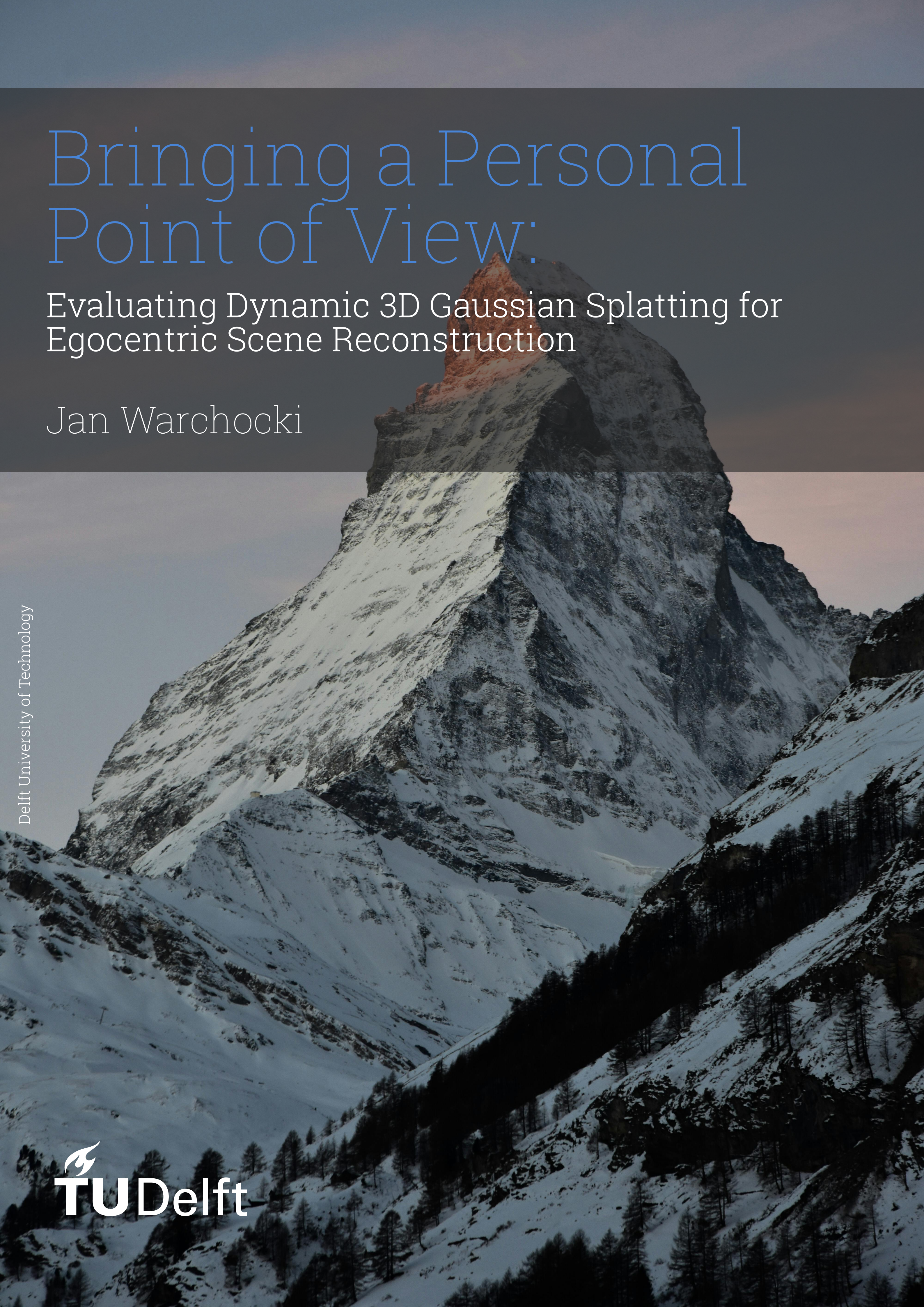
Bringing a Personal Point of View:

Evaluating Dynamic 3D Gaussian Splatting for Egocentric Scene Reconstruction

Jan Warchocki

Delft University of Technology



Bringing a Personal Point of View:

Evaluating Dynamic 3D Gaussian Splatting for Egocentric Scene Reconstruction

by

Jan Warchocki

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday June 18, 2025 at 11:00 AM.

Student number:	5344646	
Project duration:	December 1, 2024 – June 18, 2025	
Thesis committee:	Dr. Jan van Gemert	TU Delft, supervisor
	Dr. Michael Weinmann	TU Delft

TU Delft

Preface

This work is the culmination of my Master's thesis for Computer Science at the Delft University of Technology. I would like to express my sincere gratitude to my supervisor, Dr. Jan van Gemert, for his continuous support, insightful feedback, and for encouraging me to think critically and independently throughout the process. His guidance played a key role in shaping the final outcome of this thesis. I also thank Dr. Xi Wang and Jonas Kulhanek for their early input, involvement during the initial stages of the project, and for providing feedback on our submission to the British Machine Vision Conference. Although the direction of the work evolved significantly over time we worked together, I appreciated the opportunity to explore different perspectives. Lastly, I am deeply grateful to my family and friends for their encouragement and patience along the way.

Jan Warchocki
Zürich, June 2025

Contents

1
Introduction

Imagine wearing a camera on your head that captures everything you see from your own point of view. This type of video, called egocentric or first-person video, offers a unique way of understanding how people experience and interact with the world around them. It has many potential applications, especially in fields such as augmented reality (AR) [15], assistive technology [55], and robotics [18], where understanding a person's direct experience is crucial.

However, working with egocentric video is not without its challenges. For one, the camera is attached to the person's head, moving unpredictably [32, 40]. Additionally, the scenes in egocentric video can be dynamic, with fast-moving actions like hand gestures and interactions with objects [40, 54]. To address these complexities, large-scale datasets like Epic-Kitchens [4, 5], Ego4D [11], and HOI4D [25] have supported progress in tasks such as action recognition [38] and action prediction [31]. However, egocentric videos are not often used to explore 3D scene reconstruction, which is the process of creating a 3D model of the environment from the video footage. This is notable given the growing range of applications that rely on accurate 3D reconstruction [2, 16, 26, 41].

A recent approach that has shown promise for 3D reconstruction is 3D Gaussian Splatting (3DGS) [17], which allows for creating high-quality 3D models. However, 3DGS was originally designed for scenes that are static and recorded from multiple viewpoints. More recent versions of this approach have tried to adapt it for dynamic scenes captured from a single camera (monocular) [23, 47, 50, 51], such as those in egocentric video [54]. However, most of these methods have been tested on third-person (exocentric) videos [9, 30, 33, 48], not egocentric. Although the only egocentric approach, EgoGaussian [54], showed improvement over baselines [47, 50], it is unclear whether the difference stems from the difficulty of egocentric recordings or improvements to the model architecture. As such, it is currently unknown how well existing monocular and dynamic 3DGS methods perform on egocentric videos and whether methods explicitly focused on the egocentric perspective could be useful.

In this work, we explore how well existing dynamic 3DGS methods perform when applied to egocentric video. We use the EgoExo4D dataset [12] that pairs egocentric and exocentric videos of the same scene, allowing us to compare how well models perform in both settings. By analyzing the results, we aim to understand how these models handle the unique challenges of egocentric video. Additionally, we examine how the models perform on static versus dynamic regions, since these areas may present distinct challenges for reconstruction. Finally, as camera motion is a known difficulty in egocentric settings [32, 40], we study its correlation with reconstruction quality. Through this investigation, we seek to highlight the areas where improvements are needed to make these models more effective in real-world applications.

The rest of this document is structured as follows. Chapter 2 introduces the background and concepts required for the understanding of this thesis. Chapter 3 is the main document outlining our approach and findings, structured in the format of a leading computer vision conference.

2
Background

In this chapter, we familiarize the reader with the concepts required for the understanding of Chapter 3. We begin this chapter by discussing the basics: gradient descent, artificial neural networks, and the pinhole camera model.

Understanding these basics will allow us to discuss the problem, and existing approaches, of 3D reconstruction and novel view synthesis (NVS). Here, we cover the concepts of structure from motion (SfM) [29], novel view synthesis, and 3D Gaussian Splatting (3DGS) [17].

As we shall discuss, original 3DGS only works when modeling static scenes. As such, in Section 2.3 we will discuss the approaches for dynamic 3DGS, with a special focus on monocular methods [47, 50, 51].

Finally, in Section 2.4 we will discuss the definition of egocentric video and potential differences from other types of videos. In this section, we will also discuss EgoGaussian [54], which is currently the only publicly-available method focused on egocentric video for the task of dynamic 3D Gaussian Splatting from monocular video.

2.1. Basics

2.1.1. Gradient Descent

The deep learning problems described in this thesis are all *fully supervised*. The word *supervised* means in this case that we are given data X, based on which we attempt to predict given targets y. To this end, we construct a model $\mathscr{F}$, with parameters θ, which outputs predicted targets $\hat{y} = \mathscr{F}(X;\theta)$. We then aim to minimize the error between the predicted targets $\hat{y}$ and the ground-truth targets y. This error is often referred to as the *loss* and is represented with a loss function $\mathscr{L}(y,\hat{y})$. An example of a loss function is the squared error computed as $\mathscr{L}(y,\hat{y}) = \frac{1}{2}\left(y-\hat{y}\right)^2$.

Naturally, the definitions of the model $\mathscr{F}$ and the loss function $\mathscr{L}$ can vary and are task-dependent. However, ideally we would like a single algorithm to optimize the model parameters with respect to the loss function. Although various methods for this task exist [6, 24, 53], gradient descent has established itself as the de-facto standard [34], due to generalizability and efficiency on modern hardware such as GPUs [1].

In gradient descent, the loss function $\mathscr{L}$ must be differentiable [45]. The algorithm then makes use of the local gradient $\frac{\partial \mathscr{L}}{\partial \theta}$ of the loss function with respect to the parameters θ. The algorithm is iterative and the parameters θ_t at an iteration t are updated according to the formula [34]:

$$\theta_{t'} = \theta_t - \alpha \left.\frac{\partial \mathscr{L}}{\partial \theta}\right|_{\theta_t} \tag{2.1}$$

Where α is a hyperparameter called the *learning rate*. The algorithm then typically proceeds until a predefined number of iterations is reached or a desired performance is achieved. While standard gradient descent remains in use, several extensions, such as RMSProp [10] and Adam [19], have been developed to enhance its performance. These methods incorporate additional techniques like momentum [19], which can accelerate convergence or help escape poor local minima, at the cost of increased memory usage [34].

2.1.2. Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) are example deep learning models $\mathscr{F}$ inspired by the workings of natural neurons. The neurons are divided into layers, where all neurons from the current layer are connected to all neurons in the previous layer. An example neural network is visualized in Figure 2.1.

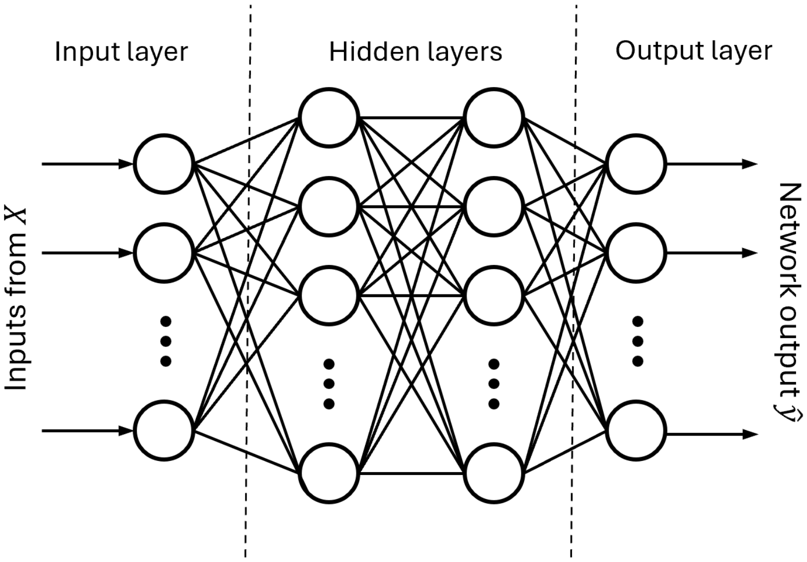


Figure 2.1: An example neural network with an input layer, hidden layers, and the output layer. The network takes samples from X on input and outputs predicted targets $\hat{y}$.

Mathematically, let $\mathbf{z}^{(i-1)} \in \mathbb{R}^n$ be the output of the previous layer $i-1$ with n neurons. The output $\mathbf{z}^{(i)} \in \mathbb{R}^m$ of the current layer i with m neurons is then computed according to the equations:

$$\mathbf{a}^{(i)} = \mathbf{W}^{(i)}\mathbf{z}^{(i-1)} + \mathbf{b}^{(i)} \tag{2.2}$$

$$\mathbf{z}^{(i)} = f\left(\mathbf{a}^{(i)}\right) \tag{2.3}$$

Where $\mathbf{W}^{(i)} \in \mathbb{R}^{m\times n}$ is a matrix containing the weights of the connections between neurons from the previous and current layer, $\mathbf{b}^{(i)} \in \mathbb{R}^m$ is called the *bias*, and f is called the *activation function* and is applied to each item of the input separately. The role of the activation function is to introduce non-linearity to the model and common examples of such functions are `sigmoid`, `ReLU`, and `tanh` [44]. The input to the first layer are the data samples from X and the targets $\hat{y}$ predicted by the network are given by the output of the last layer.

ANNs are currently at the backbone of many modern architectures [3, 7, 27, 42] and multiple versions of these networks exist, such as convolutional neural networks [21], designed for image-based data, and recurrent networks [36], designed for modeling time sequences. The models of interest for this work also often use ANNs directly [47, 50] or rely on models where ANNs are used [54].

2.1.3. The Pinhole Camera Model

The pinhole camera model has been visualized in Figure 2.2. The camera can be described with the following parameters:

- Width W and height H of the camera refer to the dimensions of the image plane.
- Focal length f is the distance between the camera and the image plane. Additionally, the focal lengths in the x and y dimensions (f_x and f_y) might be different to account for non-square pixels.
- Principal point offset (c_x, c_y) specifies the offset of the principal point from the origin point of the image.

