UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Self-Supervised 3D Reconstruction from Monocular Videos via Implicit Representation

Author:
Leonardo BOCCHI

Supervisor:
Meysam Madadi
Sergio Escalera

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

January 26, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc Fundamental Principles of Data Science Master's Thesis

Self-Supervised 3D Reconstruction from Monocular Videos via Implicit Representation

by Leonardo BOCCHI

In this thesis, we undertake the problem of 3D human reconstruction from monocular videos. We begin by analyzing the current state of the art regarding this vast field of research and introducing the different variations of this problem. After introducing the necessary notions, we tackle the problem of 3D reconstruction through self-supervised training via implicit neural representation. We adopt a reconstruction module as our baseline and present a number of experiments, with the objective of improving the quality of the reconstruction. Finally, we present the AGen model as a generalizable solution. Leveraging a canonical implicit neural representation, the model is trained in a self-supervised manner, using no additional 3D annotations on the data, in order to be able to perform a faster reconstruction over previously unseen data. The proposed model is intended as a scalable solution that does not rely on the need for precise 3D annotations, providing a fast and refinable 3D modeling tool for several applications.

Acknowledgements

I would like to express my sincere appreciation to Meysam Madadi and Sergio Escalera for their invaluable guidance and support throughout the research and writing process of this master's thesis. Their expertise, constructive feedback, and unwavering commitment significantly contributed to the successful completion of this work. Furthermore, I am grateful for the opportunity to work with the Human Pose Recovery and Behavior Analysis group as it has been of great value and it proved to be a continuous incentive to strive for knowledge, expertise, and research.
Special thanks to my family for their support in my studies, without which this invaluable experience would not have been possible. In particular, for sharing the emotional burden of being away while remaining of persisting encouragement.
I am also deeply grateful for my friends who have been by my side during this one-year-and-a-half-long path, which has been oftentimes demanding and overwhelming. Special thanks to Àlex, David, and Sara for their undoubted support not only in an academic sense but also in numerous other life-related difficult moments.
Thank you all for being an integral part of this academic journey.

Leonardo Bocchi
Universitat de Barcelona
January, 2024

Contents

Abstract **iii**

Acknowledgements **v**

1 Introduction **1**
1.1 3D Rendering and 3D Reconstruction . . . 1
1.2 Raycasting . . . 2
1.3 Neural Implicit Representations . . . 2
1.4 Methods and Work Environment . . . 3

2 Background and SOTA **5**
2.1 Different Shades of the Problem . . . 5
2.2 Experimented Approaches and Techniques . . . 5
2.3 Datasets and Evaluation . . . 6
2.4 Supervised vs Unsupervised . . . 8

3 Video Reconstruction Module Baseline **9**
3.1 Scope of the Project . . . 9
3.2 Operating Principles . . . 9
3.3 Dataset preprocessing . . . 10
3.4 Architecture of the Model . . . 10
3.5 Self-Supervised Optimization . . . 11
3.5.1 Global optimization objectives . . . 11
3.5.2 Scene decomposition objectives . . . 11

4 Video Reconstruction Module Experiments and Evaluation **13**
4.1 Experiments . . . 13
4.2 Evaluation . . . 14
4.2.1 Quantitative evaluation . . . 14
4.2.2 Qualitative evaluation . . . 16

5 AGen model baseline **21**
5.1 Scope of the Project . . . 21
5.2 Architecture of the Model . . . 21
5.3 Training . . . 22
5.4 Leveraging the pretraining of the implicit network . . . 23
5.5 Evaluation . . . 23
5.6 Dataset . . . 24

6 AGen Model Experiments and Evaluation **25**
6.1 Experiments . . . 25
6.2 Evaluation . . . 27
6.2.1 Quantitative evaluation . . . 27
6.2.2 Qualitative evaluation . . . 30

7 Conclusions and Future Directions **33**
7.1 Considerations 33
7.2 Future directions 33
7.3 Conclusions 34

Bibliography **35**

Chapter 1

Introduction

3D modeling has been an area of great interest in Computer Science for years now, resulting in a firm commitment from researchers to further optimize algorithms, in terms of computational complexity as well as in terms of 3D representation capabilities. Recently, 3D modeling has had an additional spike in interest, given the wide range of applications in fast-growing fields such as virtual and augmented reality.
New requisites for the implemented methods have risen, such as higher resolution, faster rendering times, and cheaper setup requirements. Alongside, novel techniques have been developed, including implicit representations and neural deep-learning-based reconstruction techniques, aiming for more accurate results with less prior information.
In this work, we will focus on 3D avatar reconstruction from monocular videos, that is the reconstruction of the 3D model of a clothed subject relying solely on a single video without depth information. This task is by nature a very complex one since the available information for the reconstruction is particularly limited, but it comes with countless applications.

1.1 3D Rendering and 3D Reconstruction

At its core, 3D rendering is the process of converting three-dimensional data into two-dimensional images, simulating the play of light and shadow to create visually realistic or stylized representations of objects and scenes. It comprises of three main elements: modeling and geometry, texturing and materials, lighting and shadows.
3D reconstruction is the inverse process, aiming to recreate the 3D models and geometry from a 2D scene.

1. *Models and Geometry.* 3D models can be represented in various ways. However, regardless of the representation, the final result needs to be made discrete in order to be processed. Hence, each model ultimately consists of vertices, edges, and faces, defining the surface and volume of the object.

2. *Textures and materials.* Textures refer to surface details like skin, clothing patterns, and environmental reflections applied to virtual models. Materials dictate how these surfaces interact with light, crucial for achieving realism.

3. *Lighting and shadows.* Lighting involves simulating the illumination sources, affecting how light falls on and interacts with surfaces. Realistic lighting contributes to accurate color representation and depth perception. Shadows, a natural outcome of lighting, enhance the perception of depth and spatial relationships within the virtual scene.

1.2 Raycasting

Raycasting is a fundamental technique in computer graphics and 3D rendering that involves tracing rays from the eye (or camera) to determine what objects or surfaces they intersect with. It is one of many rendering techniques, each with its own advantages and costs, e.g. rasterization, ray tracing, path tracing.
A ray in the context of raycasting is a straight line defined by an origin point (usually the position of the camera) and a direction vector that specifies the path the ray travels.

$$P(t) = O + tv$$

where $P(t)$ is a point on the ray, O is the origin point, v is the direction vector, and t is a scalar parameter determining a specific point along the ray.

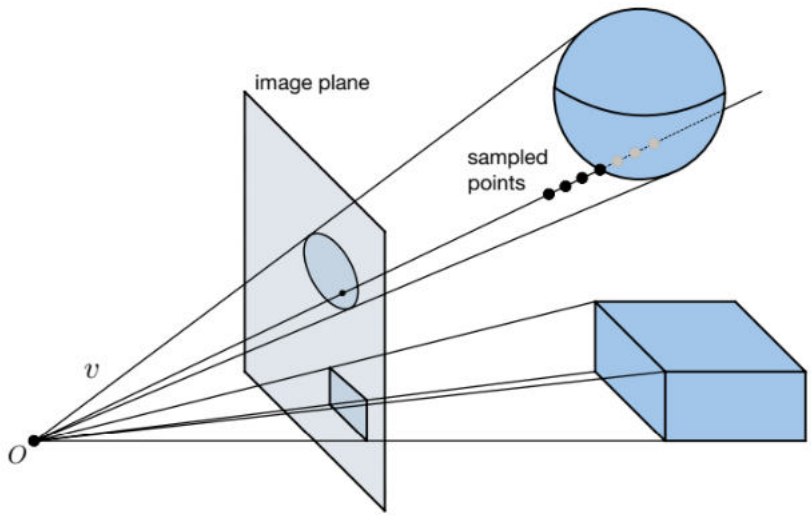

FIGURE 1.1: Ray casting

When a ray is cast into a 3D scene, it may intersect with various objects or surfaces, as depicted by figure 1.1. To find these intersections, algorithms are used to test whether the ray intersects with objects in the scene. For simple geometric shapes, the intersection can be calculated analytically. For more complex shapes, numerical search methods and sampling methods are employed.

1.3 Neural Implicit Representations

Unlike traditional geometric representations, which rely on explicit surfaces defined by vertices and polygons, neural implicit representations leverage deep learning algorithms to define 3D shapes implicitly, offering a more versatile and expressive approach to modeling intricate geometry and capturing detailed structures.
The idea is to use implicit functions to define surfaces, allowing for a smooth representation virtually unlimited in terms of resolution. Furthermore, the implicit function can be obtained through the training of a deep learning model, allowing for the definition of extremely complex functions.

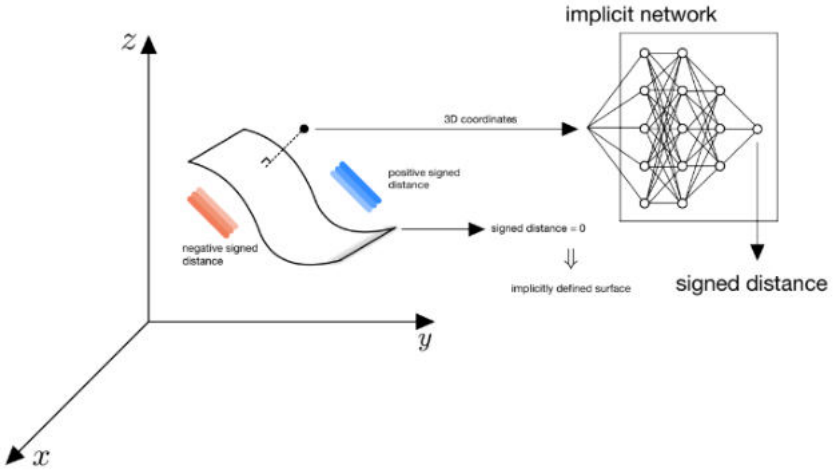

FIGURE 1.2: Neural Implicit Function

However, neural implicit functions also pose some problems and challenges, in particular, they require large amounts of evaluations in order to retrieve the represented surface. For this reason, they also heavily rely on the optimization of sampling techniques.

