3D animation. Furthermore, it might be a good approach to improving existing filtering techniques, incorporating advanced machine learning methods, and optimizing performance. During the study of the result in **chapter 4**, we saw that in some final reconstructed videos, the noise in the signal was visible. So, to remove this noise, other approaches might be possible, like machine learning techniques.

In the following, some future approaches of signal filtering are explained: Moving Pose Signal Filtering for Enhanced 3D Gaussian Splatting Animation
Future developments are expected to open up a range of promising possibilities, from

refining filtering techniques to integrating advanced machine learning approaches. Improving the smoothness, consistency, and realism of 3D animations will require ongoing work in areas such as another sort of filtering and advanced signal reconstruction. As these innovations progress, they will help in many fields of education, entertainment, and industry with fluid, visually impressive, almost real animations. A good alternative might be using machine learning for pose signal filtering. In this case, a machine learning model, such as a neural network, can be trained on labeled datasets of clean and noisy pose data. The model learns how to map noisy or incomplete pose data to a smoother signal. Then this trained model can predict and filter pose signals in real-time by identifying the underlying structure and correcting noise or inconsistencies in the input data. For example, this approach could be used to reconstruct missing pose data in motion capture sequences or to smooth erratic movements in 3D animations. Another approach might involve Recurrent Neural Networks (RNNs) for analyzing time-series data for like pose signals. These models can capture temporal dependencies and context, allowing them to filter and predict pose signals based on previous frames. Furthermore, the system can smooth the signal over time and predict future pose positions based on past data. This can help in cases where there is uncertainty or noise in the pose data, such as during fast movements or incomplete motion capture sequences. A key improvement point might be training models on noisy input data, the autoencoder learns how to filter out unwanted noise and retain the true pose signal. This approach is beneficial when labeled data is scarce or when the noise patterns are complex and difficult to model using traditional filtering techniques.

Acknowledge

To you, Mom, who could not see me complete my studies and take another step toward becoming a woman. Today, Mom, as you watch me from up there, I know that you would be proud of me.

I would like to extend my heartfelt thanks and deep gratitude to my supervisor, Prof. Gustavo Marfia, and my co-supervisors, Pasquale Cascarano and Jacopo Megliolaraldi. Their invaluable support made this project possible. Specifically, I want to thank Jacopo Meglioraldi for his support and help in the design, implementation, and completion of this work. The authors express their gratitude to the VARLab laboratory staff for creating a good environment in which everyone can express their abilities. I want to thank Giacomo Vallesciani, a special thank you goes to Lai Mengting, researcher and good friend, as well as Shirin Hajahmadi for all the good vibes and laughs during launch breaks. They created a welcoming workplace. I would like to thank Prof. Claudio Sacerdoti Coen, who was very helpful during these last months. I want to thank my friends Mirvjen, Giulio, and Annina, who have always supported and supported me.
A special thanks to my fiance Alessio Veneziano for all his love and support.

Bibliography

proceedings [1] Dimitrios Alexiadis, Dimitrios Zarpalas, and Petros Daras. "Real-time, realistic full-body 3D reconstruction and texture mapping from multiple Kinects". In: June 2013, pp. 1–4.

xperience6F [2] Chris Alton. "Experience, 60 Frames Per Second: Virtual Embodiment and the Player/Avatar Relationship in Digital Games". In: 2017.

article7 [3] Cihan Altuntas. "Review of Scanning and Pixel Array-Based LiDAR Point-Cloud Measurement Techniques to Capture 3D Shape or Motion". In: *Applied Sciences* 13 (May 2023), p. 6488.

ltiperson2d [4] Zhe Cao et al. *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2019. arXiv: 1812.08008 [cs.CV].

croeconomic [5] Philippe Goulet Coulombe and Karin Klieber. *An Adaptive Moving Average for Macroeconomic Monitoring*. 2025. arXiv: 2501.13222 [econ.EM]. URL: https://arxiv.org/abs/2501.13222.

ansplatting [6] Antoine Guédon and Vincent Lepetit. *SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering*. 2023. arXiv: 2311.12775 [cs.GR].