The focal lengths f_x and f_y alongside the principal point offsets c_x and c_y are often called the *intrinsics* of the pinhole camera. These parameters are often placed into the following *intrinsic matrix* **K** [14]:

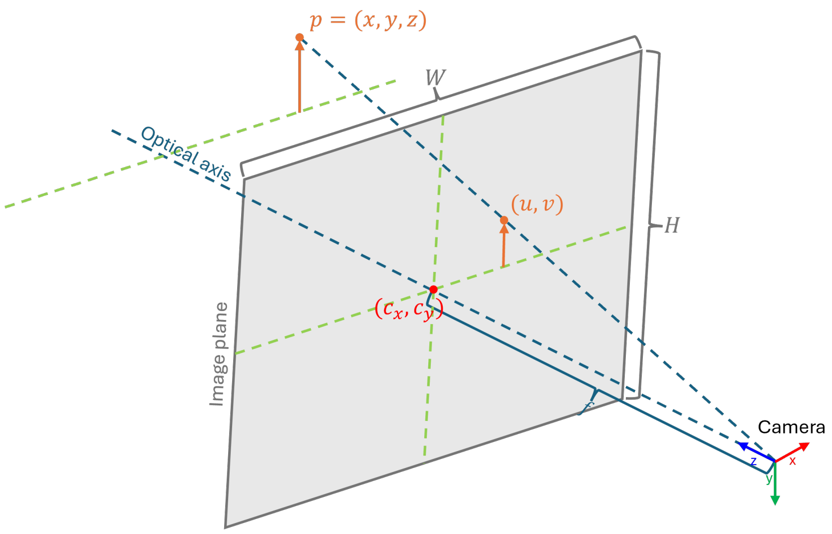


Figure 2.2: A visualization of a simple pinhole camera model. Image inspired by [52].

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.4}$$

The matrix $\mathbf{K}$ can then be used to easily project a point $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ represented in homogenous coordinates $\tilde{\mathbf{p}}$ onto the image plane. This can be done with matrix-vector multiplication as shown in Equation (2.5).

$$\mathbf{K}\tilde{\mathbf{p}} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x x + z c_x \\ f_y y + z c_y \\ z \end{bmatrix} \equiv \begin{bmatrix} f_x \frac{x}{z} + c_x \\ f_y \frac{y}{z} + c_y \\ 1 \end{bmatrix} \tag{2.5}$$

The point $\mathbf{p}$ will thus be projected to the image coordinates $(u, v) = \left(f_x \frac{x}{z} + c_x,\ f_y \frac{y}{z} + c_y\right)$. However, for this projection to work, the position of the point $\mathbf{p}$ must be specified in the camera coordinate system. The camera coordinate system will be different from the world coordinate system if the camera has been translated or rotated. To this end, apart from the intrinsic matrix $\mathbf{K}$, we also define the *extrinsic matrix* $\mathbf{T}$ [14] specifying the rotation and translation of the camera with respect to the world coordinate system. The matrix $\mathbf{T}$ is composed of rotation and translation components as shown in Equation (2.6).

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_x \\ R_{10} & R_{11} & R_{12} & t_y \\ R_{20} & R_{21} & R_{22} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

The matrix $\mathbf{T}$ can either represent the camera-to-world or the world-to-camera transformations. To project the point $\mathbf{p}$ onto the image plane, the matrix $\mathbf{T}$ must represent the world-to-camera transformation. The projection can then be calculated with matrix(-vector) multiplication as $\mathbf{KT}\tilde{\mathbf{p}}$.

2.2. 3D Reconstruction

2.2.1. Structure from Motion (SfM)

Structure from Motion (SfM) aims to derive the 3D structure of a scene given the 2D images of the scene, taken from various viewpoints [29]. The output of such an algorithm are the pinhole camera intrinsics and extrinsics of each viewpoint, alongside selected 3D points representing the scene, derived from the input images. An example, well-established algorithm is COLMAP [35], whose example output can be seen in Figure 2.3.

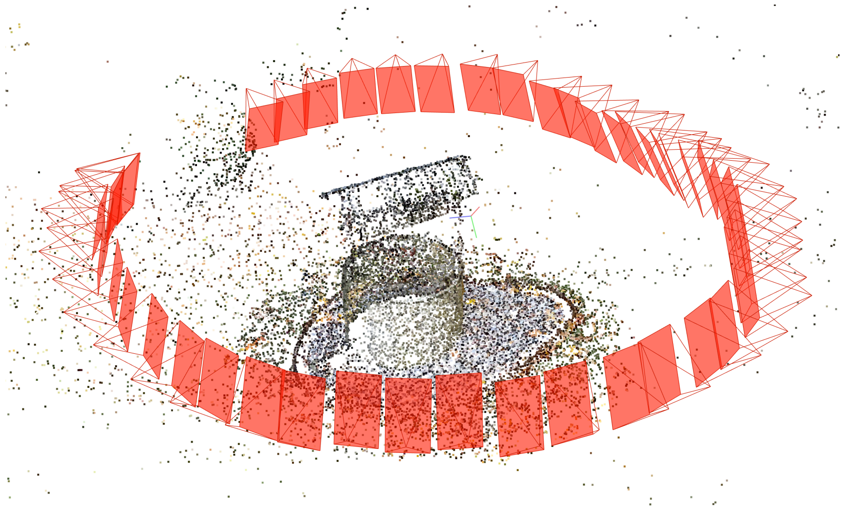


Figure 2.3: Example COLMAP output obtained using COLMAP GUI [35]. Visualized in red are the predicted camera positions. The predicted 3D points correspond to the reconstruction of a well and contain vertex color information.

2.2.2. Novel View Synthesis (NVS)

In novel view synthesis (NVS), the objective is to generate images of a scene as viewed from new, unseen camera positions, given a set of images captured from known viewpoints. Models are typically trained on a set of input images $X = \{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_n\}$, learning to reconstruct the scene from these views. Their performance is then evaluated against a separate set of images $\bar{X} = \{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_k\}$, which depict the scene from novel viewpoints not included in the training set. The quality of the model is measured by its ability to accurately predict these new views, typically via reconstruction error on $\bar{X}$. The reconstruction error is commonly evaluated using perceptual and pixel-level metrics such as peak signal-to-noise ratio (PSNR) [46], structural similarity index measure (SSIM) [43], and learned perceptual image patch similarity (LPIPS) [57], as used in prior work [47, 50, 51, 54]. Below we provide an overview of these three metrics.

Let $\mathbf{I} \in [0,1]^{3\times H\times W}$ be an image from $\bar{X}$ and $\hat{\mathbf{I}} \in [0,1]^{3\times H\times W}$ the corresponding rendering, predicted by a model. Also, let $||\cdot||_2$ indicate the L_2 norm. PSNR can then be calculated according to Equation (2.7) [46].

$$\text{PSNR}\left(\mathbf{I}, \hat{\mathbf{I}}\right) = -10\log_{10}\left(\frac{1}{3WH}\left|\left|\hat{\mathbf{I}} - \mathbf{I}\right|\right|_2^2\right) \tag{2.7}$$

Unlike PSNR, SSIM [43] takes into account luminance, contrast, and structure. The metric splits the input images into sliding windows $\mathbf{A}$ and $\mathbf{B}$. The windows are typically obtained by sliding an 11×11 Gaussian filter across the input images [28]. The SSIM metric for each pair of windows is then computed as shown in Equation (2.8) and Equation (2.9) [28].

$$\text{SSIM}(\mathbf{A}, \mathbf{B}) = l(\mathbf{A}, \mathbf{B})^\alpha \cdot c(\mathbf{A}, \mathbf{B})^\beta \cdot s(\mathbf{A}, \mathbf{B})^\gamma \tag{2.8}$$

$$l(\mathbf{A}, \mathbf{B}) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1} \qquad c(\mathbf{A}, \mathbf{B}) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2} \qquad s(\mathbf{A}, \mathbf{B}) = \frac{\sigma_{AB} + C_3}{\sigma_A\sigma_B + C_3} \tag{2.9}$$

Here, μ_A and μ_B are the mean values of the windows $\mathbf{A}$ and $\mathbf{B}$, σ_A and σ_B are their standard deviations, and σ_{AB} is the covariance between them. The constants C_1, C_2, and C_3 are small stabilizing constants introduced to avoid division by zero; typically, $C_3 = C_2/2$ [28]. The constants α, β, and γ control the importance of individual components and are often set to $\alpha = \beta = \gamma = 1$ [28]. The SSIM over the entire image is obtained by averaging over all windows.

LPIPS [57] compares images by passing them through a pre-trained deep neural network (such as AlexNet [20] or VGG [37]), extracting features from multiple layers. Let $\mathbf{A}$ and $\mathbf{B}$ be patches extracted from the input images $\mathbf{I}$ and $\hat{\mathbf{I}}$. The features at layer l are denoted $\mathbf{f}_l(\mathbf{A})$ and $\mathbf{f}_l(\mathbf{B})$. These features are normalized and compared using a weighted L_2 norm. The LPIPS score is computed as shown in Equation (2.10) [57]:

$$\mathrm{LPIPS}(\mathbf{A},\mathbf{B}) = \sum_{l=0}^{L-1} \mathbf{w}_l \cdot \left|\left|\hat{\mathbf{f}}_l(\mathbf{A}) - \hat{\mathbf{f}}_l(\mathbf{B})\right|\right|_2^2 \tag{2.10}$$

Where $\hat{\mathbf{f}}_l(\cdot)$ denotes normalized features, $\mathbf{w}_l$ are learned weights reflecting the perceptual importance of each layer, and L is the number of extracted layers. A lower LPIPS score indicates greater perceptual similarity between images. In contrast, a higher PSNR or SSIM indicates a higher similarity between images.

Common approaches for novel view synthesis include Neural Radiance Fields (NeRFs) [27], and 3D Gaussian Splatting [17]. In this work, we focus on 3D Gaussian Splatting, which offers similar (or better) rendering quality to NeRF, while providing a significantly faster rendering speed [17]. 3D Gaussian Splatting has been used in practical contexts such as robotics [26], showing its relevance.

2.2.3. 3D Gaussian Splatting (3DGS)

Proposed by Kerbl *et al.* [17], 3D Gaussian Splatting is a recent method for novel view synthesis. The algorithm begins by running a structure from motion algorithm, such as COLMAP [35], on the images from the sets X and $\bar{X}$. The result of this step are the camera intrinsics $\mathbf{K}_j$ and camera extrinsics $\mathbf{T}_j$ for each image, as well as a 3D point cloud defining the 3D structure of the scene.

The 3D point cloud is then used to initialize a set of 3D Gaussian distributions $G_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \sigma_i, \mathbf{c}_i)$ with means $\boldsymbol{\mu}_i$, covariances $\boldsymbol{\Sigma}_i$, opacities σ_i, and view-dependent colors $\mathbf{c}_i$. The covariance matrix $\boldsymbol{\Sigma}_i$ needs to remain positive semi-definite throughout optimization [17], hence, in practice, it is instead represented with scaling and rotation matrices $\mathbf{S}_i$ and $\mathbf{R}_i$ as $\boldsymbol{\Sigma}_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^T\mathbf{R}_i^T$. The scaling matrix $\mathbf{S}_i$ is then represented with a 3D vector $\mathbf{s}_i$ and the rotation matrix $\mathbf{R}_i$ is represented with a 4D quaternion vector $\mathbf{q}_i$. These steps ensure that the representation can be optimized through gradient descent.

For each training viewpoint, the 3D Gaussians are then projected according to the camera parameters using a custom, differentiable renderer [17]. The resulting render $\hat{\mathbf{I}}_j$ is then compared to the ground truth training image $\mathbf{I}_j$ according to the loss function $\mathscr{L}$ as shown in Equation (2.11).

$$\mathscr{L}\left(\mathbf{I}_j, \hat{\mathbf{I}}_j\right) = (1-\lambda)\mathscr{L}_1\left(\mathbf{I}_j, \hat{\mathbf{I}}_j\right) + \lambda\mathscr{L}_{\text{D-SSIM}}\left(\mathbf{I}_j, \hat{\mathbf{I}}_j\right) \tag{2.11}$$

Where λ is a hyperparameter, $\mathscr{L}_1$ is the mean absolute difference between $\mathbf{I}_j$ and $\hat{\mathbf{I}}_j$, and $\mathscr{L}_{\text{D-SSIM}}$ is the SSIM loss between $\mathbf{I}_j$ and $\hat{\mathbf{I}}_j$ [28, 43].

Because different areas of a scene may require varying levels of detail, 3D Gaussian Splatting incorporates adaptive density control [17]. This process dynamically adjusts the number of Gaussians by either introducing new ones or removing those that do not contribute to the rendering. New Gaussians are generated by either splitting high-variance Gaussians or duplicating smaller ones. A Gaussian qualifies for splitting or copying if its spatial gradient exceeds a predefined threshold, indicating its significant contribution to the final image. Conversely, Gaussians that are nearly invisible are discarded, as they are deemed non-contributory.

2.3. Monocular Dynamic 3D Gaussian Splatting

Although original 3DGS performs well for static scenes, it is not designed for dynamic scenes, and hence might produce artifacts such as floaters in the presence of motion [13, 54]. In this section, we present approaches for 3D Gaussian Splatting from dynamic, monocular videos. Monocular videos are special types of videos, where only a single viewpoint is available at each timestep, limiting the amount of multi-view information available [9, 50]. Additionally, the recordings contain dynamic objects, potentially increasing the difficulty of the task [9, 47].

Liang *et al.* [22] provide a comprehensive overview and a comparison of models for monocular, dynamic recordings. Below, we provide brief overviews of three methods we selected for evaluation: Deformable-3DGS [50], 4DGS [47], and RTGS [51].

2.3.1. Deformable-3DGS

In Deformable-3DGS [50], similarly to standard 3DGS [17], the Gaussians are first initialized using the 3D point cloud obtained from SfM. The motion is modeled with an artificial neural network, acting as a global deformation field. To predict the Gaussians at a given timestep t, the network accepts the timestep t and the position $\boldsymbol{\mu}_i$ of each Gaussian G_i as input. The network then predicts the location, scale, and rotation offsets, $\boldsymbol{\delta\mu}_i$, $\boldsymbol{\delta}\mathbf{s}_i$, and $\boldsymbol{\delta}\mathbf{q}_i$. The predicted Gaussian G_i^t at timestep t is then given by $G_i^t = (\boldsymbol{\mu}_i + \boldsymbol{\delta\mu}_i, \mathbf{s}_i + \boldsymbol{\delta}\mathbf{s}_i, \mathbf{r}_i +$

$\boldsymbol{\delta}\mathbf{r}_i, \sigma_i, \mathbf{c}_i)$. All Gaussians are passed through the network, rendered, and the loss function is computed as in Equation (2.11).

During the first 3000 iterations of training, only the Gaussians are optimized, while the deformation field remains fixed. This helps the Gaussians establish stable positions, despite being theoretically incorrect since it cannot model dynamics. Meanwhile, adaptive density control continues as usual throughout training. Additional techniques, such as positional encoding [39, 50] and annealing smooth training [50], are employed to enhance reconstruction quality.

Finally, it should be noted that to obtain final renderings, both the Gaussians and the deformation field are needed. Since the Gaussians are offset by the field at each timestep, the positions, rotations, and scales of the Gaussians are said to be in a *canonical space*. Since, after training, the canonical space can be very different from the real space, rendering the Gaussians on their own usually leads to poor renderings.