1.4 Methods and Work Environment

This work has been conducted in collaboration with the Human Pose Recovery and Behavior Analysis group (HuPBA), whose support has been pivotal, both in terms of resources, as well as counseling. The implementation of the study which will be presented in the following chapters was developed on the HuPBA's servers, relying on multiple GPUs that were made available. In particular, one or more *NVIDIA GeForce RTX 3090, 24GB* were employed, and are necessary for the reproduction of the results. The code produced is made available for consultation through a GitHub repository, AGen. Also, in the repository we provide the *Dockerfile* used to create and build the image and container, as it is the procedure adopted to develop the code in the first place. The code is developed using *PyTorch* and additional related libraries. To reproduce the results it is sufficient to run the AGen_test.py script, indicating the correct checkpoint file, and to retrain the model use the AGen_train.py script. All the parameters of the model, dataloader, videos, and configuration options are included in .yaml files in the 'confs' folder, making use of the configuration management library *Hydra*. Furthermore, the scripts make use of a *Weights&Biases* API as a logging system, to record and keep track of the training. Additional indications to reproduce the results are given in the repository's main README page. Finally, for transparency, the work has been conducted using multiple repositories to better divide the work through different stages, in particular IF3D has been used to work on the video reconstruction module experiments, and AGen has been used to work on the AGen model experiments. These are stored on *HuggingFace* and we provide them for completeness, but they have been used simply as storage, they are not meant to be presented to an end user. For consultation, the GitHub repository, AGen, should be considered.

Chapter 2

Background and SOTA

In this section, we provide an objective analysis of the background and state-of-the-art research pertinent to our study. We examine the multifaceted aspects of the problem, survey the methodologies explored in previous works, and discuss the datasets and evaluation techniques commonly employed. This overview serves as a pragmatic foundation, offering a clear context for our research within the existing academic landscape.

2.1 Different Shades of the Problem

The 3D reconstruction of subjects has been tackled from different angles, attempting to compromise between necessary information in the data and the quality of the results. Some have been focusing on the reconstruction from single-view or multi-view images, obtaining reliable results at inference showing good generalization over unseen data [[1], [2], [3]]. Others, to improve the quality of the reconstruction, have aided the training of the models through the use of depth information, relying on depth cameras to create large enough datasets. Finally, some have been tackling the problem with minimal information, considering only in-the-wild monocular videos, and designing models to reconstruct the subject in a specific video as needed [4]. Each one of these approaches has its advantages and presents its own challenges. In particular, supervised trained models often reach good results and generalization, but they require large 3D-labeled multi-view datasets, which are remarkably challenging to produce, given the required high-precision calibration of the measuring equipment. For this reason, the creation of these datasets has recently seen a lot of contributions [[4], [5], [6], [7], [8], [9]]. On the other hand, these models are subject to generalization problems due to the fact that the datasets on which they are trained are not comprehensive of the countless variations in garment types, body shapes, body poses, and background environments. Therefore, models that are further optimizable and tunable at inference to improve the quality of the reconstruction have been attracting more and more attention, pushing the reconstruction quality at the cost of longer inference time. Some of these models adopt the use of physical-based losses, regularization losses, and consistency losses to optimize the results even in the absence of 3D annotations or ground-truth values.

2.2 Experimented Approaches and Techniques

In dealing with the above-mentioned challenges, different solutions have been proposed, ranging from various model architectures to alternative topology representations, from different training techniques to alternative optimization and regularization losses.

Topology representation. To represent the reconstructed surfaces and volumes, the choice falls mostly between meshes, point clouds, and implicit functions. While meshes are broadly used and well-optimized, they lack the representative capabilities for complex topologies. Point clouds, on the other hand, have higher degrees of freedom to represent the challenging geometries of clothing, but they lack implicit relative consistency, which needs to be imposed with other constraints [10]. Finally, implicit functions have the flexibility to represent complex topologies, while maintaining inter-element consistency, and allow for a virtually unlimited scaling in terms of resolutions, at the cost of being computationally more demanding [[10], [11], [12], [13]].

Model architecture. The adopted model architectures heavily depend on the kind of available data and chosen topology representation. To achieve generalization through neural implicit functions leveraging supervised training, a pixel-aligned autoencoding approach is oftentimes favored [[1], [3]]. Also, body pose conditioning has proven to be extremely valuable for the training of implicit neural networks in general, frequently exploiting SMPL pre-processing estimates, both in supervised and unsupervised training settings [4].

Training approaches. The available data annotations and adopted topology representation also constrain and force the adaptation of training methods and optimization functions. When possible, physical losses comparing surface normals, mesh intersection, and vertices alignment have shown good optimization and generalization capabilities [[14], [15]]. In general, regardless of 3D annotations, rendering losses, with their different forms depending on the geometry representation, are frequently adopted, since they leverage the 2D image as ground truth. [[1], [3], [4]] Moreover, another commonly implemented loss function is the Eikonal loss, as it does not rely on additional information in the training data [16]. Finally, various regularization losses, when cautiously selected, have shown considerable contributions to the training of the models [[4], [16]].

2.3 Datasets and Evaluation

Extensive work has been carried out to create exhaustive training datasets that would be able to capture the extreme variability in terms of garments, body shapes, and poses. Furthermore, a lot of effort has been put into producing these datasets with the most amount of 3D information possible, to allow ground truth values and supervised training [[5], [6], [7], [8], [9]]. However, all these solutions need to compromise between being representative of a large variety of cases and the amount of 3D annotations and precision.
Furthermore, evaluation of the results is not a straightforward process, as there is not a systematically applicable data-agnostic metric or evaluation method to consistently compare results. The possible measures of the quality of the reconstruction rely first and foremost on the data's 3D annotations, and there are numerous, yielding oftentimes different and inconsistent results when applied for validation. Some of the most frequently adopted in this area of research are

- *Rendering quality metrics.* Needless of 3D annotations, measure the quality of the rendering image obtained from the 3D reconstruction projection. The two

major ones are **SSIM (Structural Similarity Index)**, and **PSNR (Peak Signal-to-Noise Ratio)**.

- *3D mesh accuracy metrics.* Distance measures between the ground-truth mesh and the reconstructed mesh, such as **vertex-wise Euclidean distance**, and **Chamfer distance**. They require 3D annotations.
- *Surface normals metrics.* Provide a measure of the similarity of the reconstructed 3D model with the actual one by comparing the orientation of its normals. Such metrics are, for instance, **angular error**, and **cosine similarity**.
- *Foreground rendering metrics.* Focus on the evaluation of the accuracy in the reconstruction of the foreground. They include **foreground-background separation accuracy** and **foreground mask Intersection over Union (IoU)**.
- *Subject mask accuracy metrics.* Similarly to the previous metrics, they compute the similarity of the reconstruction and the original image restricted on the subject mask, such as **pixel-wise accuracy** and **subject mask Intersection over Union (IoU)**.

In this work, we will be making use of rendering quality metrics, as they do not require 3D annotations. In particular, for our evaluation we adopt **SSIM** and **PSNR** defined as

$$\mathbf{SSIM}(GS_1, GS_2) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

$$\mathbf{PSNR}(CI_1, CI_2) = 10 \cdot \log_{10}\left(\frac{\mathrm{L}^2}{\mathrm{MSE}}\right)$$

where GS_1, GS_2 are the greyscale images, CI_1, CI_2 are the color images, and

μ_x, μ_y : average luminance values of images x and y,
σ_x^2, σ_y^2 : variances of images x and y,
σ_{xy} : covariance between images x and y,
C_1, C_2 : constants to stabilize the division, chosen as $C_1 = (k_1 L)^2, C_2 = (k_2 L)^2$,
L : dynamic range of pixel values (e.g., 255 for 8-bit images),
k_1, k_2 : small positive constants,
$\mathrm{MSE} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(CI_1(i,j) - CI_2(i,j))^2$

we also define them on the mask-conditioned images

$$\mathbf{m\_SSIM}(GS_1, GS_2) = \mathbf{SSIM}(GS_1|_{\mathrm{mask}}, GS_2|_{\mathrm{mask}})$$

$$\mathbf{m\_PSNR}(CI_1, CI_2) = \mathbf{PSNR}(CI_1|_{\mathrm{mask}}, CI_2|_{\mathrm{mask}})$$

It is important to note that these two metrics can differ in terms of computed values depending on the way they are defined. We choose to define them one on the greyscale images and one on the colored images to have a slightly more comprehensive evaluation, as the two metrics are oftentimes considerably correlated.