6618864 [7] Christian Häne et al. "Joint 3D Scene Reconstruction and Class Segmentation". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 97–104.

book [8] Berthold Horn and Michael Brooks. *Shape from Shading*. Vol. 2. Jan. 1989.

7378903 [9] Cansen Jiang et al. "Static-Map and Dynamic Object Reconstruction in Outdoor Scenes Using 3-D Motion Segmentation". In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 324–331.

oceedings19 [10] Aobo Jin and Qiang Fu. "Contour-based 3D Modeling through Joint Embedding of Shapes and Contours". In: May 2020, pp. 1–10.

ngrealtime [11] Bernhard Kerbl et al. *3D Gaussian Splatting for Real-Time Radiance Field Rendering*. 2023. arXiv: 2308.04079 [cs.GR].

2023segment [12] Alexander Kirillov et al. "Segment anything". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026.

roceedings1 [13] Joo Lee et al. "TextureFusion: High-Quality Texture Acquisition for Real-Time RGB-D Scanning". In: June 2020, pp. 1269–1277.

ensedynamic [14] Mingrui Li et al. *DDN-SLAM: Real-time Dense Dynamic Neural Implicit SLAM*. 2024. arXiv: 2401.01545 [cs.CV].

s101110356 [15] Yangming Li and Edwin B. Olson. "A General Purpose Feature Extractor for Light Detection and Ranging Data". In: *Sensors* 10.11 (2010), pp. 10356–10375. ISSN: 1424-8220.

article15 [16] Wai Chung Liu and Bo Wu. "An integrated photogrammetric and photoclinometric approach for illumination-invariant pixel-resolution 3D mapping of the lunar surface". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 159 (Nov. 2019), pp. 153–168.

roceedings2 [17] Matthew Loper et al. "SMPL: a skinned multi-person linear model". In: vol. 34. Nov. 2015.

790410 [18] D.G. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2.

0SCO2013195 [19] Marina Magnabosco and Toby P. Breckon. "Cross-spectral visual simultaneous localization and mapping (SLAM) with sensor handover". In: *Robotics and Autonomous Systems* 61.2 (2013), pp. 195–208. ISSN: 0921-8890.

lattingslam [20] Hidenobu Matsuki et al. *Gaussian Splatting SLAM*. 2024. arXiv: 2312.06741 [cs.CV].

article14 [21] Michael Mccool. "Shadow Volume Reconstruction from Depth Maps". In: *ACM Trans. Graph.* 19 (Jan. 2000), pp. 1–26.

article11 [22] Radomír Mendřický. "Impact of Applied Anti-Reflective Material on Accuracy of Optical 3D Digitisation". In: *Materials Science Forum* 919 (Apr. 2018), pp. 335–344.

cenesneural [23] Ben Mildenhall et al. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020. arXiv: 2003.08934 [cs.CV].

308479 [24] S.K. Nayar and Y. Nakagawa. "Shape from focus". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.8 (1994), pp. 824–831.

guyen20133d [25] Anh Nguyen and Bac Le. "3D point cloud segmentation: A survey". In: *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*. IEEE. 2013, pp. 225–230.

1288525 [26] D. Nister. "An efficient solution to the five-point relative pose problem". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6 (2004), pp. 756–770.

article8 [27] Xiaojuan Qi et al. "Structural Dynamic Deflection Measurement With Range Cameras". In: *The Photogrammetric Record* 29 (Mar. 2014), pp. 89–107.