2.3.2. 4DGS

4DGS [47] is similar to Deformable-3DGS [50] in that it also predicts the deformation of Gaussians with a single, global deformation field. However, in the case of 4DGS, the global field is modeled with a *HexPlane* $\mathscr{H}$. A HexPlane is a set of six, multi-resolution planes, where each plane attempts to model the relationship between each pair (u, v) of spatio-temporal coordinates $(u, v) \in \{(x, y), (x, z), (x, t), (y, z), (y, t), (z, t)\}$. Each plane $R_l(u, v) \in \mathscr{H}$ is a learnable tensor of shape $h \times lN_u \times lN_v$ with h the hidden dimension, N the basic resolution of the plane, and l the upsampling scale allowing multi-resolution.

The offsets $\boldsymbol{\delta\mu}_i$, $\boldsymbol{\delta}\mathbf{s}_i$, and $\boldsymbol{\delta}\mathbf{q}_i$ of each Gaussian G_i^t are predicted by quering the HexPlane using the Gaussian location $\boldsymbol{\mu}_i = (x_i, y_i, z_i)$ and the current time t. Querying each plane is done by bilinearly interpolating four grid vertices closest to the (u, v) pair. The feature vectors returned by each plane are multiplied and the results from different levels of the planes are concatenated, as shown in Equation (2.12) [47].

$$\begin{gathered}\mathbf{f}_h = \bigcup_l \prod \operatorname{interp}\left(R_l(u, v)\right) \\ (u, v) \in \{(x_i, y_i), (x_i, z_i), (x_i, t), (y_i, z_i), (y_i, t), (z_i, t)\}\end{gathered} \tag{2.12}$$

Where $\mathbf{f}_h \in \mathbb{R}^{lh}$ is the output feature of the HexPlane. Dimensionality reduction is then performed by a small neural network $\mathscr{N}$ to produce a feature vector $\mathbf{f}_d = \mathscr{N}(\mathbf{f}_h)$. Based on the vector $\mathbf{f}_d$, three separate neural networks $\mathscr{N}_\mu$, $\mathscr{N}_s$, and $\mathscr{N}_r$ then predict the offsets $\boldsymbol{\delta\mu}_i$, $\boldsymbol{\delta}\mathbf{s}_i$, and $\boldsymbol{\delta}\mathbf{q}_i$. Similarly to Deformable-3DGS, the predicted Gaussian location is then $G_i^t = (\boldsymbol{\mu}_i + \boldsymbol{\delta\mu}_i, \mathbf{s}_i + \boldsymbol{\delta}\mathbf{s}_i, \mathbf{r}_i + \boldsymbol{\delta}\mathbf{r}_i, \sigma_i, \mathbf{c}_i)$.

Also similarly to Deformable-3DGS, for the first 3000 iterations the Gaussians are optimized without the HexPlane. The loss function does not contain the D-SSIM loss component, but it contains a total-variational loss [8] component $\mathscr{L}_{\text{tv}}$ applied on the HexPlane grids. The total loss is therefore given by Equation (2.13).

$$\mathscr{L}\left(\mathbf{I}_j, \hat{\mathbf{I}}_j, \mathscr{H}\right) = \mathscr{L}_1\left(\mathbf{I}_j, \hat{\mathbf{I}}_j\right) + \mathscr{L}_{\text{tv}}(\mathscr{H}) \tag{2.13}$$

2.3.3. RTGS

RTGS [51], compared to 4DGS [47] and Deformable-3DGS [50], does not model the motion with a global field. Instead, the 3D Gaussians are expanded with an additional temporal dimension. Hence, the mean $\boldsymbol{\mu}_i$ becomes a 4D vector while the covariance matrix $\boldsymbol{\Sigma}_i$ becomes a 4x4 matrix. Although RTGS was shown to perform worse than field-based methods [22], we evaluate it to increase the variety of tested models.

2.4. Exocentric *vs.* Egocentric 3D Gaussian Splatting

Exocentric videos are those where the scene is recorded from a third-person perspective. This is opposed to *egocentric*, where the scene is recorded from a first-person perspective, typically via a head-mounted camera [5, 12, 25]. Existing works focusing on egocentric data, often cite issues such as camera motion [32, 40] and complexity of human actions [13, 54] as difficulties when it comes to understanding egocentric recordings. However, for the task of monocular and dynamic 3DGS specifically, there exist no evaluations that would compare model performance on the same scenes from exocentric and egocentric views. As such, it is unknown how well existing models perform on egocentric data and whether egocentric data indeed poses unique challenges for this task.

Existing methods, such as those outlined above, are tested almost exclusively on non-egocentric data [22, 47, 50, 51]. Currently, EgoGaussian [54], is the only model focused on 3D Gaussian Splatting from egocentric, monocular, and dynamic videos. However, although the baseline evaluation in EgoGaussian [54] contains Deformable-3DGS [50] and 4DGS [47], it does not compare reconstruction quality between egocentric and exocentric views. As such, it remains unknown whether reconstruction from egocentric recordings is indeed more difficult and whether specialized models are required for this task.

Below, we familiarize the reader with EgoGaussian, which we evaluate alongside other baselines.

2.4.1. EgoGaussian

Proposed by Zhang *et al.*, EgoGaussian [54] is a novel method for 3D Gaussian Splatting from dynamic and monocular videos, focused on the egocentric perspective. As with other existing models, the model requires estimated camera poses and a 3D point cloud, obtained with COLMAP [35, 40]. Additionally, however, the model also requires segmentation masks indicating the actor body parts and the manipulated object, which are obtained using EgoHOS [56] and Track Anything [49].

EgoGaussian requires the clips to be manually split into *static* segments, where no object interaction happens, and *dynamic* where the object is being manipulated [54]. We will refer to these segments as *passive* and *active* throughout this work to avoid confusing these terms with *static* and *dynamic* defined as the specific regions of the scene that are (not) moving.

The training procedure then begins by training on the passive clips only. The scene is initialized using traditional 3DGS with the hand and body parts masked out, such that they will not be reconstruted. Using binary object masks obtained previously, the model also selects Gaussians that represent the object that will be interacted with in the active segments.

Next, the active segments are processed. Here, the method attempts to learn the trajectory of the moving object. Importantly, the method explicitly assumes the movement of the object to be rigid, *i.e.* the trajectory of all Gaussians representing the object can be represented with a single translation vector and a single rotation matrix. Finally, the object Gaussians and the static scene Gaussians are optimized together to produce the final, full scene reconstruction.

Especially important for this work are the constraints imposed by EgoGaussian on the input video. The active segments must be located in-between passive segments. Additionally, only a single object may be interacted with and the transformation of the object must be rigid. Finally, the model does not reconstruct body parts of the actor. As a result of these constraints, EgoGaussian cannot be applied to all scenes, which impacts the evaluation performed in our work.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016. 3

[2] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 1

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 4

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision*, pages 720–736, 2018. 1

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 1, 8

[6] Kaelan Donatella, Samuel Duffield, Denis Melanson, Maxwell Aifer, Phoebe Klett, Rajath Salegame, Zach Belateche, Gavin Crooks, Antonio J Martinez, and Patrick J Coles. Scalable thermodynamic second-order optimization. *arXiv preprint arXiv:2502.08603*, 2025. 3

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[8] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 8

[9] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022. 1, 7

[10] Kevin Swersky Geoffrey Hinton, Nitish Srivastava. Overview of mini-batch gradient descent, 2012. 3

[11] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1

[12] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis,

Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 1, 8

[13] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. Egolifter: Open-world 3d segmentation for egocentric perception. In *Proceedings of the European Conference on Computer Vision*, pages 382–400. Springer, 2024. 7, 8

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4, 5

[15] Iason Karakostas, Aikaterini Valakou, Despoina Gavgiotaki, Zinovia Stefanidi, Ioannis Pastaltzidis, Grigorios Tsipouridis, Nikolaos Kilis, Konstantinos C. Apostolakis, Stavroula Ntoa, Nikolaos Dimitriou, George Margetis, and Dimitrios Tzovaras. A real-time wearable ar system for egocentric vision on the edge. *Virtual Reality*, 28(1):44, 2024. 1

[16] Iman Abaspur Kazerouni, Luke Fitzgerald, Gerard Dooly, and Daniel Toal. A survey of state-of-the-art on visual slam. *Expert Systems with Applications*, 205:117734, 2022. 1

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1, 3, 7

[18] Daekyum Kim, Brian Byunghyun Kang, Kyu Bum Kim, Hyungmin Choi, Jeesoo Ha, Kyu-Jin Cho, and Sungho Jo. Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view. *Science Robotics*, 4(26):eaav2949, 2019. 1

[19] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 6

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[22] Yiqing Liang, Mikhail Okunev, Mikaela Angelina Uy, Runfeng Li, Leonidas Guibas, James Tompkin, and Adam W Harley. Monocular dynamic gaussian splatting is fast and brittle but smooth motion helps. *arXiv preprint arXiv:2412.04457*, 2024. 7, 8

[23] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2642–2652. IEEE, 2025. 1

[24] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023. 3

[25] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 20981–20990, New Orleans, LA, USA, 2022. IEEE. 1, 8

[26] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *Proceedings of the European Conference on Computer Vision*, pages 349–366. Springer, 2024. 1, 7

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 4, 7

[28] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim. *arXiv preprint arXiv:2006.13846*, 2020. 6, 7

[29] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 3, 5

[30] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 1

[31] Razvan-George Pasca, Alexey Gavryushin, Muhammad Hamza, Yen-Ling Kuo, Kaichun Mo, Luc Van Gool, Otmar Hilliges, and Xi Wang. Summarize the past to predict the future: Natural language descriptions of context boost multimodal object interaction anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18286–18296, 2024. 1

[32] Suvam Patra, Kartikeya Gupta, Faran Ahmad, Chetan Arora, and Subhashis Banerjee. Ego-slam: A robust monocular slam for egocentric videos. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 31–40, 2019. 1, 8

[33] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1

[34] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 3

[35] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, page 4104–4113, Las Vegas, NV, USA, 2016. IEEE. 5, 6, 7, 9

[36] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020. 4

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[38] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019. 1

[39] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33: 7537–7547, 2020. 8

[40] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Laina, Diane Larlus, Dima Damen, and Andrea Vedaldi. Epic fields: Marrying 3d geometry and video understanding. *Advances in Neural Information Processing Systems*, 36:26485–26500, 2023. 1, 8, 9

[41] Leif Van Holland, Patrick Stotko, Stefan Krumpen, Reinhard Klein, and Michael Weinmann. Efficient 3d reconstruction, streaming and visualization of static and dynamic scene parts for multi-client livetelepresence in large-scale environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4258–4272, 2023. 1

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[43] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6, 7

[44] Wikipedia contributors. Activation function — Wikipedia, the free encyclopedia, 2025. [Online; accessed 9-June-2025]. 4

[45] Wikipedia contributors. Gradient descent — Wikipedia, the free encyclopedia, 2025. [Online; accessed 10-June-2025]. 3

[46] Wikipedia contributors. Peak signal-to-noise ratio — Wikipedia, the free encyclopedia, 2025. [Online; accessed 9-June-2025]. 6

[47] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 1, 3, 4, 6, 7, 8, 9

[48] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 1

[49] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 9

[50] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 1, 3, 4, 6, 7, 8, 9

[51] Zeyu Yang, Zijie Pan, Xiatian Zhu, Li Zhang, Yu-Gang Jiang, and Philip HS Torr. 4d gaussian splatting: Modeling dynamic scenes with native 4d primitives. *arXiv preprint arXiv:2412.20720*, 2024. 1, 3, 6, 7, 8

[52] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021. 5

[53] Zhi-Hui Zhan, Jian-Yu Li, and Jun Zhang. Evolutionary deep learning: A survey. *Neurocomputing*, 483: 42–58, 2022. 3

[54] Daiwei Zhang, Gengyan Li, Jiajie Li, Mickaël Bressieux, Otmar Hilliges, Marc Pollefeys, Luc Van Gool, and Xi Wang. Egogaussian: Dynamic scene understanding from egocentric video with 3d gaussian splatting. *arXiv preprint arXiv:2406.19811*, 2024. 1, 3, 4, 6, 7, 8, 9

[55] Jingzhe Zhang, Lishuo Zhuang, Yang Wang, Yameng Zhou, Yan Meng, and Gang Hua. An egocentric vision based assistive co-robot. In *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, page 1–7, Seattle, WA, 2013. IEEE. 1

[56] Lingzhi Zhang, Shenghao Zhou, Simon Stent, and Jianbo Shi. Fine-grained egocentric hand-object segmentation: Dataset, model, and applications. In *Proceedings of the European Conference on Computer Vision*, pages 127–145. Springer, 2022. 9

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

3

Paper

Bringing a Personal Point of View: Evaluating Dynamic 3D Gaussian Splatting for Egocentric Scene Reconstruction

Jan Warchocki
Delft University of Technology

Abstract

Egocentric video provides a unique view into human perception and interaction, with growing relevance for augmented reality, robotics, and assistive technologies. However, rapid camera motion and complex scene dynamics pose major challenges for 3D reconstruction from this perspective. While 3D Gaussian Splatting (3DGS) has become a state-of-the-art method for efficient, high-quality novel view synthesis, variants, that focus on reconstructing dynamic scenes from monocular video are rarely evaluated on egocentric video. It remains unclear whether existing models generalize to this setting or if egocentric-specific solutions are needed. In this work, we evaluate dynamic monocular 3DGS models on egocentric and exocentric video using paired ego-exo recordings from the EgoExo4D dataset. We find that reconstruction quality is consistently lower in egocentric views. Analysis reveals that the difference in reconstruction quality, measured in peak signal-to-noise ratio, stems from the reconstruction of static, not dynamic, content. Our findings underscore current limitations and motivate the development of egocentric-specific approaches, while also highlighting the value of separately evaluating static and dynamic regions of a video.

1. Introduction

Egocentric, or first-person, video captures the visual input received by an agent acting in the world, such as a human wearing a head-mounted camera. This type of data provides a natural window into how humans perceive and interact with their surroundings, making it especially valuable for applications in augmented reality [15], assistive technology [55], and robotics [18]. Unlike third-person recordings, egocentric video closely reflects the visual input an agent receives while acting in the world. Using egocentric data, we can improve an agent's ability to understand the world from a first-person perspective, leading to more intuitive and efficient interactions in real-world scenarios [55].

Egocentric data presents unique challenges. The camera is subject to rapid, often unpredictable motion driven by head or body movement [34, 44, 45]. At the same time, the scenes themselves are highly dynamic, frequently involving complex hand-object interactions [12, 45, 54]. To address these challenges, large-scale egocentric datasets such as EPIC-Kitchens [2, 3], HOI4D [26], and Ego4D [10] have driven advances in tasks like action recognition [43], action prediction [33], and hand-object segmentation [56]. Yet, one important direction remains largely underexplored: 3D reconstruction and novel view synthesis in egocentric settings. Tackling this gap is key to enabling more comprehensive spatial understanding and unlocking immersive applications such as augmented and virtual reality [12].