2.4 Supervised vs Unsupervised

It is pivotal for this work the use of self-supervised learning for the training of the models proposed. The reasons for doing so have already been pointed out, as choosing this approach allows the training needlessly of 3D annotation. This gives access to a much larger amount of data, invaluable for generalization, but at the same time, it poses additional challenges in defining suitable objective functions and possibly weakens the robustness of the model.
In this field of research, the 3D reconstruction of the subject is oftentimes performed by optimizing the separation of the foreground from the background, either in 2 or 3 dimensions. In general, the reconstruction is performed employing different optimization functions. A form of rgb loss can be always used, as the rgb image is available regardless of the annotations. However, apart from this contribution, if the 3D ground-truth information is not available, the remaining possibilities are physical and regularization losses. The former is useful for imposing strong physical conditions to be satisfied on the reconstructed surface, at the cost of being computationally expensive and increasing the training time. The latter can be tailored easily depending on the problem and they can be kept computationally not onerous, but they do not necessarily improve the training and results of the overall model. This type of loss constitutes one of the major challenges of unsupervised training, requiring extensive experimenting and fine-tuning of the parameters.

Chapter 3

Video Reconstruction Module Baseline

In this chapter, we present the baseline we adopted for the video reconstruction module. We begin our work from the research of [4], following their proposed approach to perform a self-supervised 3D reconstruction of the video through 3D scene separation. In the following sections, we present the scope of the project, the reasoning behind this approach, and the architecture of the model.

3.1 Scope of the Project

With their work, [4], they propose a model that aims at reconstructing the 3D avatar from a monocular video using self-supervised techniques. Their approach allows for the model to learn the 3D properties of the scene, relying purely on the 2D frames of the video.
While the suggested techniques obtain impressive results in terms of the quality of the reconstruction, the limitations in terms of possible applications are not few. In fact, the model needs to be trained on the specific video the user seeks to reconstruct, for which the fitting time is considerable (~24h/48h) and the needed resources to achieve this 'speed' are significant (NVIDIA GeForce RTX 3090, 24GB).
On the other hand, the approach they propose relies on the lowest amount of information possible (monocular videos, absence of depth information), and it is independent of 3D annotations on the data, which are often inaccurate, due to the numerous challenges in measurement calibration and consistency.

3.2 Operating Principles

This approach directly tackles the challenges of scene decomposition and surface reconstruction in 3D. This is accomplished by implicitly modeling both the human subject and the background within the scene. These components are parameterized using distinct neural fields, which are learned concurrently from images to create a unified representation of the entire scene. The adopted objectives aim to address the inherent ambiguity between body and scene parts in contact and to enhance surface delineation. These objectives leverage the dynamically evolving human shape in canonical space to regularize the ray opacity, ensuring clearer distinction between different surfaces.
Specifically, the 3D geometry and texture of clothed humans are represented as a pose-conditioned implicit signed-distance field (SDF) and texture field in canonical space. Simultaneously, the background is modeled using a separate neural radiance field (NeRF). The human shape and appearance fields, in conjunction with the

background field, are learned from images through differentiable composited neural volume rendering techniques. Additionally, the dynamically updated canonical human shape is used to refine the ray opacities further. The training approach is formulated as a global optimization process, where the dynamic foreground and static background fields, along with the per-frame pose parameters are jointly optimized.

3.3 Dataset preprocessing

The model relies on a preprocessing of the data to obtain initial SMPL [17] body shape and pose estimations. This is done through the use of a pretrained ROMP model [18] in combination with OpenPose [19], [20], [21], [22]. Through these initial estimates obtained on inference over the raw video frames, the video to be reconstructed is provided with additional information regarding

- frame masks (the outline of the subject in each frame)
- camera intrinsics (estimates of the normalized 3D location, field of view etc.)
- SMPL parameters (parameters estimates of the SMPL rig), including:
    1. joint's position
    2. joint's rotation
    3. mean shape

These estimates and parameters are used for sampling and projecting points in the 3D space, and are further refined during training, as described in the following sections.

3.4 Architecture of the Model

Following this approach, the process begins with a ray denoted as r emanating from the camera center o with a direction v. Along this ray, points are densely sampled (x_d) and coarsely sampled (x_b) within the spherical inner volume and outer volume respectively. The points sampled within the foreground sphere are projected into a canonical space using inverse warping. The Signed Distance Function (SDF) of these canonical correspondences x_c is evaluated using the canonical shape network f^H_{sdf}. The spatial gradient of the sampled points in the deformed space is computed and concatenated with the canonical points x_c, the pose parameters θ, and the extracted geometry feature vectors z. This combined information serves as input for the canonical texture network f^H_{rgb}, which predicts color values for the canonical points x_c. Surface-based volume rendering is applied for the dynamic foreground, while standard volume rendering is used for the background. The resulting foreground and background components are composited to obtain the final pixel color.
To train the model, a loss function L is minimized, which measures the discrepancy between the predicted color values and the observed image data. Alongside it, different scene decomposition objectives are incorporated into the optimization process.

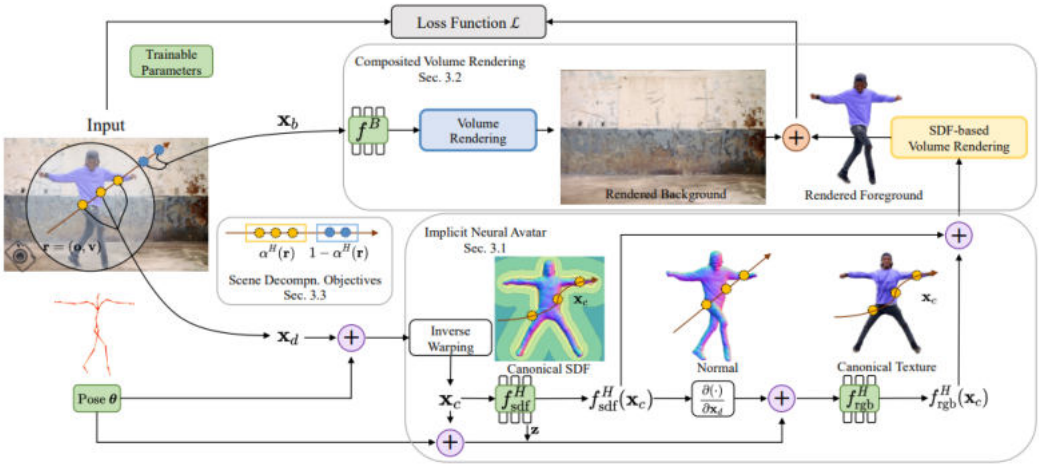

FIGURE 3.1: vid2avatar model diagram (image from [4])

3.5 Self-Supervised Optimization

The optimization is handled by combining four different loss functions, each responsible for different objectives. The full loss function is

$$\mathcal{L}(\Theta) = \sum_{i=1}^{F} \mathcal{L}^i_{\text{rgb}}(\Theta^H, \Theta^B) + \lambda_{\text{dec}} \mathcal{L}^i_{\text{dec}}(\Theta^H) + \lambda_{\text{eik}} \mathcal{L}^i_{\text{eik}}(\Theta^H) \tag{3.1}$$

where, Θ^H and Θ^B represent the sets of optimized parameters for the human and background models, respectively. Θ^H encompasses the weights of the shape network (Θ^H_{sdf}), the texture network (Θ^H_{rgb}), and the per-frame pose parameters (θ_i). On the other hand, Θ^B comprises the weights of the background density and radiance networks. These terms are divided into global optimization objectives and scene decomposition objectives.

3.5.1 Global optimization objectives

Eikonal Loss. As proposed by [16], the term $\mathcal{L}^i_{eik}$ is used to force the shape network f^H_{sdf} to satisfy the Eikonal equation in canonical space:

$$\mathcal{L}^i_{\text{eik}} = \mathbb{E}_{x_c}(\|\nabla f^H_{sdf}(x_c)\| - 1)^2 \tag{3.2}$$

Reconstruction Loss. This loss forces f^H_{rgb} to obtain corresponding rendering images similar to each frame. It corresponds to the L^1 distance between the pixel's rendered color $C(r)$ and the pixel's RGB color $\hat{C}(r)$

$$\mathcal{L}^i_{\text{rgb}} = \frac{1}{|\mathcal{R}^i|} \sum_{r \in \mathcal{R}^i} |C(r) - \hat{C}(r)| \tag{3.3}$$

3.5.2 Scene decomposition objectives

Opacity Sparseness Regularization. To enforce regularization on ray opacity using the dynamically updated human shape in canonical space, a technique is employed

that involves warping sampled points into the canonical space and determining the signed distance to the human shape. Subsequently, non-zero ray opacities for rays that do not intersect with the subject are penalized. This specific set of rays is denoted as $\mathcal{R}^i_{\text{off}}$ for each frame i. This approach ensures that rays not intersecting with the human subject have their opacities regulated, contributing to a more accurate and coherent representation of the scene.

$$\mathcal{L}^i_{\text{sparse}} = \frac{1}{|\mathcal{R}^i_{\text{off}}|} \sum_{r \in \mathcal{R}^i_{\text{off}}} |\alpha^H(r)| \tag{3.4}$$

A conservative approach in updating the Signed Distance Function (SDF) of the human shape consistently is adopted across the entire training process. This meticulous updating strategy ensures a precise alignment of human and background rays, contributing to the accuracy and reliability of the model's representations.

