# Report of GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians

Victor Ferrari
UFF
victorferrari@id.uff.br

Leonardo Nanci
UFF
leonardonanci@id.uff.br

Hacker Horácio Macêdo
UFF
horaciomacedo@id.uff.br

PhD Student Alberto Arkader Kopiler
IMPA
alberto.kopiler@impa.br

## 1. Revision

### 1.1. Summary

The GaussianAvatar paper introduces a novel approach to 3D avatar modeling and animation using animated 3D Gaussians. The system explicitly represents human surfaces and enables efficient fusion of 3D appearances from 2D observations. The implementation follows a two-stage process, beginning with a framework optimization stage focusing on basic framework optimization without pose-dependent information.

The process begins with a dynamic integration stage that performs 2D-to-3D mapping and incorporates dynamic properties to achieve realistic movement. A significant innovation in this approach is the joint optimization of motion and appearance during the modeling phase, which is facilitated by a dynamic appearance network and an optimizable feature tensor designed for motion-to-appearance mapping. This method not only corrects any initial misalignments in motion but also enhances the overall quality of the appearance.

### 1.2. Related Work

A survey of existing research reveals two primary approaches in the field. The first is Neural Rendering for Human Reconstruction, which is NeRF-based. While these methods can give good results, they often do not meet expectations when it comes to accurately representing surfaces and capturing intricate geometrical details. Implicit 3D volume approaches struggle to manage the complex shapes of human forms, which restricts their practical usability.

The second approach focuses on Avatar Modeling from Monocular Videos. Despite performing visually effective outcomes, regression-based methods cannot depict dynamic appearance changes for realistic animations. Pre-scanned templates are unsuitable for real-time applications. Implicit NeRF models used for pose estimation have demonstrated bad trustworthiness and accuracy. These limitations underscore the need for better methods to address human avatars' geometric intricacies and dynamic nature.

### 1.3. Technical Achievements

The GaussianAvatar system is an improvement over existing methods. It's rendering speed of 35 FPS, surpass systems like HumanNeRF (0.22 FPS) and InstantAvatar (3.87 FPS). And the training efficiency requires just 30 minutes compared to HumanNeRF's 13 hours. Performance metrics, such as PSNR and SSIM, show gains across datasets, with PSNR improving by 2–8 points and SSIM increasing by 0.01–0.04 over baseline models. The system is compatible with SMPL and SMPL-X models, enabling capabilities like hand animation. It maintains consistent 3D appearance quality even when animated with out-of-distribution motions, a necessary feature for real-world use.

### 1.4. Limitations

The GaussianAvatar project also has some limitations. Handling loose clothing, such as dresses, remains challenging due to constraints in the skinning weights derived from the SMPL model. This can lead to blurry clothing visuals and incomplete point clouds for deformations. Also, the system relies heavily on precise foreground segmentation. Segmentation errors can show artifacts, like black lines on the

surface, which affect the output. The technical implementation also needs a careful setup, usually because of the need for manual adjustments when automatic segmentation falls short, making the process time-consuming.

### 1.5. Assesment

Rating: 4 out of 5 (Possibly Accept)

The GaussianAavatar system is a step forward in human avatar modeling and animation. It addresses issues like surface detail representation and dynamic appearance modeling while improving rendering speed and training efficiency. It is very good at handling complex geometries and maintaining appearance consistency across different poses and viewpoints. However, areas such as managing loose clothing and dependency on accurate segmentation need improvement. Despite these limitations, the system's advancements lay a basis for further development.

## 2. Archeologist

Several approaches explore the reconstruction of humans from a set of images or a video. In this section, we highlight techniques that enable the creation of GaussianAvatar. In addition, we present contemporary works that compete to be the state-of-the-art in creating human avatar via 3D Gaussian Splatting.

### 2.1. Previous work

The input to the pipeline proposed in GaussianAvatar includes a human mesh fitted to the current frame. Such representation is acquired using a *Skinned Multi-Personal Model* (SMPL) [15]. Given a template mesh, a set of corresponding joints and their blending weights, this technique learns to distort the vertices to fit different human body shapes. Then, when animating the model, the method learns to perform fine mesh deformations that capture soft-tissue behavior, such as muscle tension and relaxation. Extending the work of Loper et al. [15], the article *Expressive Body Capture: 3D Hands, Face and Body from a Single Image* (SMPL-X) [16] proposes to additionally capture facial expressions and hand poses in one single pipeline. The authors argue that unifying such pose features leads to better results, due to the context provided by body poses. SMPL-X is another input option for the GaussianAvatar pipeline.

Improving previous techniques for rendering human avatars, in 2018 Alldieck et al. proposed a novel pipeline in the paper *Video Based Reconstruction of 3D People Models* [7]. The first step consists in fitting SMPL meshes to the input images, then unpose them. Afterwards, a consesnsus mesh is determined to better represent all meshes previously computed. Finally, a texture map is extracted, the canonical pose is animated and the final image is rendered. GaussianAvatar heavily builds on top of this pipeline, modifying the texture extraction and rendering steps to use volumetric 3D Gaussians.