3D Gaussian Splatting (3DGS) [17] has recently emerged as a state-of-the-art approach for high-quality and efficient 3D reconstruction and novel view synthesis with use cases in fields such as robotics [27]. Extensions have been proposed to the 3DGS framework to allow the handling of dynamic scenes viewed from a single camera (monocular) [6, 7, 23, 49, 51, 52, 54]. These methods, however, are commonly only evaluated on scenes filmed from a third-person, exocentric perspective, rather than from the egocentric perspective [8, 22, 31, 35, 50]. Hence, it remains unclear how well these models perform when applied to egocentric video, and whether models specialized on the egocentric perspective, are needed.

To our knowledge, EgoGaussian [54] is the only existing dynamic monocular 3DGS model specifically targeting egocentric vision. While EgoGaussian has demonstrated improved rendering quality over monocular baselines [49, 51] on egocentric data, the evaluation is limited to egocentric recordings alone. It remains unclear whether the observed improvements stem from the model design, the data itself, or from other factors.

In this work, we aim to address this research gap and answer the question of how well existing monocular dynamic 3D Gaussian Splatting models perform when applied to egocentric data. To this end, we compare the existing models on paired ego and exo perspective recordings of the same scene from the EgoExo4D dataset [11]. Because dynamic regions may be crucial in practical applications [14], we compare the performance of the models on the static and dynamic regions separately. Since rapid camera motion is often cited as a challenge of egocentric vision [34, 45], we investigate if it correlates with reconstruction quality. Overall, our main contributions can be summarized as follows:

1. We compare the performance of four existing dynamic 3DGS models on paired ego-exo scenes from the EgoExo4D dataset.
2. We propose an evaluation protocol that compares the performance of existing models on static and dynamic regions of the scene.
3. We propose methods to study the correlation between camera motion and 3D reconstruction quality in egocentric settings.
4. Our results suggest the need for egocentric-specific approaches, while also showing that future evaluations of methods could benefit from evaluating the static and dynamic regions of the scenes separately.

Our code and data have been made available at https://github.com/Jaswar/evaluation-thesis.

2. Related work

Egocentric vision. Egocentric, or first-person vision, focuses on capturing visual data from the viewpoint of a wearable or head-mounted camera. This special type of vision has recently garnered attention due to its importance for applications such as augmented reality [15] and robotics [18, 55]. Although egocentric video is considered difficult due to issues such as the complexity of human actions [12, 53] and varied camera motion [34, 45], egocentric datasets such as Epic-Kitchens [2, 3] have helped advancements in fields such as video understanding and human-object interaction [4, 33]. In this work, we compare egocentric recordings to other types of recordings to verify whether the challenges posed by egocentric vision impact the performance of monocular 3D Gaussian Splatting methods.

3D Gaussian Splatting for dynamic scenes. 3D Gaussian Splatting (3DGS) [17] has recently emerged as a promising method for novel view synthesis of static scenes, outperforming existing NeRF-based approaches [1, 29] in both rendering quality and speed [17]. 3DGS assumes the scene to be static, leading to artefacts, such as floaters, in the reconstruction of dynamic scenes [12, 54]. As such, special methods have been proposed to handle scene motion [6, 9, 28]. Of these, monocular methods [7, 23, 25, 30, 49, 51, 52], which require only a single camera, are especially interesting [22]. In this work, we focus on evaluating monocular models in egocentric dynamic scenes.

Evaluation of existing monocular models. Various datasets are used for the evaluation of models for 3D Gaussian Splatting from monocular videos with dynamics [8, 22, 31, 32, 35, 50]. D-Nerf [35] contains synthetic scenes captured with a rapidly moving camera without motion blur. Nerfies [31], HyperNerf [32], and DyCheck [8] all contain real-world recordings of kitchen activities, animals, and other moving objects. However, even the scenes involving human actors are not recorded from an egocentric perspective, but rather from an exocentric, third-person perspective. As such, it is currently unknown how well existing monocular 3DGS models perform when the recording is captured from an egocentric point of view and whether these models perform better or worse than with other types of recordings. In this work, we provide such an evaluation.

EgoGaussian. To the best of our knowledge, EgoGaussian [54] is the only publicly-available model for monocular 3DGS reconstruction of dynamic scenes from an egocentric perspective. The model requires each clip to be manually split into passive (no interaction) and active (object manipulation) segments[1]. Using provided object masks, passive segments are then used to initialize the background and object shape, while active segments are used to refine both and estimate the object's pose. The method assumes fully rigid object motion and does not model the actor. We compare its reconstruction quality to monocular models not tailored to egocentric settings.

Concurrent to our work, DeGauss [47] has emerged as an alternative model for dynamic 3DGS focused on the egocentric perspective. Rather than relying on provided object masks, the method attempts to segment dynamics via a learned mask. The method also does not rely on the split between passive and active segments and is not constrained to rigid motion. Although promising, the model does not currently have a publicly-available implementation and is hence excluded from evaluation.

Static and dynamic modeling. Modeling dynamic objects is crucial for practical applications [14]. Although the majority of dynamic and monocular 3DGS methods model static and dynamic regions of the video together [6, 7, 49, 51, 52], methods exist where the static and dynamic are being modeled separately [23, 54]. In [22] the authors show, however, that existing methods reconstruct the static and dynamic regions similarly. Since the evaluation in [22] focuses on non-egocentric data, we perform a similar analysis in egocentric settings. In this way, we aim to show whether future models could benefit from modeling the static and dynamic regions separately.

Simultaneous Localization and Mapping (SLAM). SLAM methods estimate an agent's trajectory (localization) while building a map of the environment (mapping) [16]. Unlike 3DGS, which typically assumes known camera poses obtained offline, SLAM estimates poses online and

[1] Referred to as *static* and *dynamic* in the original paper; we adopt different terms to avoid confusion.

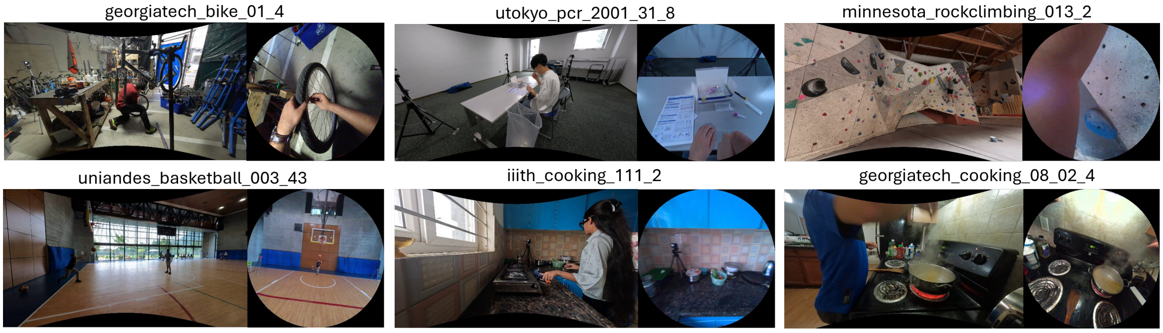

georgiatech_bike_01_4
utokyo_pcr_2001_31_8
minnesota_rockclimbing_013_2
uniandes_basketball_003_43
iiith_cooking_111_2
georgiatech_cooking_08_02_4

Figure 1. First frame of selected scenes from the exo (left, first exo camera) and ego (right) views. First 4 scenes are random; last 2 are selected EgoGaussian-style scenes. Frames shown after undistortion. As we can see, the scenes are varied. Best viewed zoomed in.

often runs in real-time. Extensions have been proposed to handle dynamic environments [19, 46, 58] while Ego-SLAM [34] showed the need for egocentric-specific approaches. While this work focuses on 3DGS, recent research integrates 3DGS into SLAM pipelines [19, 58], suggesting our findings could also inform SLAM. Separately, fast view synthesis methods like FaDIV-Syn [38] offer an alternative by generating novel views without full 3D reconstruction, differing from the approaches studied here.

3. Experiments

3.1. Ego vs. exo

In this experimental section, we aim to answer whether reconstruction from egocentric data is indeed different from other types of data. To this end, we compare the performance of existing models on egocentric data against the performance on exocentric, third-person data. We choose to compare against exocentric recordings as they form a natural opposite of egocentric recordings.

Dataset. EgoExo4D [11], contains 1,286 hours of paired egocentric and exocentric recordings of skilled human activities. We choose it over other ego-exo datasets [13, 20, 21, 36, 39, 41, 42] due to its scene diversity and availability of ground truth camera intrinsics, extrinsics, and a semi-dense 3D point cloud for initializing Gaussians. These ground truth parameters are crucial for isolating baseline performance from errors introduced by structure-from-motion methods such as COLMAP [40, 54].

We select 8 random clips, each exactly 300 frames (10 seconds) long, which approximately corresponds to the length of existing dynamic clips for the task of monocular 3DGS of dynamic scenes [54]. To evaluate EgoGaussian [54], we manually select 2 additional clips that contain rigid motion and are split into passive and active segments, matching the requirements of EgoGaussian. First frames from both exo and ego views of selected scenes are presented in Figure 1. As we can see, the clips contain diverse scenarios. The exo cameras tend to capture more depth information about the scene and the exo cameras are static. These differences may cause the scene to be reconstructed with a different quality from the ego and exo perspectives. All 10 scenes are visualized in Appendix C.

Models. Apart from EgoGaussian, we select three other baseline methods for monocular 3D Gaussian Splatting of dynamics scenes. The baselines are: Deformable-3DGS [51], 4DGS [49], and RTGS [52]. We select Deformable-3DGS and 4DGS due to their strong performance on dynamic scenes as shown by [22]. Since both Deformable-3DGS and 4DGS define the motion with a global field [22, 49, 51], we include RTGS for model variety, as it does not rely on an explicit motion field [22, 52].

Data preprocessing. The ego and exo cameras in Ego-Exo4D are fisheye and incompatible with the standard 3D Gaussian Splatting framework [11, 17, 24]. We address this by undistorting frames using known intrinsics, following the official guidelines [48]. This process introduces artifacts, producing black regions at the top and bottom of exo images [48], as shown in Figure 1. To filter invalid pixels, we undistort a binary mask alongside the images and use it within the 3D Gaussian Splatting pipeline.

We manually provide the split into passive and active segments for EgoGaussian. Furthermore, we obtain the object and actor masks by annotating frames using Segment Anything 2 [37].

Evaluation protocol. We compare model performance on time-synchronized ego and exo recordings of the same scene. For ego views, the training set uses even-indexed frames, validation uses frames where index $i \equiv 1 \pmod{4}$, and test where $i \equiv 3 \pmod{4}$. This mirrors EgoGaussian [54], but with a validation split added for per-scene hyperparameter tuning.

Since exo cameras are static, a single-view recording lacks sufficient multi-view information for training 3DGS models [59]. Instead, we generate the sequence by randomly selecting a viewpoint at each index i. This ensures

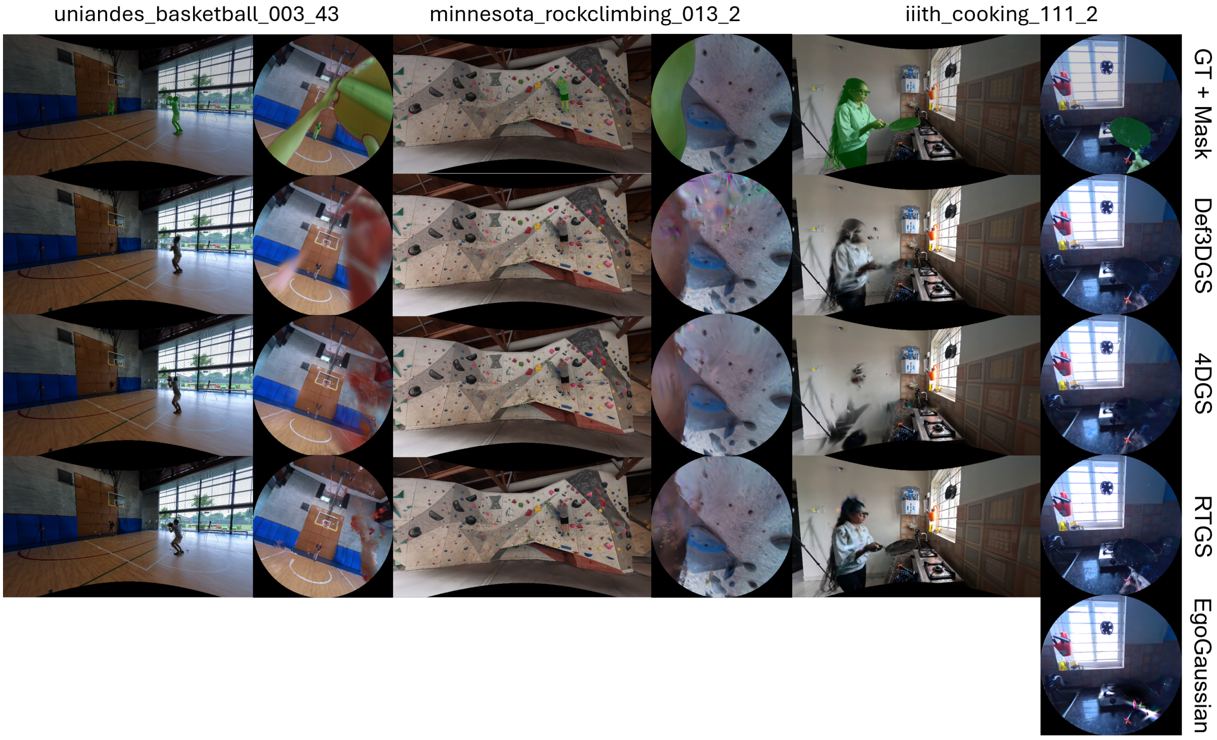


Figure 2. Ground truths and model renderings on 2 random and 1 EgoGaussian scene. Top row shows ground truth with dynamic masks overlaid in green. Models consistently reconstruct dynamic regions with lower visual fidelity. Best viewed zoomed in.