Self-supervised Ray Classification. Despite the shape regularization introduced in equation 3.4, the human fields tend to model portions of the background. This behavior arises from the inherent flexibility and expressive power of Multi-Layer Perceptrons (MLPs), particularly when the subject is in contact with the scene. To address this issue and enhance the distinction between the dynamic foreground and background, an additional loss term is incorporated. This term encourages ray distributions containing either fully transparent or fully opaque rays, further refining the separation between the dynamic foreground and the background components.

$$\mathcal{L}^i_{\text{BCE}} = -\frac{1}{|\mathcal{R}^i|} \sum_{r \in \mathcal{R}^i} (\alpha^H(r) \log(\alpha^H(r))) + (1 - \alpha^H(r)) \log(1 - \alpha^H(r)) \tag{3.5}$$

The introduced term penalizes deviations of ray opacities from a binary 0, 1 distribution through the binary cross-entropy loss. Essentially, this encourages opacities to be zero for rays hitting the background and one for those intersecting with the human shape. This approach intuitively guides the model to distinguish clearly between foreground and background elements.
The final formulation of the scene decomposition loss, denoted as $\mathcal{L}_{\text{dec}}$, is given by:

$$\mathcal{L}^i_{\text{dec}} = \lambda_{\text{BCE}} \mathcal{L}_{\text{BCE}} + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}} \tag{3.6}$$

Chapter 4

Video Reconstruction Module Experiments and Evaluation

Starting from the Vid2Avatar model, we experiment with different modifications in order to increase the quality of the reconstruction. In this phase, we remain focussed on the video-specific reconstruction, in order to observe what changes can benefit the reconstruction in terms of training objectives, surface regularization, time consistency, and sampling techniques. In the following, we present some of the performed experiments.

4.1 Experiments

Hyperparameter tuning. We dedicate some experiments to hyperparameter searches, aiming at some tuning to obtain even so slight improvements in terms of training behavior and the quality of the reconstruction. Furthermore, we do so in order to observe how the training of the model is affected, considering the partial unpredictability of unsupervised training with a composite loss. Most importantly, in hindsight, different learning rates and the number of samples have a focal role.

Time consistency. We attempted to implement an additional regularization loss responsible for imposing time consistency on the canonical surface, through the different frames. To do so, we experimented with two approaches

- *Displacement regularization.* With this, we imposed as an objective to have small values of the norm of the first differences between the canonical surface points' coordinates

$$\mathcal{L}_{\text{time\_cons}} = \sum_{c \in C} \min_{c' \in C'} \left(d(c, c') \right)$$

  where C and C' are the sets of canonical surface points and previous canonical surface points.

- *Sdf values regularization.* With in mind the idea of reducing the constraints on the sampling of the model, while still imposing some consistency of the representation over time, we imposed as an objective to have similar signed distance values on the sampled points across frames

$$\mathcal{L}_{\text{sdf\_time\_cons}} = \text{MSE}(S, S') = \frac{1}{n} \sum_i (S_i - S'_i)^2$$

  where S and S' are the sets of signed distance values of the sampled points, and the previous signed distance values on those points.

Incremental sampling. One of the limitations of models making use of implicit neural networks is the demanding memory requirements for the sampling of the 3D points and the inferring on the implicit network. There are two sampling processes taking place, one that is responsible for shooting a number of rays N_{rays}, and one that samples point on each ray $N_{\text{samples\_x\_ray}}$. The latter makes use of the current state of the signed distance function to error-adjust the sampling.
Hence, we leverage this by starting with a larger $N_{\text{samples\_x\_ray}}$, and then reducing it in favor of a broader set of rays, thus increasing N_{rays} and reducing accordingly $N_{\text{samples\_x\_ray}}$ during training. We experiment with doing so by using two different incremental profiles

- *Linear.* N_{rays} is incremented by 1024, while $N_{\text{samples\_x\_ray}}$ is reduced by 32, corresponding to a quarter of its initial value of 128. This is done to maintain almost constant the required amount of memory during training.
- *Squared.* N_{rays} doubles, while $N_{\text{samples\_x\_ray}}$ halves. This does not necessarily maintain constant the required amount of memory during training.

In the next section, we present the results obtained from the above-discussed experiments.

4.2 Evaluation

We perform quantitative and qualitative evaluations on a video of the NeuMan dataset. Since 3D annotations are not available, for the quantitative evaluation we rely on SSIM and PSNR as metrics. Furthermore, we introduce the m_SSIM and m_PSNR metrics, which are the SSIM and PSNR computed on the reconstructed and original images, with the application of the subject mask. These two metrics are introduced as the model is focussed on the reconstruction of the subject, and later in this work, we will restrict the focus on the subject even more.
Here we include the tables with the quantitative evaluation as well as the qualitative images for some of the frames. To draw conclusions, we also observe and analyze the loss behavior through the use of a *Weights&Biases* API, and we analyze the evolution of the reconstructed canonical representation during training.

4.2.1 Quantitative evaluation

Hyperparameter tuning. Starting with the customary hyperparameter searches, we experiment with tuning a few of the hyperparameters to see how the training is affected. In particular, here we include the results regarding the use of different learning rates and numbers of samples.

Experiment	SSIM ↑	PSNR ↑	m_SSIM ↑	m_PSNR ↑
Baseline (pre-trained)	0.714	26.644	0.993	42.489
Baseline	0.708	26.310	**0.994**	44.232
Incremented sampling	0.711	26.437	**0.995**	**46.373**
Learning rate 1.0e-5	0.625	17.218	0.994	33.543
Learning rate 5.0e-5	0.669	21.935	0.987	38.912
Learning rate 1.0e-4	0.686	24.057	0.991	42.930
Learning rate 1.0e-3	**0.716**	**26.703**	**0.996**	**46.636**
Learning rate 5.0e-3	0.682	23.661	0.994	42.397

TABLE 4.1: Hyperparameters tuning - quantitative evaluation on NeuMan.

The results presented in table 4.1 show how an increased number of samples in terms of the range of rays employed, N_{rays} from 512 (baseline) to 1024, results in a slight improvement in the quality of the reconstruction. Experimenting with larger numbers posed some challenges in terms of available vram, for which we proposed the use of incremental sampling. In terms of learning rate, we see that the best results were achieved using a value of 1.0e-3, improving slightly on all the metrics.

Time consistency. We attempt to include an additional regularization loss term, aiming at a more consistent surface representation by conditioning the value of the implicit signed distance function on the ones obtained on previous frames.

Experiment	SSIM ↑	PSNR ↑	m_SSIM ↑	m_PSNR ↑
Baseline (pre-trained)	**0.714**	**26.644**	0.993	42.489
Baseline	0.708	26.310	**0.994**	44.232
Displacement regularization	**0.714**	26.539	**0.994**	**45.203**
Sdf values regularization 1[1]	0.703	25.795	0.989	38.814
Sdf values regularization 2[2]	0.703	25.538	0.900	39.576

TABLE 4.2: Time consistency - quantitative evaluation on NeuMan.

From table 4.2 it is clear that the additional loss terms do not contribute particularly to the training of the model. The use of the displacement regularization loss yielded some improvements in the mask-conditioned psnr metric. However, considering the slight increase in time per optimization step, this gain is not enough to justify its adoption in future experiments.

Incremental sampling. We experiment with a broadening of the sampling during training, as described in the previous section. In doing so, some attention has to be given to the balancing of the number of epochs after which the model switches to a broader number of samples, at the expense of the number of samples on each ray. Here we include the results for 800 epochs. We chose such a number as, from the training behavior of previous runs employing a lower number of epochs, we observed that the model was not able to learn a good enough implicit representation to aid the search for the sampling on each ray, while 800 epochs provide a more stable training.

[1] constant loss regularization weight
[2] incremental loss regularization weight

Experiment	SSIM ↑	PSNR ↑	m_SSIM ↑	m_PSNR ↑
Baseline (pre-trained)	0.714	26.644	0.993	42.489
Baseline	0.708	26.310	0.994	44.232
Squared IS profile (2x800epochs)	0.715	26.746	0.994	44.472
Linear IS profile (2x800epochs)	**0.726**	**27.548**	**0.997**	**47.347**

TABLE 4.3: Incremental sampling - quantitative evaluation on NeuMan.

As we can see from table 4.3, a squared increment profile seems to be a little ambitious, while a linear increment profile results in a much smoother training curve, as it can be observed from figure 4.1, and overall better reconstruction.

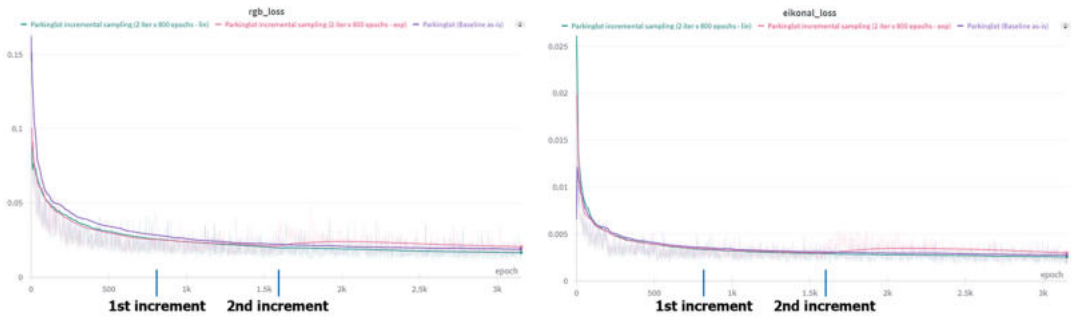

FIGURE 4.1: Incremental sampling loss charts

4.2.2 Qualitative evaluation

Qualitative results: reconstruction of a video in the Neumann dataset. Including the original frame, the reconstructed rendering, and the computed normals.