Romero_2017 [28] Javier Romero, Dimitrios Tzionas, and Michael J. Black. "Embodied hands: modeling and capturing hands and bodies together". In: *ACM Transactions on Graphics* 36.6 (Nov. 2017), pp. 1–17. ISSN: 1557-7368.

amm20102022 [29] Robert Schmoll Sebastian Schramm Phil Osterhold and Andreas Kroll. "Combining modern 3D reconstruction and thermal imaging: generation of large-scale 3D thermograms in real-time". In: *Quantitative InfraRed Thermography Journal* 19.5 (2022), pp. 295–311.

cientneural [30] Ruizhi Shao et al. *Tensor4D : Efficient Neural 4D Decomposition for High-fidelity Dynamic Reconstruction and Rendering*. 2023. arXiv: 2211.11610 [cs.CV].

altimehuman [31] Zhijing Shao et al. *SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting*. 2024. arXiv: 2403.05087 [cs.GR].

unknown [32] Daniel Sogbey. *Design and Analysis of a Low-Pass Passive RC Filter for Audio Signal Applications*. Jan. 2025.

61123-1_183 [33] Peter Sturm and Bill Triggs. "A factorization based algorithm for multi-image projective structure and motion". In: *Computer Vision — ECCV '96*. Ed. by Bernard Buxton and Roberto Cipolla. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 709–720.

9271925 [34] Kai Sun et al. "DRCNN: Dynamic Routing Convolutional Neural Network for Multi-View 3D Object Recognition". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 868–877.

7743326 [35] Raúl Vargas et al. "Background intensity removal in structured light three-dimensional reconstruction". In: *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*. 2016, pp. 1–6.

article19 [36] jérémie Voumard et al. "Pros and Cons of Structure for Motion Embarked on a Vehicle to Survey Slopes along Transportation Lines Using 3D Georeferenced and Coloured Point Clouds". In: *Remote Sensing* 10 (Nov. 2018), p. 1732.

article9 [37] Xue Wang and Peijun Li. "Extraction of urban building damage using spectral, height and corner information from VHR satellite images and airborne LiDAR data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 159 (Jan. 2020), pp. 322–336.

1545285 [38] J. Weingarten and R. Siegwart. "EKF-based 3D SLAM for structured environment reconstruction". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 3834–3839.

7580.267583 [39] Frederick M. Weinhaus and Venkat Devarajan. "Texture mapping 3D models of real-world scenes". In: *ACM Comput. Surv.* 29.4 (Dec. 1997), pp. 325–365. ISSN: 0360-0300.

article12 [40] Robert Woodham. "Photometric Method for Determining Surface Orientation from Multiple Images". In: *Optical Engineering* 19 (Jan. 1992).

latentspace [41] Jiajun Wu et al. *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*. 2017. arXiv: 1610.07584 [cs.CV].

10086697 [42] Aoran Xiao et al. "Unsupervised Point Cloud Representation Learning With Deep Neural Networks: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.9 (2023), pp. 11321–11339.

article10 [43] L Xu et al. "Real-Time 3D Profile Measurement Using Structured Light". In: *Journal of Physics: Conference Series* 48 (Oct. 2006), p. 339.

article6 [44] Yuko Yamashita and Tetsuya Yamamoto. "Effect of virtual reality self-counseling with the intimate other avatar". In: *Scientific Reports* 14 (July 2024).

evisualslam [45] Chi Yan et al. *GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting*. 2024. arXiv: 2311.11700 [cs.CV].

article16 [46] Hyeonwoo Yu and Jean Oh. "Anytime 3D Object Reconstruction Using Multi-Modal Variational Autoencoder". In: *IEEE Robotics and Automation Letters* PP (Jan. 2022), pp. 1–1.

s24072314 [47] Linglong Zhou et al. "A Comprehensive Review of Vision-Based 3D Reconstruction Methods". In: *Sensors* 24.7 (2024). ISSN: 1424-8220.

article13 [48] Linglong Zhou et al. "A Comprehensive Review of Vision-Based 3D Reconstruction Methods". In: *Sensors* 24 (Apr. 2024), p. 2314.