		Random scenes			EgoGaussian scenes		
Model	**View**	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓
EgoGaussian	Ego	-	-	-	27.46 ± 0.27	0.78 ± 0.002	0.31 ± 0.003
Def3DGS	Ego	30.42 ± 0.06	0.82 ± 0.004	0.27 ± 0.006	29.95 ± 0.03	0.82 ± 0.000	0.25 ± 0.000
	Exo	36.06 ± 0.04	0.97 ± 0.000	0.11 ± 0.000	31.85 ± 0.11	0.93 ± 0.000	0.17 ± 0.000
4DGS	Ego	29.75 ± 0.03	0.80 ± 0.001	0.32 ± 0.001	29.61 ± 0.05	0.80 ± 0.001	0.30 ± 0.002
	Exo	33.23 ± 0.18	0.92 ± 0.002	0.21 ± 0.003	30.47 ± 0.09	0.90 ± 0.004	0.22 ± 0.006
RTGS	Ego	30.03 ± 0.02	0.81 ± 0.001	0.30 ± 0.000	29.50 ± 0.01	0.81 ± 0.000	0.28 ± 0.002
	Exo	32.64 ± 0.25	0.93 ± 0.006	0.17 ± 0.008	29.03 ± 0.54	0.89 ± 0.008	0.22 ± 0.015

Table 1. Performance of selected models on paired ego/exo views from 8 random scenes and 2 EgoGaussian-style scenes [54]. Results are averaged over 3 runs (± std). Grey indicates whether ego or exo scored higher. As we can observe, models perform noticeably better on exo views.

monocular input while preserving 3D cues, and is similar to existing setups [31, 32, 35]. We apply the same train, validation, and test split as in the egocentric case.

Evaluation metrics. Following [22], we report masked peak signal-to-noise ratio (mPSNR), masked structural similarity index measure (mSSIM), and masked learned perceptual image patch similarity (mLPIPS). For the eight random scenes, the mask corresponds to the undistorted binary mask defined earlier. For the EgoGaussian scenes, the mask further excludes the actor, as it is not within the reconstruction scope of EgoGaussian [54]. All metrics are computed on the test set. Each model is re-trained 3 times per scene using the same hyperparameters.

Hyperparameters. We perform a random search to select per-scene hyperparameters for Deformable-3DGS, 4DGS, and RTGS. For 4DGS and RTGS, the search space includes parameter values from existing configurations. As Deformable-3DGS lacks such configurations, we instead search over the width and depth of the deformation network, as well as the total number of iterations. The exact search parameters for all models are provided in Appendix A.

The search runs for a fixed duration of 4 hours per scene and per model. Each configuration is evaluated on the validation set, and the one with the highest masked PSNR is selected. All searches are performed on an NVIDIA A40 GPU located on the Delft AI Cluster [5]. Training EgoGaussian on each scene exceeded 4 hours, hence no hyperparameter search was conducted for that model.

Results. The results of this experiment are presented in Table 1, where the metrics are averaged over the 8 random and 2 EgoGaussian scenes separately. As we can observe, the models almost always perform better on the exocentric recordings with very low variance in results. The only exception is the mPSNR score measured for the RTGS model on the EgoGaussian scenes. However, other metrics in this setting are still better for exocentric recordings. This difference in reconstruction quality between ego and exo suggests that the reconstruction from egocentric perspective is more difficult for existing models on average.

The gap in mPSNR performance between ego and exo appears larger in the random scenes than the EgoGaussian scenes. However, EgoGaussian clips consist of passive segments where no object interaction happens. Additionally, the hands are excluded from egocentric metric calculation and the object movement is fully rigid. These differences may cause the scene dynamics to be easier to reconstruct from the egocentric perspective, which could explain the mPSNR difference between ego and exo being smaller than in random scenes. This observation suggests that EgoGaussian clips may not be fully representative of models' performance on egocentric video.

Comparing EgoGaussian to other models, we observe that it obtains worse reconstruction quality across all metrics. This is a surprising result, as in the original paper, EgoGaussian was shown to outperform both Deformable-3DGS and 4DGS on egocentric data [54].

Example renderings are presented in Figure 2. As we can see from the first two, random sequences, the overall reconstruction from the ego perspective is visibly worse than the exo view. In the basketball sequence, the difference is most visible due to the poor reconstruction of the basketball and the hands of the actor. Although the qualitative results reinforce the conclusion that the reconstruction from the egocentric perspective is more difficult, it is yet unclear whether this difference comes from the reconstruction of dynamic or static objects.

3.2. Dynamic vs. static

Accurate dynamic modeling is essential for practical applications involving dynamic 3DGS [14]. Additionally, dynamic objects pose different challenges than static objects and some methods model them separately [23, 54]. In this section, we thus aim to answer whether current methods model static and dynamic regions with the same accuracy.

Dynamic masks. We use Segment Anything 2 [37] to manually annotate each dynamic object in selected clips. The mask for a given object at a given frame i is only considered dynamic if the object visibly moved between frames i and $i-1$. Example resulting dynamic masks have been overlaid in green in Figure 2.

Evaluation metrics. Similarly to the previous section, we use mPSNR, mSSIM, and mLPIPS to evaluate the models. The dynamic mask corresponds to the combined masks of all dynamic objects at the given frame. The static mask contains only the background, *i.e.* all objects that are not currently moving. The static and dynamic masks are combined with the undistortion masks obtained previously to ensure only valid pixels are evaluated.

Results. The results for the dynamic and static masks are presented in Table 2. Firstly, as we can see, the model variance remains low on both dynamic and static parts of the scene. Secondly, we observe that it is unclear whether the reconstruction of dynamics is easier from the ego or exocentric views. In random scenes, the dynamic mPSNR for Deformable-3DGS and 4DGS is higher for the egocentric view. The egocentric mPSNR for RTGS is also much closer to the exo perspective than in Table 1. At the same time, the static reconstruction is again of higher quality in terms of mPSNR in exocentric views. Both mSSIM and mLPIPS are better in the exocentric view on static and dynamic regions. Hence, these results suggest that the gap in reconstruction quality between ego and exo in terms of mPSNR comes from the reconstruction of static regions, not dynamic.

In the EgoGaussian scenes, we observe that the dynamic reconstruction is almost always better in the egocentric case. This reinforces the previous hypothesis that the dynamics in EgoGaussian videos are easier to reconstruct in the egocentric view. Additionally, this further suggests that the EgoGaussian clips may not form a representative sample of real videos.

Furthermore, the difference in performance between static and dynamic reconstruction is clearly visible both from Table 2 as well as Figure 2. Across both egocentric and exocentric recordings, the models perform better when reconstructing the static regions of the scene. This highlights a key limitation of current methods in handling motion and suggests that future work should focus on explicitly improving dynamic scene understanding.

As with the previous results, EgoGaussian again performs worse than other baselines. The performance is worse in both the static and dynamic regions. This is again unexpected considering the original results [54].

3.3. EgoGaussian vs. others

As shown in Tables 1 and 2, our results for EgoGaussian differ from the original paper, where it outperformed 4DGS and Deformable-3DGS both quantitatively and qualitatively [54]. To validate our pipeline, we re-evaluate EgoGaussian and the baselines on the original EgoGaussian data from Epic-Kitchens [2, 3] and HOI4D [26].

Evaluation protocol. We maintain the evaluation protocol from EgoGaussian and hence only split the data into a train and a test set. It is unknown which hyperparameters were used for the baselines in the EgoGaussian paper [54], hence we use default configurations. Both for the baselines and EgoGaussian we do not measure the reconstruction quality of body parts. Therefore, we measure masked PSNR, SSIM, and LPIPS.

Furthermore, it should be noted that in the official

		Dynamic masks						Static masks					
		Random scenes			EgoGaussian scenes			Random scenes			EgoGaussian scenes		
Model	**View**	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓
EgoGaussian	Ego	-	-	-	21.65 ± 1.03	0.59 ± 0.050	0.41 ± 0.047	-	-	-	27.94 ± 0.10	0.79 ± 0.001	0.30 ± 0.002
Def3DGS	Ego	25.47 ± 0.05	0.65 ± 0.003	0.38 ± 0.004	24.58 ± 0.28	0.63 ± 0.010	0.36 ± 0.010	31.17 ± 0.06	0.83 ± 0.004	0.27 ± 0.007	30.23 ± 0.04	0.82 ± 0.000	0.25 ± 0.000
	Exo	24.21 ± 0.10	0.70 ± 0.004	0.27 ± 0.002	22.70 ± 0.22	0.66 ± 0.004	0.37 ± 0.006	38.85 ± 0.01	0.97 ± 0.000	0.11 ± 0.000	34.85 ± 0.08	0.96 ± 0.000	0.14 ± 0.000
4DGS	Ego	24.56 ± 0.14	0.62 ± 0.004	0.42 ± 0.004	25.41 ± 0.26	0.66 ± 0.006	0.35 ± 0.008	30.60 ± 0.05	0.80 ± 0.001	0.31 ± 0.001	29.88 ± 0.05	0.81 ± 0.001	0.30 ± 0.002
	Exo	24.05 ± 0.13	0.71 ± 0.005	0.28 ± 0.004	22.72 ± 0.39	0.64 ± 0.011	0.40 ± 0.010	34.56 ± 0.20	0.92 ± 0.002	0.21 ± 0.003	32.45 ± 0.30	0.92 ± 0.005	0.20 ± 0.007
RTGS	Ego	25.33 ± 0.05	0.65 ± 0.001	0.38 ± 0.002	26.29 ± 0.15	0.67 ± 0.010	0.32 ± 0.006	30.82 ± 0.02	0.82 ± 0.001	0.30 ± 0.000	29.67 ± 0.01	0.82 ± 0.000	0.28 ± 0.002
	Exo	25.60 ± 0.43	0.75 ± 0.009	0.21 ± 0.008	23.42 ± 0.12	0.66 ± 0.003	0.34 ± 0.010	33.58 ± 0.24	0.93 ± 0.006	0.17 ± 0.008	30.45 ± 0.70	0.91 ± 0.010	0.20 ± 0.017

Table 2. Performance of selected models on dynamic and static masks for paired ego/exo views from 8 random scenes and 2 EgoGaussian-style scenes [54]. Results are averaged over 3 runs (± std). Grey indicates whether ego or exo view performs better. mPSNR is similar between ego/exo for dynamic masks; exo remains easier for static.

	Epic-Kitchens						HOI4D					
	Passive segments			Active segments			Passive segments			Active segments		
Model	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓
EgoGaussian (original)	28.33	0.85	0.19	28.34	0.88	0.17	30.99	0.96	0.08	30.33	0.95	0.09
EgoGaussian (ours*)	28.76	0.86	0.18	30.55	0.89	0.15	30.52	0.96	0.09	31.12	0.96	0.09
EgoGaussian (ours†)	28.61	0.85	0.25	30.35	0.88	0.22	30.43	0.95	0.13	30.97	0.95	0.13
Def3DGS (original)	27.63	0.86	0.17	23.27	0.82	0.21	28.09	0.94	0.08	26.92	0.94	0.10
Def3DGS (ours†)	37.54	0.96	0.12	32.94	0.94	0.16	34.38	0.97	0.08	33.06	0.96	0.10
4DGS (original)	28.90	0.87	0.16	23.13	0.80	0.23	28.69	0.94	0.08	27.33	0.94	0.10
4DGS (ours†)	34.40	0.93	0.18	29.61	0.89	0.23	36.73	0.97	0.09	35.13	0.97	0.11

Table 3. The performance of EgoGaussian and baselines on the original EgoGaussian Epic-Kitchens and HOI4D data [3, 26, 54]. Ours* uses original metrics; ours† uses corrected ones. Grey indicates the best score across ours† runs. While ours* matches the original, baselines perform better than EgoGaussian, contrary to prior findings.

EgoGaussian evaluation, the masked out areas are zeroed, rather than ignored [54], which will lead to biased metric values. Additionally, EgoGaussian does not normalize input images to the $[-1, 1]$ range, which is necessary for LPIPS [57]. We report EgoGaussian results with (ours†) and without (ours*) correction in metrics.

Results. The results are presented in Table 3. As we can observe, EgoGaussian without metric changes (ours*) closely matches the original paper. However, the baselines perform much better than originally reported. Indeed, when comparing with updated metric calculations (ours†), the baselines tend to outperform EgoGaussian itself. These results therefore reinforce the findings from the previous sections and show that the sudden difference in performance does not come from the data or the lack of hyperparameter tuning for EgoGaussian.

3.4. Effects of camera motion

Rapid and often unpredictable camera motion, caused by head or body movement, is an inherent property of egocentric data [45]. Thus, understanding the impact or correlation of camera motion on reconstruction quality is essential. In this section, we aim to answer whether egocentric camera motion correlates with the reconstruction quality.

Definition of camera motion. We measure camera motion in two aspects: camera velocity, defined as the speed of the camera between frames, and camera baseline, defined as the distance traveled by the camera. An increase in camera velocity causes the egocentric test camera poses to be further away from training poses, which may influence reconstruction quality. An increase in the camera baseline might provide more multi-view information, which could increase the quality [8, 22]. We note that although metrics have been proposed to measure the amount of multi-view information in a monocular setting [8], these metrics are either difficult to compute in practice [8, 22], or were shown not to correspond well to actual reconstruction quality [22].

3.4.1 Camera velocity

Evaluation protocol. Let $\mathbf{v}_t \in \mathbb{R}^3$ be the camera linear velocity between time steps t and $t-1$ stemming from camera translation. The maximal components of the velocity at a given time step t are then computed as $\hat{v}_t = \max_{1 \leq i \leq 3} |\mathbf{v}_{ti}|$. Since the range of velocities varies in each scene, we normalize them to the $[0, 1]$ range per scene. The velocities are plotted on a logarithmic scale for readability. Hence, the final linear velocities $\bar{v}_t$ used for analysis are computed according to the equation:

$$\bar{v}_t = \ln\left(\frac{\hat{v}_t - \min_t(\hat{v}_t)}{\max_t(\hat{v}_t) - \min_t(\hat{v}_t)}\right) \tag{1}$$

We then plot the linear velocities at time step t against mLPIPS achieved by the model on the static part of the scene at the same time step. We choose to evaluate against LPIPS as it was shown to correspond with human perception better than PSNR and SSIM [57]. Since mLPIPS also varies per scene, we normalize it to the $[0,1]$ range, similarly to the velocity. Only test frames are evaluated.

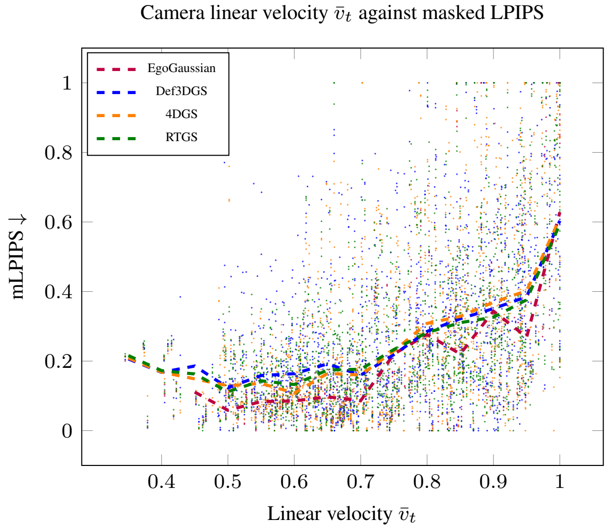


Figure 3. Camera linear velocity $\bar{v}_t$ plotted against mLPIPS. Additional trend lines are plotted. As we can observe, as linear velocity increases, mLPIPS increases, which corresponds to worse reconstruction quality.