FIGURE 4.2: Baseline using the provided pre-trained model checkpoint

FIGURE 4.3: Baseline trained from scratch

FIGURE 4.4: Increased sampling N_{rays}

FIGURE 4.5: Learning rate 1.0e-5

FIGURE 4.6: Learning rate 5.0e-5

FIGURE 4.7: Learning rate 1.0e-4

FIGURE 4.8: Learning rate 1.0e-3

FIGURE 4.9: Learning rate 5.0e-3

FIGURE 4.10: Time consistency loss: displacement regularization

FIGURE 4.11: Time consistency loss: sdf values soft regularization

FIGURE 4.12: Time consistency loss: sdf values soft regularization incremental weight

FIGURE 4.13: Incremental sampling squared profile

FIGURE 4.14: Incremental sampling linear profile

Chapter 5

AGen model baseline

In this chapter, we introduce our novel model, the AGen model. The model leverages the previously introduced video reconstruction model to generalize the 3D reconstruction of monocular videos, and it does so via self-supervised learning, allowing it to be scalable and independent of accurate, precise, and consistent 3D annotated datasets. It aims at providing a reliable inference-time reconstruction, with the possibility of much faster optimization over never-seen videos to produce more polished results.

5.1 Scope of the Project

The model is proposed as a solution to perform 3D reconstruction from monocular videos at inference time. It aims at solving or improving the limitations pointed out in section 3.1, in particular the time constraints. The training of the model is performed via self-supervised learning to deal with the issues mentioned in section 2.3 and to provide a model that is extremely flexible in terms of scalability and generalizability. We can state our objectives as

Definition 5.1.1. *(Scope of the project) Given as input a series of frames from a monocular video, the model shall be able to yield*

1. *Inference-time 3D avatar reconstruction*
2. *Short-time optimized 3D avatar reconstruction*

It goes without saying that the first objective constitutes a much more difficult task, as it seeks a reconstruction with minimal information and extreme time requirements. However, it should be considered as the most desirable result we aim to obtain. For this reason, the second objective is included, as it is more achievable and it still represents a valuable result in terms of applications.

5.2 Architecture of the Model

The model leverages the previously described approach to be trained in a self-supervised manner over each video of the trainset. However, trying to generalize the 3D reconstruction of any video as a whole would be, if not an ill-posed problem, for sure an incredibly challenging one. For this reason, we initialize general models only for the foreground, leaving the background models used during training as video-specific models. These are trained in the 3D scene separation to aid the training of the foreground networks.

Hence, we have two sets of networks

generalizable networks:
- implicit network
- rendering network

video-specific networks:
- background implicit network
- background rendering network

The generalizable models are trained over all the videos in the training dataset, while for video-specific models different instances are used for each video, reloading their previous states each epoch to aid the training.

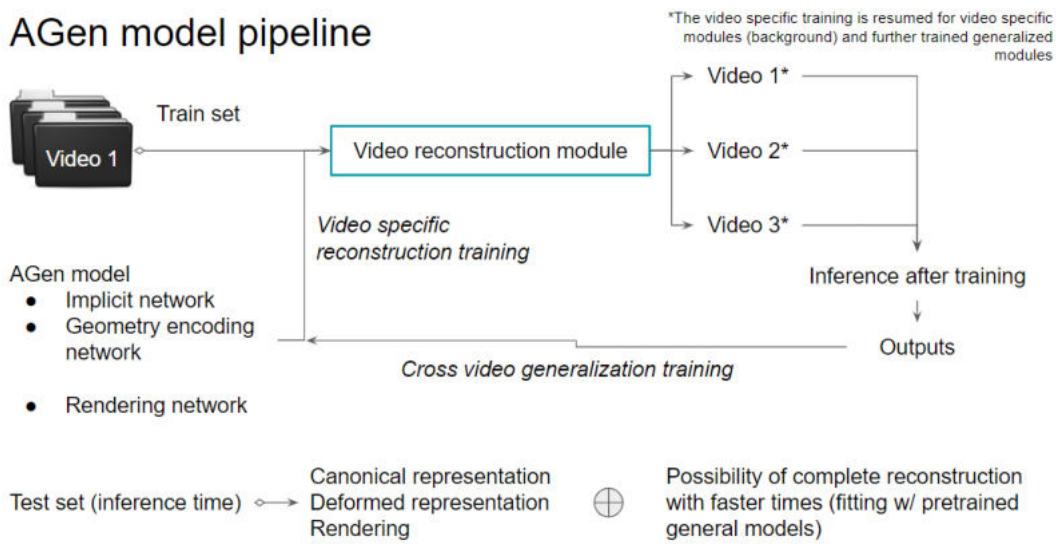

FIGURE 5.1: AGen model diagram

To define our baseline, we consider the model without any particular additional implementation with respect to the video reconstruction module. In other words, the baseline essentially implements the training of different instances of the video reconstruction module over each video, passing the generalizable networks, and loading previous states of the video-specific networks.

Observation 5.2.1. *It is important to note that, in principle, it would be more appropriate to assemble all the frames of the different videos into one unique trainset. However, we choose to maintain this distinction for experimental purposes, in order to be able to better analyze the differences in the training process between different experiments.*

5.3 Training

Training is performed on a multi-gpu setting, parallelizing the training of the instances of the video reconstruction module over each video. To generalize the process, and avoid unbalanced training over the trainset, the model is fit over each batch for a time limit, as videos in the trainset can differ a lot in terms of the number of frames.

5.4 Leveraging the pretraining of the implicit network

Leveraging the training of the implicit network is not a trivial problem. By nature, neural implicit networks are expensive at inference, since they require point search algorithms to extrapolate the implicitly defined surface. To achieve this, there are different options, each requiring a different amount of computational time. Each one of these clearly heavily relies on the fact that the pre-learned implicit function is representative enough of the not previously seen surface.

1. *Pre-trained sdf for ErrorBoundSampling.* This approach does not slow down inference or the optimization step when adopted in the video reconstruction module, as the implicit signed distance function is regularly used for sampling. However, this is not optimal since the sampling is still performed from the frame to the 3D canonical space, and not vice-versa, losing potentially valuable information carried by the pretrained implicit representation.

2. *Canonical surface sampling.* We perform the sampling from the canonical surface, instead of the image, and perform an inverse warp to obtain the corresponding 2D points. Such an inference step is computationally equivalent to the surface reconstruction necessary for inference, and it is considerably demanding in terms of computational time. Even though it has the potential to considerably speed up the optimization of the reconstruction, there is no guarantee that dedicating that time to more reconstruction optimization steps would yield much better results.

3. *Uniform/Gaussian inference sampling.* Another approach is to perform the sampling starting from a faster search algorithm on the implicit representation, reducing the computational time and allowing for more error in the distance from the actual 0-surface. These estimates can be used as 'educated guesses' for the sampling by scattering points around the estimates either uniformly or including Gaussian noise.

5.5 Evaluation

In this setting, the evaluation is perhaps the most delicate matter. Since the training is performed in a self-supervised manner, on the one hand, the optimization of the loss functions is by itself an indicator of how well the model is training. On the other hand, it is advisable to have some metrics to validate the quality of the reconstruction during training. At the same time, it is necessary to validate the results over the validation dataset including never-seen data. However, we must keep in mind the computational burden represented by these operations in this setting, which can slow down training considerably. For these reasons, we define the following validation procedures.

Intra-training, data-agnostic validation: 2D metrics
For this kind of validation, we implement metrics that do not rely on having the availability of 3D annotations in the data and can be computed for any selected dataset. These metrics are **structural similarity index measure (SSIM)** and **peak signal-to-noise ratio (PSNR)**, along with their respective mask-conditioned values.

1. *Trainset validation.* After the fitting of the video reconstruction module over a batch (video), we perform one validation step of the module.

2. *Validset inference-time validation.* Every 5 epochs, over the whole trainset, we perform a validation step of the AGen model, over the never-seen data of the validset. We do so without any fitting of the reconstruction, meaning we impose the shortest possible inference time.

3. *Validset short-time validation.* After training is finished, we perform a validation step of the AGen model, over the never-seen data of the validset. We do so by fitting the reconstruction for a short time, to optimize and enhance the reconstruction.

Test method validation: inference after training
In general, we propose our model so that it would be able to infer never-seen data with the following methods

1. *Inference time/short-time.* Either perform inference without refinement or perform a short-time refinement of the networks on the video to be reconstructed.

2. *Tiny/reduced/full.* The 'tiny' option is used to perform the reconstruction over one of the frames; it is used for fast validation during training to supervise the behavior of the model. The 'reduced' option is used to evaluate the reconstruction and metrics over 42 uniformly sampled frames. The latter, the 'full' option, is used to reconstruct all the frames, and it is for obvious reasons considerably more demanding in terms of computational time.

3. *Pretrained/non-pretrained.* Either to use or not pre-trained networks. This is intended for the experimental phase, to assert the improvements.

5.6 Dataset

Selecting one dataset is not a straightforward choice in this area, as other research is not always consistent in the choice, oftentimes preferring the creation of a tailored dataset for the training of their model. Also, it is not given that the most recent datasets have been made available.
For our experimenting and proof of concept, we adopt a subset of the 3DPW dataset, [6], which is comprised of videos of subjects in the wild from a single moving camera. We select this dataset because of the variety of subjects, actions, and backgrounds. However, we select a subset as a number of videos contained are out-of-scope for our problem, for instance, they include multiple subjects in the scene.