### 2.2. Contemporary work

*Animatable Gaussians: Learning Pose-Depandent Gaussian Maps for High Fidelity Human Avatar Modelling* [13] takes a different approach from GaussianAvatar. Instead of creating a SMPL model, the authors propose to capture shape using a signed distance function, then map animations via two 2D position maps (front and back canonical pose views). Subsequently, a neural network pass defines the parameters of the primitives in two gaussian maps; and the gaussians are positioned in canonical pose space. In the end, the model is posed and rendered. Due to the implicit initial representation, this method seems to better capture loose clothing, such as dresses and skirts.

*MIGS: Multi-Identity Gaussian Splatting via Tensor Decomposition* [8] proposes a method to represent multiple identities (textures) storing a reduced number of parameters. Arguing that part of the information is redundant across identities, the authors apply CANDECOMP/PARAFAC tensor decomposition, resulting in 3 low-rank tensors that reflect the main components of the original tensor. Using a set of neural networks, the gaussians are positioned in 3D space and then rendered.

*Generalizable Human Gaussians for Sparse View Synthesis* [12] calculate multiple dilations of the human shape (called scaffolds). The features of each scaffold are stored in feature maps that are input to an encoder-decoder neural network. The output is a set of gaussian maps, one for each scaffold, that are used to position and render the 3D gaussians.

*GPS-Gaussian: Generalizable Pixel-wise 3D Gaussian Splatting for Real-Time Human Novel View Synthesis* [18] is heavily inspired in MVSplat [9]. The authors use a cost volume as input to a depth estimator. An encoder-decoder network takes the depth and outputs gaussian parameters to be used in rendering. Gaussian position and color are extracted directly from the estimated depth and input images.

## 3. Code and Experiments

GaussianAvatar provides, alongside the paper, a git repository providing code for running the technique from preprocessing to visualization in Python scripts and links to other repositories. It also provides a simple set of instructions to run the entire process, from top to bottom. Aside from real-time animation avatars from a monoscopic camera, which is a promised feature to be made public in the future, every other feature the paper promises is present in the repository and can be executed in some way, shape or form. Most of the code featured on the repository is adapted from other works: PoP (Power of Points), InstantAvatar, HumanNeRF and 3D Gaussian Splatting.

Setting up the environment to work as instructed is troublesome since the work does not provide a complete list of dependencies. Installation is marginally easier on Ubuntu than on Windows since dependencies are easier to install using a package manager other than finding the correct binaries for working with older versions of PyOpenGL. It is uncertain if the newer PyOpenGL versions are broken or incompatible with the project. The setup also requires a working compilation of 3D Gaussian Splatting, but that is well documented enough, and InstantAvatar's dependencies in order to fully pre-process new datasets.

For this experiment, we evaluated the GaussianAvatar's reproducibility using a total of two datasets. The first dataset was the one pointed out as an example by the authors since it required no previous setup in order to work. The second one, also provided by the original authors, required a certain amount of pre-processing before training. Since it did not require the entire pre-processing routine and the instructions were minimal, the hacker struggled to understand which steps should have been taken to train using that dataset properly. The data used to feed the training process

The pre-processing step is complicated and wasn't fully dominated when writing this report. The instructions on navigating the process and the desired state of the dataset to conduct the subsequent training are unclear, and many states require different pre-processing solutions. A great part of the pre-processing step involves accessing InstantAvatar's repository to retrieve the necessary scripts and learn how to use them. The project provides scripts to adapt InstantAvatar's dataset, which requires previous knowledge in how the InstantAvatar preprocesses its dataset to understand how and when they are used.

The entire training routine, composed of two stages, is mainly handled by one single script filled with errors. The first stage has many syntax errors that demand manual correction but works mostly fine, given the appropriate batch size for the amount of available VRAM. The second stage requires an intermediate processing step (that is sufficiently documented) between both training stages and does not work right out of the gate. The first stage has the Loss function as described in the paper, while the second stage's Loss function is missing one of the components it should have according to the paper. For training purposes, this report's author copied the part missing from the first stage into the second stage, since the paper points to both functions having the same component.

Evaluating results is possible, although troublesome. The evaluation script is highly dependent on the PyOpenGL and only ran on the output of the first stage. The first stage was well evaluated in both cases, showing extremely good PSNR, SSIM and LPIPS measurements in both datasets and showing a subtle decrease on the dataset with looser clothing and floating hair. No second-stage evaluation was conducted since the output of the second stage is seemingly incompatible with the evaluation script. No complete evaluation from both stages was possible during this investigation.

The same problem was observed while trying to render novel poses using the provided script for this task; we could render those new poses after the first stage of training but not after the second one. Evaluating and rendering novel poses after the complete cycle of two-stage training leads us to having the scripts looking for input files that are not generated in any step of the way.