Results. The resulting scatter plot for linear velocity is presented in Figure 3. Additionally, we plot trend lines by averaging the mLPIPS over buckets of size 0.05. As we can observe from both the scatter plot and the trend lines, as camera velocity increases, the average masked mLPIPS increases, which indicates worse reconstruction. This suggests that, contrary to some prior expectations [8], increased camera movement in egocentric video does not always yield better reconstructions, and may in fact hinder performance. Further analysis in Appendix D shows that the same correlation holds for increased camera rotation and when measuring mPSNR and mSSIM instead of mLPIPS.

Due to the high variance observed in the scatter plot, one might question whether the observed positive correlation is statistically significant. To this end, we measure the Pearson and Spearman coefficients. Both indicate a positive correlation of around 0.5 at a p-value, with the null hypothesis that the correlation is 0, far below 0.05. Appendix D contains detailed results of the significance tests.

3.4.2 Camera baseline

Evaluation protocol. Liang *et al.* [22] build a synthetic monocular dataset where the camera moves along an arch. The camera baseline is then defined as the distance between the start and end points of the camera. Since the trajectory in egocentric videos is more complex, we instead define the camera *linear* baseline as the maximal distance between any two points alongside the camera trajectory. We compute the camera linear baseline per scene and report the mean mLPIPS over the test frames of the given sequence. EgoGaussian is not evaluated due to it only being tested on two scenes.

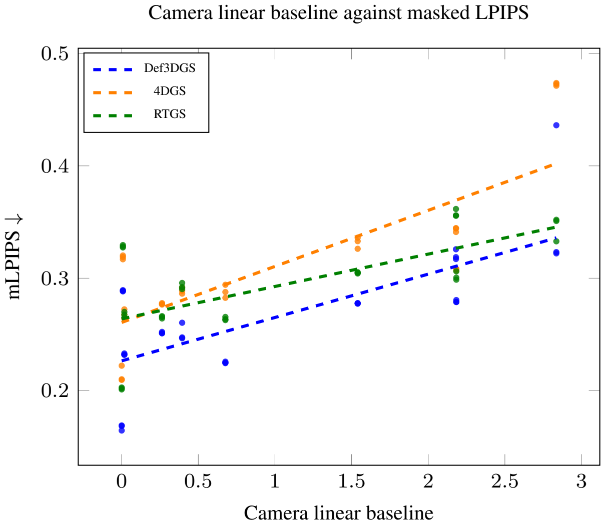


Figure 4. Camera linear baseline plotted against mLPIPS. Additional linear regression models fitted are shown. As we can observe, as camera baseline increases, mLPIPS increases, which corresponds to worse reconstruction quality.

Results. The results are shown in Figure 4 where additional best-fit linear regression models are shown. As we can observe, as camera baseline increases, the reconstruction quality worsens. As with the previous results, both Pearson and Spearman coefficients show a positive correlation with a p-value of below 0.05. Together with the velocity analysis, these findings reinforce that increased egocentric camera motion, whether rapid or spatially extensive, tends to degrade reconstruction quality, challenging assumptions from prior work [8, 22].

Similarly to camera velocity, we can define a baseline based on angular motion, which we refer to as the *angular* baseline. Appendix D presents this additional analysis, showing that the same negative correlation holds, and includes detailed significance test results.

4. Discussion

Limitations. One might question the fairness of comparing egocentric and exocentric recordings due to inherent differences in aspects such as camera motion and multi-view coverage. Egocentric video is recorded from a moving, first-person perspective, while exocentric footage relies on multiple static cameras positioned around the scene. However, these differences are not artifacts of the evaluation setup but fundamental properties of each modality. Fixed-camera, multi-view exocentric recordings are standard in ego-exo datasets [11, 13, 20, 21, 36, 39, 41], and reflect how such data is typically collected in practice. Therefore, while the two modalities differ substantially, comparing them still offers meaningful insights into model performance under realistic and representative conditions.

Due to the static exo cameras, a limitation of our current setup is that the exocentric camera poses used for testing are also seen during training, but at different timesteps. This allows the models to potentially memorize specific viewpoints, rather than generalize to novel views, effectively reducing the difficulty of the exo task. In contrast, the egocentric setting naturally enforces both temporal and spatial generalization due to the moving camera. While this introduces an asymmetry in the evaluation, it reflects a practical constraint of working with existing ego-exo datasets, where only a limited number of static cameras is available [11, 13, 20, 21, 36, 39, 41, 42]. Future datasets could address this issue through denser exo coverage or the use of moving exocentric cameras.

Our evaluation is limited to 10 scenes due to the computational cost of training dynamic 3D Gaussian Splatting models, which often requires several hours of optimization per scene per method. This evaluation scale is in line with common practice in dynamic scene reconstruction, where prior datasets such as D-NeRF (8 scenes) [35], Nerfies (4 scenes) [31], HyperNeRF (17 scenes) [32], NeRF-DS (7 scenes) [50], and DyCheck (14 scenes) [8] have been used for benchmarking. While a limited number of scenes might make the reported numbers sensitive to outliers, we address this in Appendix B by including per-scene comparisons between egocentric and exocentric views, showing that our main conclusions hold consistently across scenes.

Egocentric difficulty. Results from Table 1 suggest that reconstruction from the egocentric perspective is more challenging for existing models than from the corresponding exocentric views. Interestingly, as shown in Table 2, the difficulty of reconstructing from the ego perspective appears to stem from the static regions of the scene, with dynamic regions reconstructed at comparable mean mPSNR across both modalities. While these findings should be interpreted in light of the evaluation asymmetry, where exocentric test views coincide with training camera poses, this trend is further reinforced by the camera motion results.

The camera motion results showed a negative correlation between camera motion in egocentric videos and reconstruction quality. This contrasts with prior claims in non-egocentric settings [8, 22]. Hence, these observations reinforce the finding that reconstruction from an egocentric perspective presents distinct challenges compared to other types of data. Importantly, camera motion is unlikely to be the sole factor affecting quality. Its correlation with reconstruction performance may also reflect underlying influences such as body movement.

Dynamic difficulty. The results from Table 2 show that the reconstruction of dynamics is worse than the reconstruction of the static background. Existing methods reconstruct the static background with a higher visual fidelity than the dynamic objects. These results may appear surprising, as in [22], the authors found that the performance of dynamic Gaussian methods ‘does not change much‘ after masking out the static components. Our results thus show that it is still worthwhile to evaluate static and dynamic regions separately when developing future models. Likewise, it might be beneficial to model static and dynamic regions separately, such as in [23] or [54].

Future work. Our results suggest that egocentric reconstruction is more challenging than exocentric, but due to limited static exo cameras, spatial generalization in exocentric evaluation is restricted. This asymmetry means the ego-exo comparison should be interpreted cautiously. Still, reinforced by the negative correlation of egocentric camera motion to reconstruction quality, the findings highlight the need for egocentric-specific models. Future models should also not neglect the reconstruction of static regions, as they may contribute to the difference between ego and exo reconstruction. Contrary to prior work [22], we find that dynamic regions are reconstructed less accurately than static ones, revealing a key blind spot in current models. Future benchmarks should therefore separate static and dynamic evaluation. Finally, future datasets with denser or moving exocentric cameras are needed to enable fairer ego-exo comparisons and drive improved methods.

5. Conclusion

In this work, we answered the question of how well existing, monocular dynamic 3D Gaussian Splatting models perform in egocentric settings. To this end, we compared the performance of existing models on paired ego-exo views from the EgoExo4D dataset. We found that models tend to achieve better reconstruction quality of scenes captured from the exocentric perspective. Additionally, our results suggest that this difference, measured in masked peak signal-to-noise ratio (PSNR), comes from the reconstruction of static parts of the scene, as the dynamic regions tend to be reconstructed with similar PSNR quality between ego and exo views. Overall, our results show the need for models specialized in egocentric reconstruction, as current models struggle with the challenges posed by egocentric video.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision*, pages 720–736, 2018. 1, 2, 5

[3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 1, 2, 5, 6

[4] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. *Advances in Neural Information Processing Systems*, 35:13745–13758, 2022. 2

[5] Delft AI Cluster (DAIC). The delft ai cluster (daic), rrid:scr_025091, 2024. 4

[6] Junli Deng and Yihao Luo. Gaussians on their way: Wasserstein-constrained 4d gaussian splatting with state-space modeling. *arXiv preprint arXiv:2412.00333*, 2024. 1, 2

[7] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1, 2

[8] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022. 1, 2, 6, 7, 8

[9] Qiankun Gao, Yanmin Wu, Chengxiang Wen, Jiarui Meng, Luyang Tang, Jie Chen, Ronggang Wang, and Jian Zhang. Relaygs: Reconstructing dynamic scenes with large-scale and complex motions via relay gaussians. *arXiv preprint arXiv:2412.02493*, 2024. 2

[10] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1

[11] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 2, 3, 8

[12] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. Egolifter: Open-world 3d segmentation for egocentric perception. In *Proceedings of the European Conference on Computer Vision*, pages 382–400. Springer, 2024. 1, 2

[13] Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, et al. Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22072–22086, 2024. 3, 8

[14] Yiming Huang, Beilei Cui, Long Bai, Ziqi Guo, Mengya Xu, Mobarakol Islam, and Hongliang Ren. Endo-4dgs: Endoscopic monocular scene reconstruction with 4d gaussian splatting. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 197–207. Springer, 2024. 2, 5

[15] Iason Karakostas, Aikaterini Valakou, Despoina Gavgiotaki, Zinovia Stefanidi, Ioannis Pastaltzidis, Grigorios Tsipouridis, Nikolaos Kilis, Konstantinos C. Apostolakis, Stavroula Ntoa, Nikolaos Dimitriou, George Margetis, and Dimitrios Tzovaras. A real-time wearable ar system for egocentric vision on the edge. *Virtual Reality*, 28(1):44, 2024. 1, 2

[16] Iman Abaspur Kazerouni, Luke Fitzgerald, Gerard Dooly, and Daniel Toal. A survey of state-of-the-art on visual slam. *Expert Systems with Applications*, 205:117734, 2022. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 1, 2, 3

[18] Daekyum Kim, Brian Byunghyun Kang, Kyu Bum Kim, Hyungmin Choi, Jeesoo Ha, Kyu-Jin Cho, and Sungho Jo. Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view. *Science Robotics*, 4(26): eaav2949, 2019. 1, 2

[19] Mangyu Kong, Jaewon Lee, Seongwon Lee, and Euntai Kim. Dgs-slam: Gaussian splatting slam in dynamic environment. *arXiv preprint arXiv:2411.10722*, 2024. 3

[20] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10138–10148, 2021. 3, 8

[21] Fernando De la Torre Frade, Jessica K. Hodgins, Adam W. Bargteil, Xavier Martin Artal, Justin C. Macey, Alexandre Collado I Castells, and Josep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. Technical Report CMU-RI-TR-08-22, Carnegie Mellon University, Pittsburgh, PA, 2008. 3, 8

[22] Yiqing Liang, Mikhail Okunev, Mikaela Angelina Uy, Runfeng Li, Leonidas Guibas, James Tompkin, and Adam W Harley. Monocular dynamic gaussian splatting is fast and brittle but smooth motion helps. *arXiv preprint arXiv:2412.04457*, 2024. 1, 2, 3, 4, 6, 7, 8

[23] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2642–2652. IEEE, 2025. 1, 2, 5, 8

[24] Zimu Liao, Siyan Chen, Rong Fu, Yi Wang, Zhongling Su, Hao Luo, Li Ma, Linning Xu, Bo Dai, Hengjie Li, et al. Fisheye-gs: Lightweight and extensible gaussian splatting module for fisheye cameras. *arXiv preprint arXiv:2409.04751*, 2024. 3

[25] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024. 2

[26] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 20981–20990, New Orleans, LA, USA, 2022. IEEE. 1, 5, 6

[27] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *Proceedings of the European Conference on Computer Vision*, pages 349–366. Springer, 2024. 1

[28] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 2

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[30] Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26866–26875, 2025. 2

[31] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 1, 2, 4, 8

[32] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2, 4, 8

[33] Razvan-George Pasca, Alexey Gavryushin, Muhammad Hamza, Yen-Ling Kuo, Kaichun Mo, Luc Van Gool, Otmar Hilliges, and Xi Wang. Summarize the past to predict the future: Natural language descriptions of context boost multimodal object interaction anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18286–18296, 2024. 1, 2

[34] Suvam Patra, Kartikeya Gupta, Faran Ahmad, Chetan Arora, and Subhashis Banerjee. Ego-slam: A robust monocular slam for egocentric videos. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 31–40, 2019. 1, 2, 3

[35] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1, 2, 4, 8

[36] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos

Niebles. Home action genome: Cooperative compositional action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11184–11193, 2021. 3, 8

[37] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 5

[38] Andre Rochow, Max Schwarz, Michael Weinmann, and Sven Behnke. Fadiv-syn: Fast depth-independent view synthesis using soft masks and implicit blending. *arXiv preprint arXiv:2106.13139*, 2021. 3

[39] Luca Rossetto, Werner Bailer, Duc-Tien Dang-Nguyen, Graham Healy, Björn THór Jónsson, Onanong Kongmeesub, Hoang-Bao Le, Stevan Rudinac, Klaus Schöffmann, Florian Spiess, et al. The castle 2024 dataset: Advancing the art of multimodal understanding. *arXiv preprint arXiv:2503.17116*, 2025. 3, 8

[40] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, page 4104–4113, Las Vegas, NV, USA, 2016. IEEE. 3

[41] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 3, 8

[42] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018. 3, 8

[43] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019. 1

[44] Pengzhan Sun, Junbin Xiao, Tze Ho Elden Tse, Yicong Li, Arjun Akula, and Angela Yao. Visual intention grounding for egocentric assistants. *arXiv preprint arXiv:2504.13621*, 2025. 1

[45] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Laina, Diane Larlus, Dima Damen, and Andrea Vedaldi. Epic fields: Marrying 3d geometry and video understanding. *Advances in Neural Information Processing Systems*, 36:26485–26500, 2023. 1, 2, 6

[46] Leif Van Holland, Patrick Stotko, Stefan Krumpen, Reinhard Klein, and Michael Weinmann. Efficient 3d reconstruction, streaming and visualization of static and dynamic scene parts for multi-client live-telepresence in large-scale environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4258–4272, 2023. 3

[47] Rui Wang, Quentin Lohmeyer, Mirko Meboldt, and Siyu Tang. Degauss: Dynamic-static decomposition with gaussian splatting for distractor-free 3d reconstruction. *arXiv preprint arXiv:2503.13176*, 2025. 2

[48] Xizi Wang. Tutorial 3: Undistort frames and overlay annotations, 2024. 3

[49] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 1, 2, 3

[50] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 1, 2, 8

[51] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 1, 2, 3

[52] Zeyu Yang, Zijie Pan, Xiatian Zhu, Li Zhang, Yu-Gang Jiang, and Philip HS Torr. 4d gaussian splatting: Modeling dynamic scenes with native 4d primitives. *arXiv preprint arXiv:2412.20720*, 2024. 1, 2, 3

[53] Jiangwei Yu, Xiang Li, Xinran Zhao, Hongming Zhang, and Yu-Xiong Wang. Video state-changing object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20439–20448, 2023. 2

[54] Daiwei Zhang, Gengyan Li, Jiajie Li, Mickaël Bressieux, Otmar Hilliges, Marc Pollefeys, Luc Van Gool, and Xi Wang. Egogaussian: Dynamic scene understanding from egocentric video with 3d gaussian splatting. *arXiv preprint arXiv:2406.19811*, 2024. 1, 2, 3, 4, 5, 6, 8

[55] Jingzhe Zhang, Lishuo Zhuang, Yang Wang, Yameng Zhou, Yan Meng, and Gang Hua. An egocentric vision based assistive co-robot. In *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, page 1–7, Seattle, WA, 2013. IEEE. 1, 2

[56] Lingzhi Zhang, Shenghao Zhou, Simon Stent, and Jianbo Shi. Fine-grained egocentric hand-object segmentation: Dataset, model, and applications. In *Proceedings of the European Conference on Computer Vision*, pages 127–145. Springer, 2022. 1

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6, 7

[58] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. *arXiv preprint arXiv:2504.03886*, 2025. 3

[59] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10324–10335, 2024. 3

Bringing a Personal Point of View: Evaluating Dynamic 3D Gaussian Splatting for Egocentric Scene Reconstruction

Supplementary Material

A. Hyperparameter search space

The search spaces of hyperparameters for Deformable-3DGS, RTGS, and 4DGS are shown in Tables 4, 5, and 6 respectively. For 4DGS and RTGS these search spaces were obtained by combining all existing configurations. For Deformable-3DGS, we search over the shape of the deformation field as well as the total number of iterations.