Chapter 6

AGen Model Experiments and Evaluation

In this chapter we present the results of some experiments performed on the AGen model, starting from the baseline, as defined in section 5.2, leading up to the inclusion of a geometry encoding network aiming at a better generalization for the model. In this phase, we are mainly concerned about improving the generalization capabilities of the model. We do so by comparing the qualitative and quantitative evaluation of the reconstruction on previously unseen videos, as well as monitoring the training of the model. The latter is an important indicator as the behavior of the model right after the change of the video being reconstructed shows how the model is able to leverage the training over previous data to quickly improve on the next video.

6.1 Experiments

Agen model baseline. We begin our experimentation process by evaluating the results of our baseline model on never-seen data, compared with the results obtained with the straightforward reconstruction using the reconstruction module.

Geometry encoding network. Secondly, we focus our effort on the implementation of a geometry encoding network with the objective of improving the generalization of the model. The task of this network is twofold:

1. Improve the overall reconstruction by picking up on garment details such as wrinkles and fabric by leveraging the frame embedding.

2. Aid the training of the implicit sdf model on never-seen data by applying a frame-conditioned 3D displacement to the sampled points.

This network is comprised of a first part, the encoder, which is trained to take as input the image frame and output an embedding. The network then takes as input the canonical points, meaning the sampled points warped into the canonical space, it concatenates them to the frame embedding and passes them through some trainable fully connected layers to output a new 3D point.

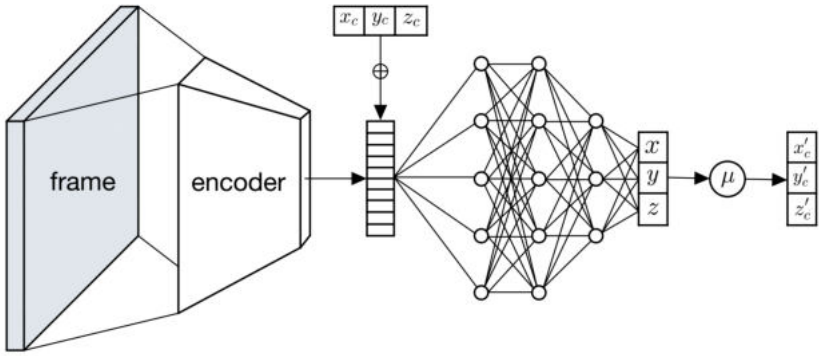

FIGURE 6.1: Geometry encoding network diagram

This architecture, while being reasonable to achieve our task, immediately presents one important inconvenience, as the initial point coordinates outputted by the network will likely be outside the range of our normalization. To ensure this does not happen while avoiding the loss of potential trainability of the network, we use the outputted coordinates to perform a displacement of the canonical points, tempered by a morphing factor μ.

$$x' = x_c + \mu(x - x_c)$$

This allows the initialization of the model with outputs close to the inputs while leaving the freedom of increasing the displacement during training overcoming the morphing factor if necessary to improve on the other losses.
Furthermore, we implement an optimization function as a regularization term to avoid the explosion of some displacements, resulting in out-of-normalization bounds points, and thus NaNs. We first experiment with a simple loss, aiming to contain **all** the displacements

$$\mathcal{L}^{\text{v1}}_{\text{geometry\_morphing\_loss}} = \sum_{i \in S} \|x'_i - x_{c,i}\|$$

where S is the set of sampled canonical points. Then, we also experiment with containing only the **average** displacement, thus allowing more for larger displacements on specific points or regions

$$\mathcal{L}^{\text{v2}}_{\text{geometry\_morphing\_loss}} = \frac{1}{N_{\text{samples}}} \sum_{i \in S} \|x'_i - x_{c,i}\|$$

Now, to obtain the frame embedding we experiment with two different approaches, using a pre-trained image classification model to extract a general frame embedding, or training a Unet-like encoder to output a pixel-aligned embedding.

Pre-trained backbone. Using a pre-trained backbone, losing the last classification layer, and possibly even a pooling layer, to obtain an embedding of the image. The fact that the embedding is general for the whole frame is slightly reductive but allows for faster computation time on the trainer step, as the embeddings can be obtained during preprocessing. However, if the backbone has to be trained, then this is a corner that cannot be cut and the computational cost is still considerable (making it less efficient than the second approach).

Unet encoder. Training an Unet encoder, from the frame image we obtain a pixel-aligned embedding, of dimensions [image_height, image_width, embedding_dim]. This can be used to concatenate to each canonical point the embedding corresponding to the pixel intersected by the ray on which the point was sampled.

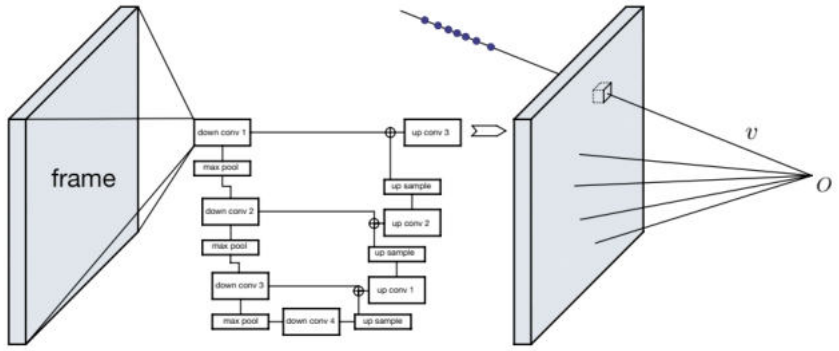

FIGURE 6.2: Unet encoder diagram

6.2 Evaluation

6.2.1 Quantitative evaluation

Here we present the results obtained by experimenting with the *short-time validation* using 10 refinement epochs. It is important to note that 10 epochs is an extremely low number of epochs for the video reconstruction module to obtain good results, as it was intended by the authors to be trained for around 6000 epochs. However, we select such a low number of epochs for two main reasons. Firstly, we are particularly interested to see if the training over the reconstruction of other videos can be leveraged by the model to speed up the training. Secondly, the amount of frames included in the 3DPW sequences is much higher, and it thus requires a much longer training time, making it more difficult to perform multiple experiments with longer-time reconstructions.

AGen model baseline. Regarding the quantitative evaluation of the baseline against the straightforward video reconstruction using the video reconstruction module we obtain the results included in table 6.1.

Experiment	**SSIM** ↑	**PSNR** ↑	**m_SSIM** ↑	**m_PSNR** ↑
Reconstruction module	0.293	14.889	0.974	27.204
AGen baseline	0.293	**14.901**	**0.975**	**27.855**

TABLE 6.1: AGen model baseline - quantitative evaluation on 3DPW.

As we see the baseline achieves ever so slightly better results on three of the four metrics. However, we have a considerable increase in the quality of the reconstruction, as can be observed from the qualitative evaluation in figures 6.5 and 6.6. As a matter of fact, the reconstruction presents different and persistent erroneous constructs, while the AGen model does not. This is due to the pretraining of the implicit

network which is able to aid the reconstruction even in the first few epochs.

Geometry encoding network. Here we present some of the results obtained while experimenting with the geometry encoding network. In particular, table 6.2 includes the results for the network equipped with a trainable Unet encoder, using different loss regularization weights and adopting the two different geometry morphing losses. This experimenting phase has been extensive, but it presented multiple technicalities and issues related to the code implementation. For this reason, we only include some of the more interesting results, also considering the training behavior of each model. Furthermore, we choose not to include experiments made using a non-pixel-aligned pretrained backbone as an encoder, as they did not yield particularly interesting results, showing an almost unaffected training behavior.

Experiment	**SSIM** ↑	**PSNR** ↑	**m_SSIM** ↑	**m_PSNR** ↑
Reconstruction module	0.293	14.889	0.974	27.204
AGen baseline	0.293	14.901	**0.975**	**27.855**
geometry encoding net v1 w:0.1	(0.316)	(15.056)	(0.976)	(29.962)
geometry encoding net v1 w:10	0.291	14.911	0.973	27.315
geometry encoding net v2 w:10	0.293	**14.921**	0.974	27.549

TABLE 6.2: Geometry encoding network - quantitative evaluation on 3DPW.

From our experiments, the use of the geometry encoding network does not produce better results in terms of our metrics. The second version of the regularization loss with a large weight value produces slight improvements on the PSNR value, but not enough to be conclusive, as the mask-conditioned value worsens. From the qualitative results, we see the reconstruction is good, but it does not improve particularly. Observing the qualitative results of the model using a regularization weight of 0.1 in figure 6.7, we see the quality of the reconstruction suffers, showing multiple problems. This can be attributed to the use of a low regularization weight, leaving the displacements on the canonical points too unconstrained. The optimization of the geometry morphing loss, along with the rgb loss can be observed in figure 6.3, in which case the 3D reconstruction suffers as the model focuses on the optimization of the rendered image.