Ultimately, this work is barely functional in its state, and Github's issues pages are evidence of such. This is a sad state of affairs, since the results that we can see are promising and look good, even when dealing with difficult features. Unfortunately, to get to those results, the user must play the role of a fixer in order to fix the code properly. Updates on the GitHub with fixes and rewriting instructions for greater clarity are crucial for its code to be properly evaluated, although the first results are promising.

Shortly after presenting this work, the repository received an update after a few months. However, the content of this report remains relevant, since none of the problems we talk about in this report seem to be addressed.

## 4. PhD Project

According to the authors [10] there are limitations to the proposed model. Their method may generate artifacts due to inaccurate video foreground segmentation and encounter

challenges in modeling loose outfits such as dresses. So one of the possible improvements of this work would be to invest in better foreground segmentation and texture modeling. These ideas first came to the PhD student's mind to accomplish a rule that determined that the work following this article would only be possible after the publication.

Despite this, the PhD student proposed an idea inspired by the input being a video and the output being an animated avatar. The name of the proposed project is Generative 3D Animated Avatar. The original inputs were 2D monocular video and pose references, which were processed by tasks responsible for creating novel view synthesis and making a 3D Reconstruction based on Gaussian Splatting [11] generating an animated avatar as output. The PhD student proposal is to maintain the monocular 2D video as input or with one shot or a few shots of images of the same recorded scene and replace the pose reference with text instructions. The tasks could be classified as style transfer, pose transfer, and skin transfer. The tasks could be applied alone or all of them. In either case, it would produce an animated avatar (video or gif).

For multimodality, you could also expand the input to music or sound, to get an animated avatar with synchronized sound or music along with its animation. So you are using text, one or more images, video, sound, and music to address the multimodality so that you can try using an MLM (Multimodal Large Language Model) to accomplish the proposed tasks.

To prove the viability of the proposed idea we have done some research and found the following references SAM2 [17], Deepmotion [2, 3], autodesk wonder animation video to 3D scene [1], mediapipe tracking4all [6], justsketch.me [5], sora [14], runwayml gen-3 [4], just to name a few.

## 5. Conclusions

This analysis examines "GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians" from four perspectives: a conference reviewer, archaeologist, hacker, and PhD student. The archaeologist positions GaussianAvatar as an evolution of existing avatar methods, building on the SMPL model pipeline but innovating in texture extraction and rendering using 3D Gaussians. The reviewer recommended "Possibly Accept," praising the interactive rendering speed and improved handling of challenging features like loose clothing and hair, though noting manual segmentation as a limitation. The hacker identified implementation issues, including poorly documented scripts, coding errors, and setup difficulties. Despite these problems, the technique produces impressive results when working properly. The PhD student suggested future research directions, including improving foreground segmentation, enabling avatar-to-avatar data transfer, and developing audio-synchronized animations. While the paper represents meaningful progress in real-time avatar rendering with 3D Gaussians, it could benefit from better loss component analysis and improved implementation quality.

## References


[1] Autodesk wonder-animation-video-to-3d-scene. https://adsknews.autodesk.com/pt-br/news/autodesk-launches-wonder-animation-video-to-3d-scene-technology/, 2024. (accessed Dec. 11, 2024). 4

[2] Deepmotion - saymotion-video-text-to-3d-animation. https://www.deepmotion.com/post/saymotion-v2-3-video-text-to-3d-animation-in-one-platform-animate-3d-integration, 2024. (accessed Dec. 11, 2024). 4

[3] Deepmotion - animate 3d. https://www.deepmotion.com/animate-3d, 2024. (accessed Dec. 11, 2024). 4

[4] Gen3 - runwayml. https://runwayml.com/research/introducing-gen-3-alpha, 2024. (accessed Dec. 11, 2024). 4

[5] Justsketch.me. https://justsketch.me/pt/, 2024. (accessed Dec. 11, 2024). 4

[6] Mediapipe tracking4all. https://www.tracking4all.com/, 2024. (accessed Dec. 11, 2024). 4

[7] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[8] Aggelina Chatziagapi, Grigorios G. Chrysos, and Dimitris Samaras. Migs: Multi-identity gaussian splatting via tensor decomposition. In *Computer Vision – ECCV 2024*, pages 388–408, Cham, 2025. Springer Nature Switzerland. 2

[9] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024. 2

[10] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 634–644, 2024. 3

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 4

[12] Youngjoong Kwon, Baole Fang, Yixing Lu, Haoye Dong, Cheng Zhang, Francisco Vicente Carrasco, Albert Mosella-Montoro, Jianjin Xu, Shingo Takagi, Daeil Kim, Aayush

Prakash, and Fernando De la Torre. Generalizable human gaussians fornbsp;sparse view synthesis. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVIII*, page 451–468, Berlin, Heidelberg, 2024. Springer-Verlag. 2

[13] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[14] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024. 4

[15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), 2015. 2

[16] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[17] 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. 4

[18] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2