Parameter name	Values
iterations	$1.5 \cdot 10^4$, $2.5 \cdot 10^4$, $4.0 \cdot 10^4$
deform_depth	6, 8, 10
deform_width	128, 256, 512

Table 4. The hyperparameter search space for Deformable-3DGS.

Parameter name	Values
env_map_res	0, 500
env_optimize_until	$1.0 \cdot 10^9$, $5.0 \cdot 10^3$
iterations	$2.0 \cdot 10^4$, $3.0 \cdot 10^4$
position_lr_max_steps	$1.5 \cdot 10^4$, $3.0 \cdot 10^4$
densification_interval	200, 100
densify_until_iter	$1.0 \cdot 10^4$, $1.5 \cdot 10^4$

Table 5. The hyperparameter search space for RTGS.

B. Per-scene results

The results in the main paper represent the metrics averaged over scenes and over 3 runs. Since the scenes can vary greatly in difficulty, outliers may skew the metrics. To this end, in this section, we evaluate the models per-scene. For each model and metric, we count in how many runs the exo performance was better than ego and we report the ratio of this number to the total number of runs. The total number of runs is 24 for random scenes and 6 for EgoGaussian scenes.

Full results. The results without the static-dynamic separation are presented in Table 7. As we can see for the random scenes, all ratios are above 0.5, indicating that most scenes were reconstructed better from the exo view, which reinforces previous findings. In the EgoGaussian scenes, mSSIM and mLPIPS are always better on the exo views while mPSNR is better in half the runs. This corresponds to the previous results, where we observed the mPSNR to be closer between ego and exo on the EgoGaussian scenes.

Dynamic and static results. The comparison on the dynamic masks is shown in Table 8. As we can see, the mPSNR ratio has visibly dropped to either below 0.5 or close to 0.5, which is in line with the previous findings, where the mean mPSNR was either higher in the ego view or similar. In the static reconstruction from Table 9 we again observe that the static mPSNR is higher in more scenes on the exo view than the ego. Therefore, these results also suggest that the mPSNR-performance between ego and exo views stems primarily from the reconstruction of static objects. Since for mSSIM and mLPIPS the exo reconstruction is also easier on the dynamic masks, no such conclusion can be made for these metrics.

C. Dataset

All 10 scenes included in our dataset have been shown in Figure 5.

D. Extra results for camera motion

D.1. Camera velocity

Results for angular velocity. Let $\bar{\omega}_t$ be the angular velocity of the camera, normalized identically to $\bar{v}_t$. Figure 8 then shows the relationship between $\bar{\omega}_t$ and the reconstruction quality measured in terms of mLPIPS. As we can observe, as angular camera motion increases, reconstruction worsens. This is in line with the results for linear velocity.

Results for other metrics. Figure 6 presents the results for linear and angular velocities when measuring mPSNR and mSSIM. As we can observe, both metrics tend to decrease. This again shows worsening reconstruction quality and hence reinforces previous results.

Significance test results. Table 10 presents the Pearson and Spearman coefficient results for linear velocity on all 3 metrics. As we can observe, all coefficients coincide with a negative correlation between camera velocity and reconstruction quality. Additionally, all p-values are far below 0.05, indicating statistical significance of results. Similar results can be seen for angular velocity in Table 11.

D.2. Camera baseline

Results for angular baseline. Similarly to the linear baseline from Section 3.4.2, we can define the angular baseline to be the highest angular difference between any two camera poses. Figure 9 presents the camera angular baseline plotted against mLPIPS. As we can observe, an increase in camera baseline correlates with worse reconstruction quality, similarly to the linear baseline.

Parameter name	**Values**
`grid_dimensions`	2
`input_coordinate_dim`	4
`output_coordinate_dim`	16, 32
`resolution[-1]`	250, 150, 100, 80, 75, 50, 25
`multires`	[1, 2, 4], [1, 2]
`defor_depth`	1, 0
`net_width`	128, 64
`plane_tv_weight`	$2.0 \cdot 10^{-4}$, $1.0 \cdot 10^{-4}$
`time_smoothness_weight`	$1.0 \cdot 10^{-3}$, $1.0 \cdot 10^{-2}$
`l1_time_planes`	$1.0 \cdot 10^{-4}$
`no_do`	True, False
`no_dshs`	True, False
`no_ds`	True, False
`iterations`	$1.4 \cdot 10^{4}$, $1.5 \cdot 10^{4}$, $2.0 \cdot 10^{4}$
`batch_size`	1, 2
`coarse_iterations`	$3.0 \cdot 10^{3}$
`densify_until_iter`	$1.0 \cdot 10^{4}$, $1.5 \cdot 10^{4}$
`opacity_reset_interval`	$3.0 \cdot 10^{3}$, $3.0 \cdot 10^{6}$
`grid_lr_init`	$1.6 \cdot 10^{-3}$
`grid_lr_final`	$1.6 \cdot 10^{-4}$, $1.6 \cdot 10^{-5}$
`opacity_threshold_coarse`	$5.0 \cdot 10^{-3}$
`opacity_threshold_fine_init`	$5.0 \cdot 10^{-3}$
`opacity_threshold_fine_after`	$5.0 \cdot 10^{-3}$
`pruning_interval`	100, $8.0 \cdot 10^{3}$
`deformation_lr_init`	$1.6 \cdot 10^{-4}$
`deformation_lr_final`	$1.6 \cdot 10^{-5}$, $1.6 \cdot 10^{-6}$
`deformation_lr_delay_mult`	$1.0 \cdot 10^{-2}$

Table 6. The hyperparameter search space for 4DGS.

	Random scenes			**EgoGaussian scenes**		
Model	mPSNR	mSSIM	mLPIPS	mPSNR	mSSIM	mLPIPS
Deformable-3DGS	1.00	1.00	1.00	0.50	1.00	1.00
4DGS	0.88	0.88	0.75	0.50	1.00	1.00
RTGS	0.79	0.79	0.83	0.50	1.00	1.00

Table 7. Per-scene results of the baselines on 8 random and 2 EgoGaussian scenes with 3 runs per scene. Each entry represents the ratio of runs where the performance on the exo view was higher than the corresponding ego view to the total number of runs.

Results for other metrics. Figure 7 presents the results for linear and angular baselines when measuring mPSNR and mSSIM. As we can observe, both metrics tend to decrease. This again shows worsening reconstruction quality and hence reinforces previous results.

Significance test results. Table 12 presents the Pearson and Spearman coefficient results for linear baseline on all 3 metrics. As we can observe, all coefficients coincide with a negative correlation between camera baseline and reconstruction quality. Additionally, all p-values are far below 0.05, indicating statistical significance of results. Similar results can be seen for angular baseline in Table 13.

	Random scenes			EgoGaussian scenes		
Model	mPSNR	mSSIM	mLPIPS	mPSNR	mSSIM	mLPIPS
Deformable-3DGS	0.33	0.75	0.96	0.50	0.50	0.50
4DGS	0.54	0.96	1.00	0.50	0.50	0.00
RTGS	0.58	0.88	1.00	0.50	0.50	0.33

Table 8. Per-scene results of the baselines on 8 random and 2 EgoGaussian scenes with 3 runs per scene. Dynamic mask considered only. Each entry represents the ratio of runs where the performance on the exo view was higher than the corresponding ego view to the total number of runs.

	Random scenes			EgoGaussian scenes		
Model	mPSNR	mSSIM	mLPIPS	mPSNR	mSSIM	mLPIPS
Deformable-3DGS	1.00	1.00	1.00	1.00	1.00	1.00
4DGS	0.88	0.88	0.75	1.00	1.00	1.00
RTGS	0.79	0.79	0.79	0.50	1.00	1.00

Table 9. Per-scene results of the baselines on 8 random and 2 EgoGaussian scenes with 3 runs per scene. Static mask considered only. Each entry represents the ratio of runs where the performance on the exo view was higher than the corresponding ego view to the total number of runs.

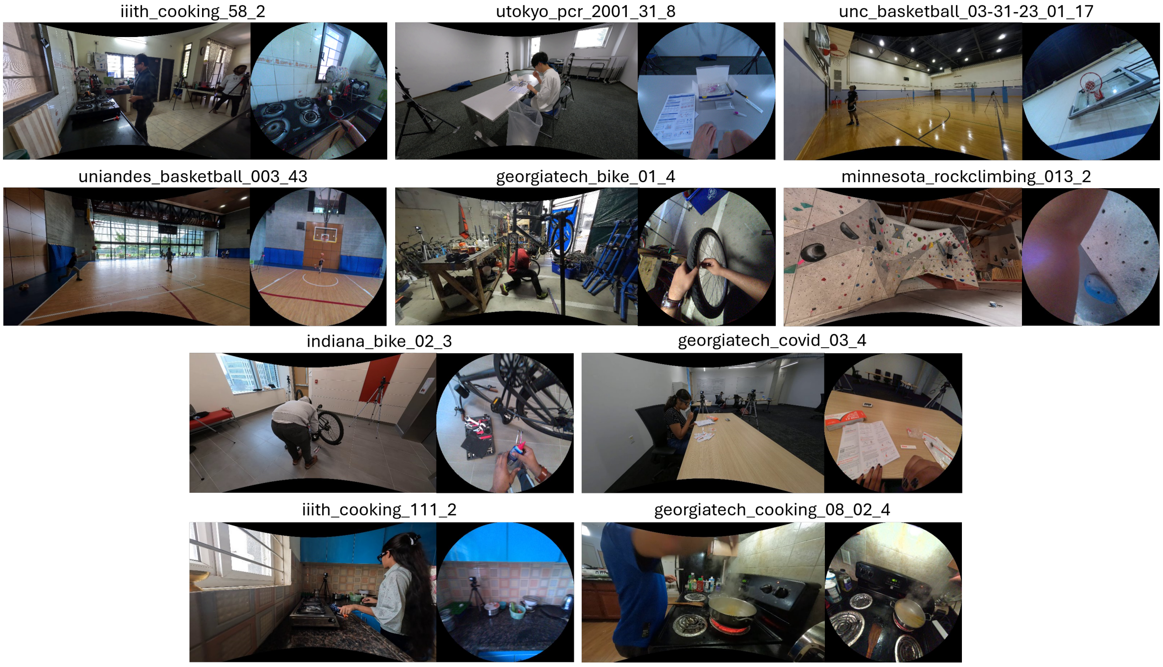


Figure 5. First frame of each scene from the exo (left, first exo camera) and ego (right) views. First 8 scenes are random; last 2 (bottom) are selected EgoGaussian-style scenes. Frames shown after undistortion. As we can see, the scenes are varied. Best viewed zoomed in.