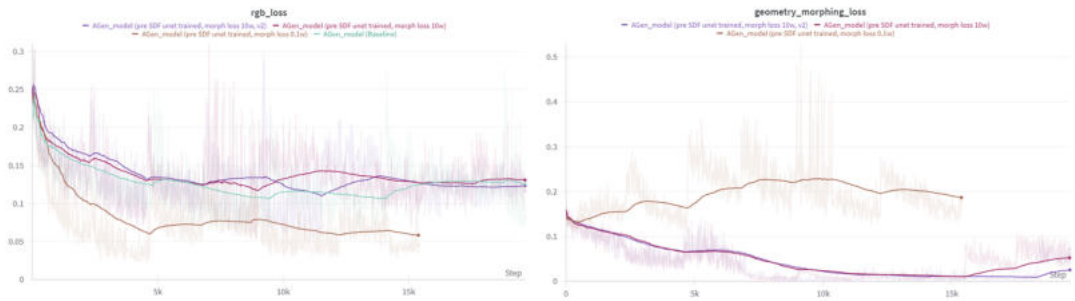

FIGURE 6.3: Geometry morphing loss

It is for this reason that we disregard the quantitative results obtained on the model with a regularization weight of 0.1. Furthermore, the use of the geometry morphing loss has proven to be necessary when the geometry encoding network is employed.

In fact, using low values on the regularization weight, or not adopting such a loss term at all, results in the network being unconstrained on the displacement of the points. In such a case, the network can learn that it is able to better reconstruct the subject in terms of different loss terms, in particular the rgb loss, if it projects the points on a plane, parallel to the image plane. When this happens, the model loses all capabilities of correctly training on the correct reconstruction of the surface and produces results similar to the ones shown in figure 6.4.

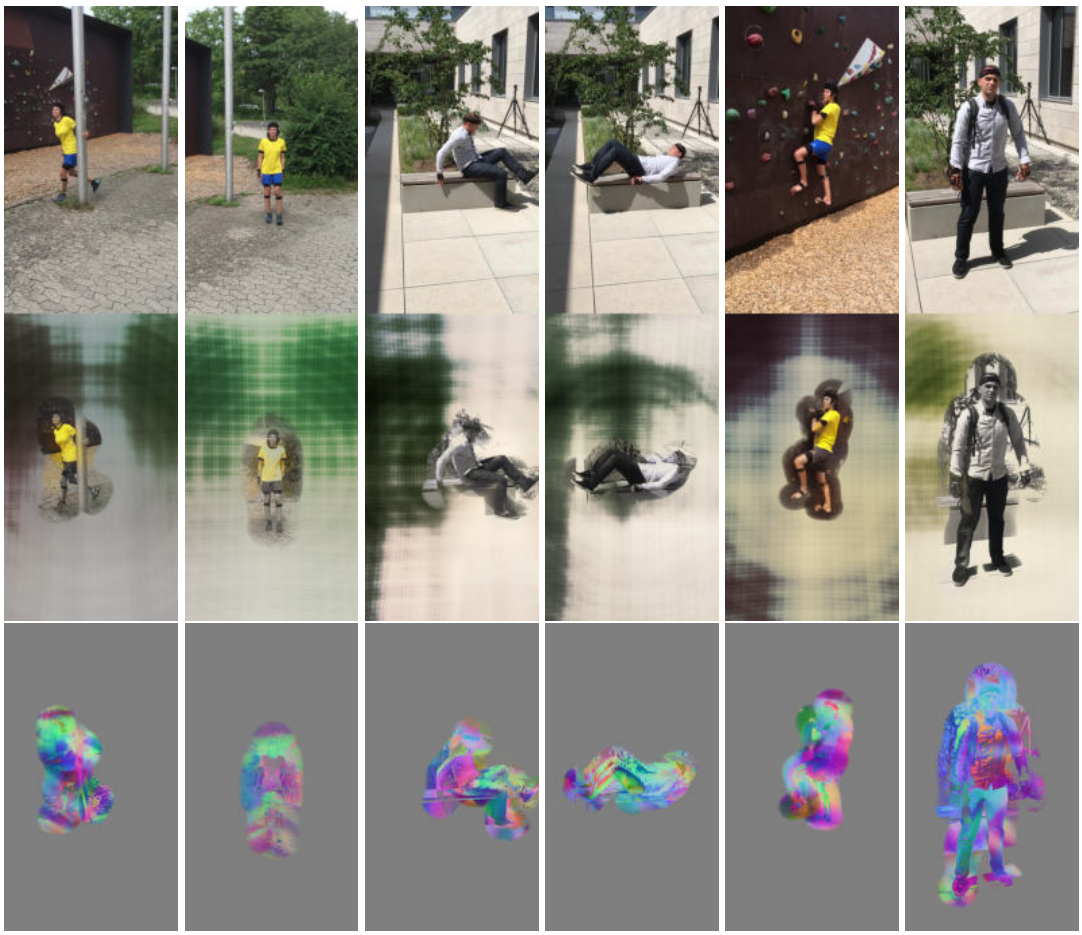

FIGURE 6.4: Collateral cases with unconstrained geometry encoding network

Fine-tuning from previous results. Finally, we experiment with adopting a higher learning rate and implementing the use of incremental sampling, as they had shown improving results during the previous experiments.

Experiment	SSIM ↑	PSNR ↑	m_SSIM ↑	m_PSNR ↑
Reconstruction module	0.293	14.889	0.974	27.204
AGen baseline	0.293	14.901	**0.975**	**27.855**
Incremental sampling	0.290	14.789	0.971	26.289
Geometry encoding[1]	0.291	14.779	0.973	27.297

TABLE 6.3: Fine tuning - quantitative evaluation on 3DPW.

[1]Geometry encoding network with incremental sampling and learning rate 1.0e-3

As it is clear from table 6.3, the use of a higher learning rate, in this case, does not aid the training of the model. While initially the training is faster, the learned representation from previous videos is quickly overcome with the training on newer data, resulting in a representation with poor generalization, as can be seen from the qualitative evaluation in figure 6.10 and figure 6.11. Also, the incremental sampling does not improve the results as it did for the video reconstruction module, as the error-bound sampling algorithm cannot leverage the implicit function as it did in the video-specific case.

6.2.2 Qualitative evaluation

Qualitative results: reconstruction of a video in the 3DPW testset. Including the original frame, the reconstructed rendering, and the computed normals.

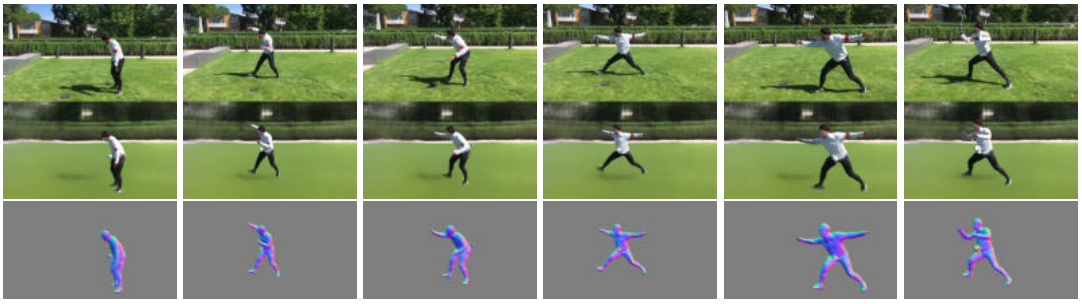

FIGURE 6.5: Reconstruction without pre-training

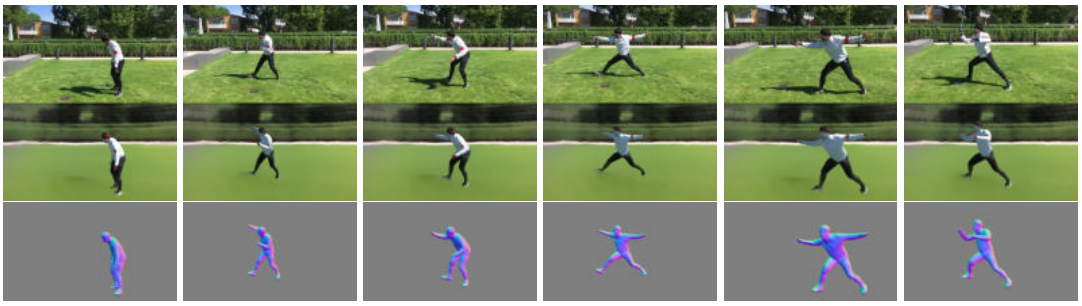

FIGURE 6.6: AGen model baseline

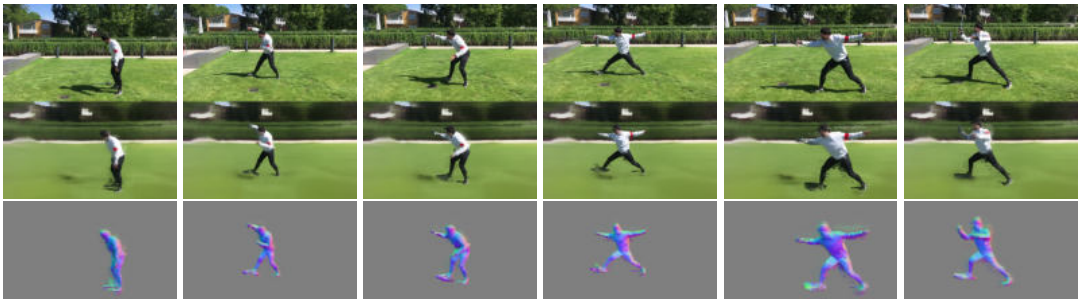

FIGURE 6.7: geometry encoding network with Unet encoder, morphing loss v1, weight 0.1

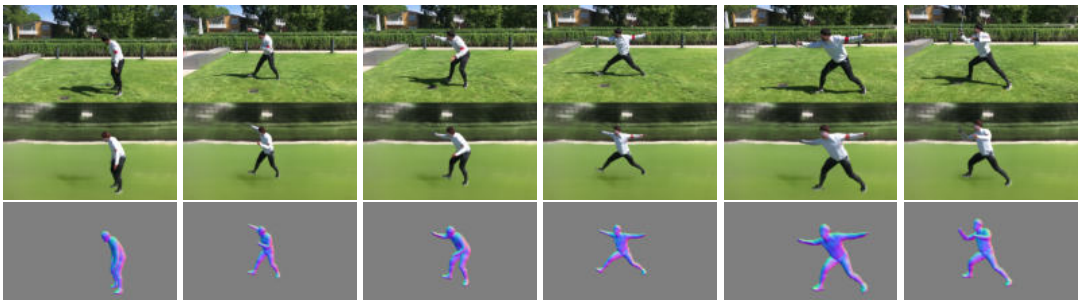

FIGURE 6.8: geometry encoding network with Unet encoder, morphing loss v1, weight 10

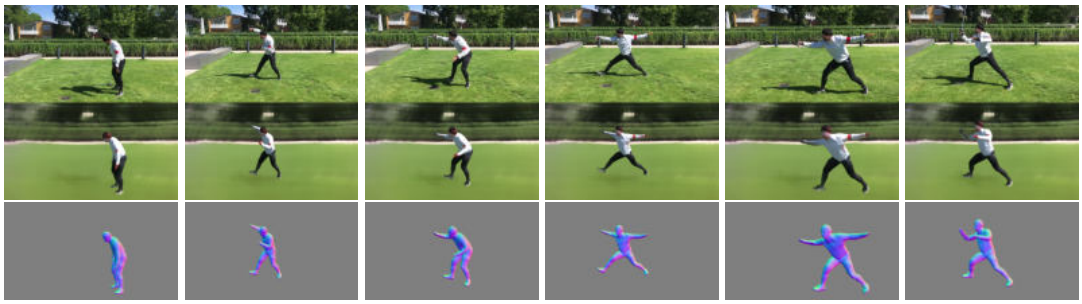

FIGURE 6.9: geometry encoding network with Unet encoder, morphing loss v2, weight 10

FIGURE 6.10: incremental sampling, with learning rate 1.e-3

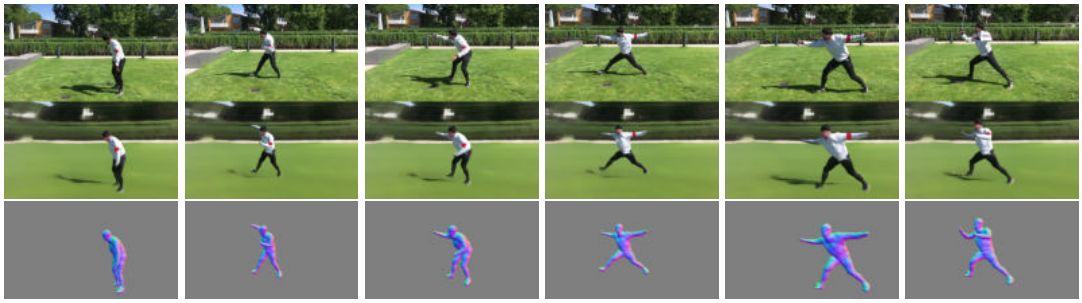

FIGURE 6.11: incremental sampling, geometry encoding network, with learning rate 1.e-3

Chapter 7

Conclusions and Future Directions

To conclude, and summarize the above-presented contributions, let us denote some results, considerations, and future efforts.

7.1 Considerations

The above-presented analysis shows the advantage of pretraining the implicit network over a dataset to leverage it during the reconstruction of previously unseen data, improving the quality of the reconstruction after a short refinement time. However, the generalization capabilities of the model are not exceptional. This can be mainly attributed to the use of a non-representative enough dataset, and the conditioning of the model.

- The former can be easily improved, as the data does not require 3D labeling. Preprocessing a larger dataset might allow for a much better implicit representation to be learned, thus improving the increase in the quality of the results. We chose to maintain the dataset more contained during this experimentation phase due to the training time requirements, which would have not allowed for a good number of experiments as the current state of the model already requires a considerable training time.

- The implementation of additional networks to condition the implicit representation and the rendering network would allow for better generalization of the model. The approach we proposed, making use of the geometry encoding network, while it did not produce considerable improvements, leaves room for enhancement. Different encoders, such as transformers, may be able to produce much better embeddings to aid the generalizability of the implicit representation.

7.2 Future directions

As some of the results of the AGen model, in particular in terms of qualitative results have been promising, there are a number of aspects that can be improved. To name a few

- The current state of the code is extremely inefficient as it was developed for research and proof of concept purposes. An effort to improve the code could allow for computational optimization and thus better training and results.

- The dataset composition can be improved both in terms of number of videos as well as quality. The adopted dataset includes videos that are particularly challenging to reconstruct, and not necessarily the most consistent in terms of

settings. While this is interesting to analyze the capabilities of the model, it probably hinders the ability to learn a common implicit representation. Using a more consistent and canonical dataset might improve the training considerably.

- Also, for the training of the model, it could be advantageous to reduce the number of frames in the included videos, to allow the model for a larger number of epochs and thus better training overall. The 3DPW dataset made use of high frame-per-second cameras; for this reason, the data might be more redundant than advisable for good training.
- Due to time constraints, we have not had the chance to experiment using other sampling approaches that could leverage the pretrained implicit representation more, as described in section 5.4. The implementation of such sampling techniques could allow for a faster refinement of the model on unseen data, and thus a better quality of the reconstruction.
- The model would benefit from the training of a more generalizable implicit representation. This could be achieved also by projecting to a higher dimensional canonical space, obviously losing the possibility of interpretation of the canonical representation.

7.3 Conclusions

Overall, we believe the model we have proposed shows capabilities of producing high-quality 3D avatar reconstruction with no requirement for additional information on the data and with faster times. A lot of improvements are still necessary to obtain a large advantage over a straight-forward non-pretrained reconstruction, as outlined in the previous section. However, considering the promising results, and numerous applications brought by such a model, we believe more efforts will soon bring forward an extremely interesting and valuable solution.

Bibliography

[1] Shunsuke Saito et al. *PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization*. 2019. arXiv: 1905.05172 [cs.CV].

[2] Yuliang Xiu et al. *ICON: Implicit Clothed humans Obtained from Normals*. 2022. arXiv: 2112.09127 [cs.CV].

[3] Thiemo Alldieck, Mihai Zanfir, and Cristian Sminchisescu. *Photorealistic Monocular 3D Reconstruction of Humans Wearing Clothing*. 2022. arXiv: 2204.08906 [cs.CV].

[4] Chen Guo et al. *Vid2Avatar: 3D Avatar Reconstruction from Videos in the Wild via Self-supervised Scene Decomposition*. 2023. arXiv: 2302.11566 [cs.CV].

[5] Wei Cheng et al. "DNA-Rendering: A Diverse Neural Actor Repository for High-Fidelity Human-centric Rendering". In: *arXiv preprint* arXiv:2307.10173 (2023).

[6] Timo von Marcard et al. "Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera". In: *European Conference on Computer Vision (ECCV)*. 2018.

[7] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. *CLOTH3D: Clothed 3D Humans*. 2020. arXiv: 1912.02792 [cs.CV].

[8] Wei Jiang et al. *NeuMan: Neural Human Radiance Field from a Single Video*. 2022. arXiv: 2203.12575 [cs.CV].

[9] Weipeng Xu et al. "MonoPerfCap: Human Performance Capture From Monocular Video". In: *ACM Trans. Graph.* 37.2 (May 2018), 27:1–27:15. ISSN: 0730-0301. DOI: 10.1145/3181973. URL: http://doi.acm.org/10.1145/3181973.

[10] Qianli Ma et al. "The Power of Points for Modeling Humans in Clothing". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 10974–10984.

[11] Shaohui Liu et al. "DIST: Rendering Deep Implicit Signed Distance Function With Differentiable Sphere Tracing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[12] Jianchuan Chen et al. *Pixel2ISDF: Implicit Signed Distance Fields based Human Body Model from Multi-view and Multi-pose Images*. 2022. arXiv: 2212.02765 [cs.CV].

[13] Baorui Ma et al. "Towards Better Gradient Consistency for Neural Signed Distance Functions via Level Set Alignment". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 17724–17734.

[14] Mihai Fieraru et al. "Three-dimensional reconstruction of human interactions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7214–7223.

[15] Mihai Fieraru et al. "Remips: Physically consistent 3d reconstruction of multiple interacting people under weak supervision". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 19385–19397.

[16] Amos Gropp et al. "Implicit geometric regularization for learning shapes". In: *arXiv preprint arXiv:2002.10099* (2020).

[17] Matthew Loper et al. "SMPL: A Skinned Multi-Person Linear Model". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.

[18] Yu Sun et al. "Monocular, One-stage, Regression of Multiple 3D People". In: *ICCV*. 2021.

[19] Z. Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[20] Tomas Simon et al. "Hand Keypoint Detection in Single Images using Multiview Bootstrapping". In: *CVPR*. 2017.

[21] Zhe Cao et al. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *CVPR*. 2017.

[22] Shih-En Wei et al. "Convolutional pose machines". In: *CVPR*. 2016.