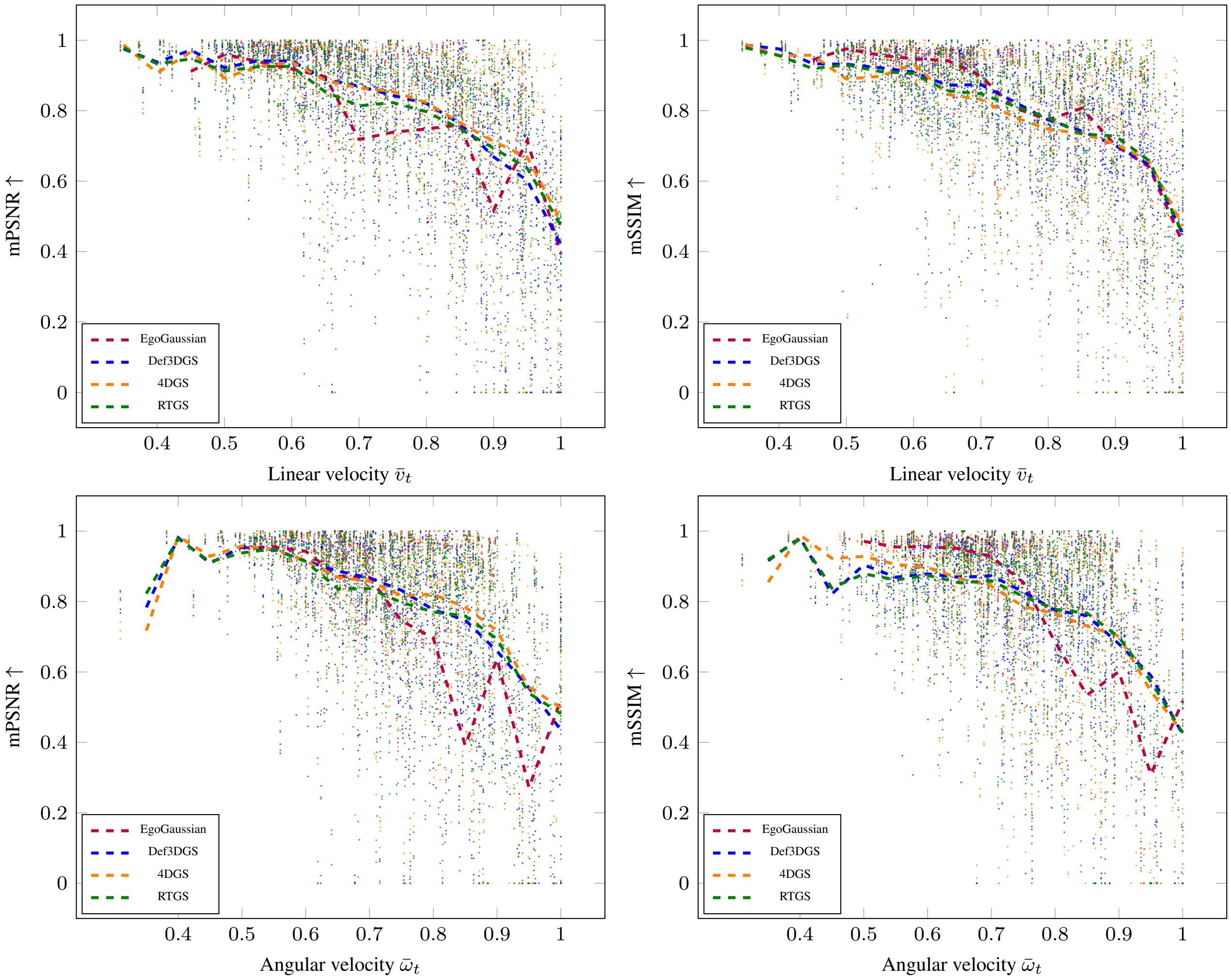


Figure 6. Camera linear and angular velocity results for mPSNR and mSSIM. As we can see, increasing either velocity correlates with a decrease in metrics, indicating worse reconstruction performance.

		mPSNR ↑		mSSIM ↑		mLPIPS ↓	
Model	**Coefficient**	Value	p-value	Value	p-value	Value	p-value
EgoGaussian	Pearson	-0.57	$2.0 \cdot 10^{-31}$	-0.60	$9.8 \cdot 10^{-36}$	0.59	$1.0 \cdot 10^{-33}$
	Spearman	-0.59	$5.3 \cdot 10^{-34}$	-0.60	$1.9 \cdot 10^{-35}$	0.55	$5.4 \cdot 10^{-30}$
Def3DGS	Pearson	-0.60	$3.4 \cdot 10^{-211}$	-0.57	$4.4 \cdot 10^{-184}$	0.48	$5.5 \cdot 10^{-126}$
	Spearman	-0.66	$3.2 \cdot 10^{-264}$	-0.64	$2.5 \cdot 10^{-249}$	0.48	$6.1 \cdot 10^{-125}$
4DGS	Pearson	-0.50	$2.6 \cdot 10^{-139}$	-0.47	$4.8 \cdot 10^{-118}$	0.52	$4.7 \cdot 10^{-151}$
	Spearman	-0.53	$3.9 \cdot 10^{-158}$	-0.51	$5.7 \cdot 10^{-144}$	0.53	$4.2 \cdot 10^{-158}$
RTGS	Pearson	-0.49	$1.1 \cdot 10^{-132}$	-0.52	$1.5 \cdot 10^{-152}$	0.51	$7.9 \cdot 10^{-141}$
	Spearman	-0.57	$6.6 \cdot 10^{-183}$	-0.51	$2.0 \cdot 10^{-144}$	0.51	$7.1 \cdot 10^{-145}$

Table 10. Significance test results for the linear velocity experiments. The p-value approximately indicates the probability that the correlation coefficient is 0. As we can observe, as linear velocity increases, the reconstruction quality decreases with p-values of far below 0.05.

		mPSNR ↑		**mSSIM ↑**		**mLPIPS ↓**	
Model	**Coefficient**	Value	p-value	Value	p-value	Value	p-value
EgoGaussian	Pearson	-0.64	$1.8 \cdot 10^{-42}$	-0.67	$1.6 \cdot 10^{-46}$	0.66	$3.8 \cdot 10^{-46}$
	Spearman	-0.64	$4.1 \cdot 10^{-41}$	-0.69	$1.2 \cdot 10^{-50}$	0.58	$2.2 \cdot 10^{-33}$
Def3DGS	Pearson	-0.52	$6.8 \cdot 10^{-147}$	-0.47	$3.4 \cdot 10^{-121}$	0.46	$3.6 \cdot 10^{-112}$
	Spearman	-0.53	$8.5 \cdot 10^{-156}$	-0.47	$1.5 \cdot 10^{-119}$	0.43	$1.0 \cdot 10^{-98}$
4DGS	Pearson	-0.44	$2.0 \cdot 10^{-100}$	-0.44	$4.1 \cdot 10^{-104}$	0.49	$1.0 \cdot 10^{-132}$
	Spearman	-0.43	$3.9 \cdot 10^{-98}$	-0.43	$9.6 \cdot 10^{-99}$	0.47	$8.8 \cdot 10^{-117}$
RTGS	Pearson	-0.43	$1.6 \cdot 10^{-99}$	-0.44	$4.1 \cdot 10^{-103}$	0.51	$1.2 \cdot 10^{-143}$
	Spearman	-0.48	$5.0 \cdot 10^{-126}$	-0.39	$3.2 \cdot 10^{-79}$	0.49	$2.4 \cdot 10^{-128}$

Table 11. Significance test results for the angular velocity experiments. The p-value approximately indicates the probability that the correlation coefficient is 0. As we can see, as angular velocity increases, the reconstruction quality decreases with low p-values.

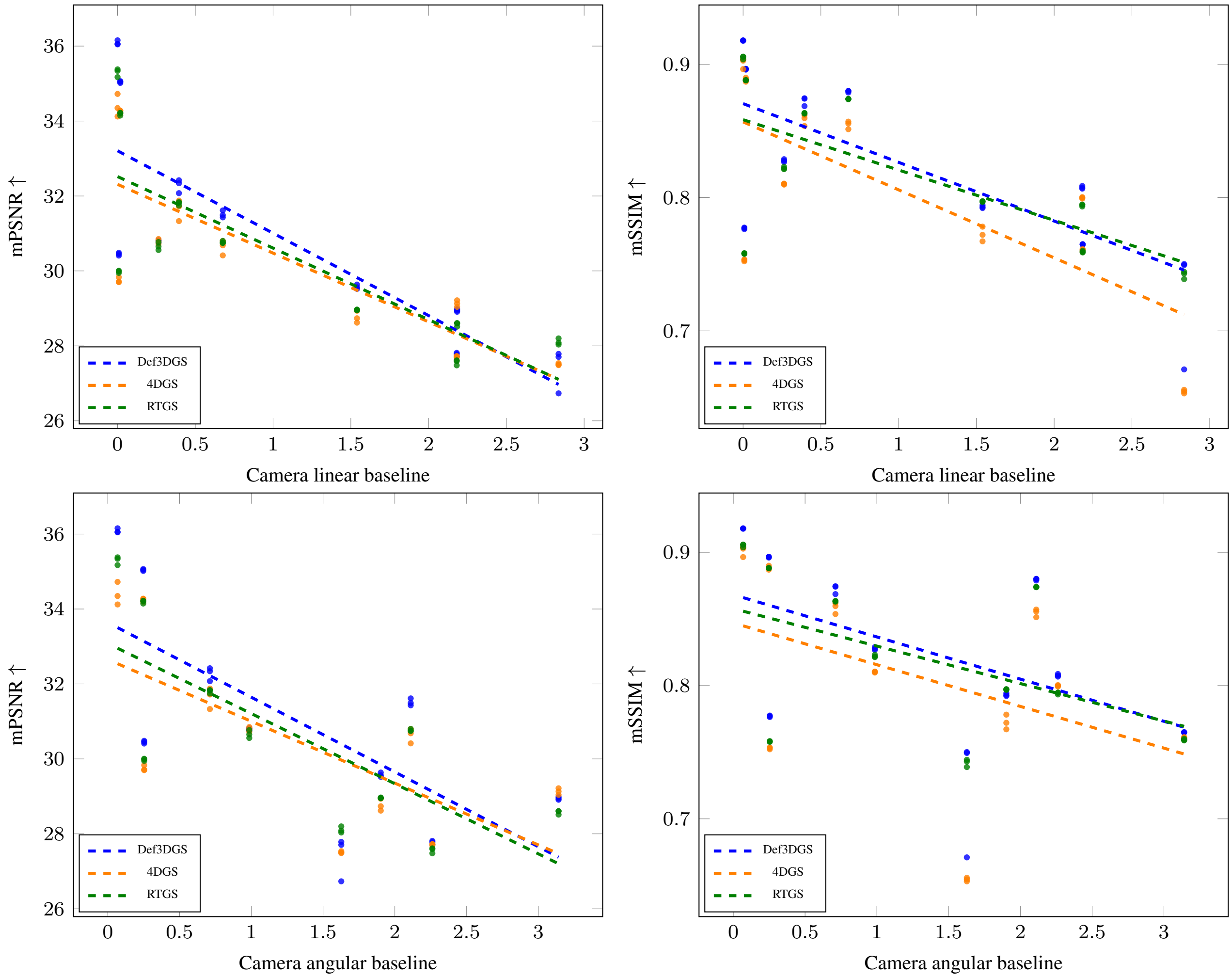


Figure 7. Camera linear and angular baseline results for mPSNR and mSSIM. As we can see, increasing any baseline is correlated with a decrease in metrics, indicating worse reconstruction performance.

Model	**Coefficient**	**mPSNR** ↑		**mSSIM** ↑		**mLPIPS** ↓	
		Value	p-value	Value	p-value	Value	p-value
Def3DGS	Pearson	-0.82	$2.3 \cdot 10^{-8}$	-0.73	$4.5 \cdot 10^{-6}$	0.74	$3.6 \cdot 10^{-6}$
	Spearman	-0.82	$2.7 \cdot 10^{-8}$	-0.68	$3.3 \cdot 10^{-5}$	0.63	$1.7 \cdot 10^{-4}$
4DGS	Pearson	-0.81	$4.5 \cdot 10^{-8}$	-0.74	$3.6 \cdot 10^{-6}$	0.79	$1.7 \cdot 10^{-7}$
	Spearman	-0.82	$3.9 \cdot 10^{-8}$	-0.61	$3.8 \cdot 10^{-4}$	0.75	$2.1 \cdot 10^{-6}$
RTGS	Pearson	-0.80	$1.0 \cdot 10^{-7}$	-0.69	$2.7 \cdot 10^{-5}$	0.67	$4.5 \cdot 10^{-5}$
	Spearman	-0.81	$6.8 \cdot 10^{-8}$	-0.61	$3.5 \cdot 10^{-4}$	0.61	$3.9 \cdot 10^{-4}$

Table 12. Significance test results for the linear baseline experiments. The p-value approximately indicates the probability that the correlation coefficient is 0. As we can observe, as linear baseline increases, the reconstruction quality decreases with p-values of far below 0.05.

Model	**Coefficient**	**mPSNR** ↑		**mSSIM** ↑		**mLPIPS** ↓	
		Value	p-value	Value	p-value	Value	p-value
Def3DGS	Pearson	-0.72	$8.4 \cdot 10^{-6}$	-0.50	$4.5 \cdot 10^{-3}$	0.44	$1.6 \cdot 10^{-2}$
	Spearman	-0.73	$5.6 \cdot 10^{-6}$	-0.54	$2.2 \cdot 10^{-3}$	0.47	$9.0 \cdot 10^{-3}$
4DGS	Pearson	-0.70	$1.5 \cdot 10^{-5}$	-0.43	$1.7 \cdot 10^{-2}$	0.40	$2.8 \cdot 10^{-2}$
	Spearman	-0.71	$1.1 \cdot 10^{-5}$	-0.49	$6.0 \cdot 10^{-3}$	0.59	$5.3 \cdot 10^{-4}$
RTGS	Pearson	-0.75	$1.5 \cdot 10^{-6}$	-0.49	$5.8 \cdot 10^{-3}$	0.44	$1.4 \cdot 10^{-2}$
	Spearman	-0.76	$9.8 \cdot 10^{-7}$	-0.48	$7.7 \cdot 10^{-3}$	0.45	$1.2 \cdot 10^{-2}$

Table 13. Significance test results for the angular baseline experiments. The p-value approximately indicates the probability that the correlation coefficient is 0. As we can see, as angular baseline increases, the reconstruction quality decreases with low p-values.

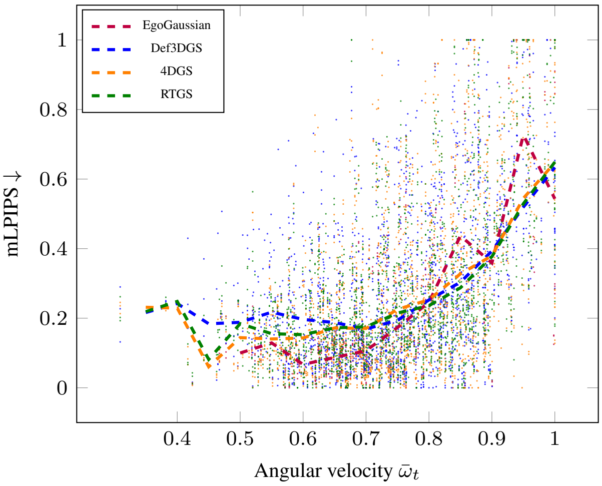


Figure 8. Camera angular velocity $\bar{\omega}_t$ plotted against mLPIPS. Additional trend lines are plotted. As we can observe, as angular velocity increases, mLPIPS increases, which corresponds to worse reconstruction quality.

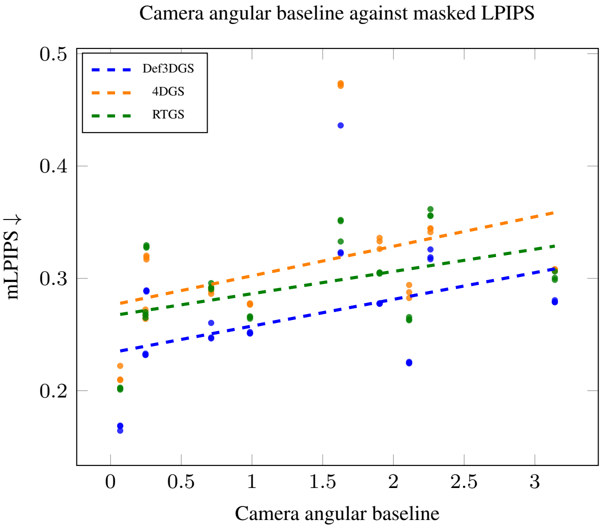


Figure 9. Camera angular baseline plotted against mLPIPS. Additional linear regression models fitted are shown. As we can observe, as camera baseline increases, mLPIPS increases, which corresponds to worse reconstruction quality.