MoAngelo: Motion-Aware Neural Surface Reconstruction for Dynamic Scenes

Mohamed Ebbed
University of Bonn
Lamarr Institute for Machine Learning
and Artificial Intelligence
mebbed@uni-bonn.de

Zorah Lähner
University of Bonn
Lamarr Institute for Machine Learning
and Artificial Intelligence
laehner@uni-bonn.de

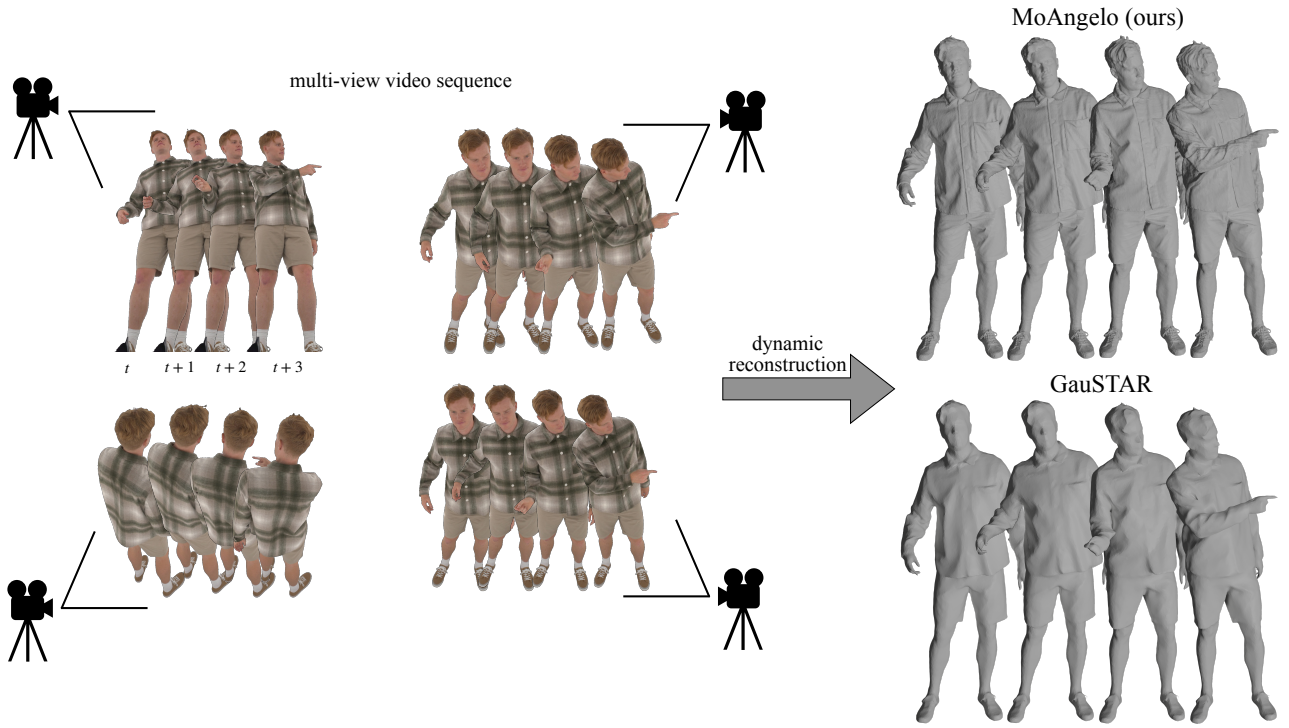

Figure 1. We present MoAngelo, a dynamic multi-view reconstruction method, that can produce more detailed geometry than competitors and avoids smoothing out details. Our method is based on a static template reconstruction using NeuralAngelo [14], however, in contrast previous approaches, our template is flexible and refined during the optimization of the deformation fields which leads to better accuracy.

Abstract

Dynamic scene reconstruction from multi-view videos remains a fundamental challenge in computer vision. While recent neural surface reconstruction methods have achieved remarkable results in static 3D reconstruction, extending these approaches with comparable quality for dynamic scenes introduces significant computational and representational challenges. Existing dynamic methods focus on novel-view synthesis, therefore, their extracted meshes tend to be noisy. Even approaches aiming for geometric fidelity often result in too smooth meshes due to the ill-posedness of the problem. We present a novel framework for highly detailed dynamic reconstruction that extends the static 3D reconstruction method NeuralAngelo to work in dynamic settings. To that end, we start with a high-quality template scene reconstruction from the initial frame using NeuralAngelo, and then jointly optimize deformation fields that track the template and refine it based on the temporal sequence. This flexible template allows updating the geometry to include changes that cannot be modeled with the deformation field, for instance occluded parts or the changes in

the topology. We show superior reconstruction accuracy in comparison to previous state-of-the-art methods on the ActorsHQ dataset.

1. Introduction

Dynamic reconstruction, the task of recovering the dynamic geometry of a scene captured from multi-view videos, remains a challenging problem with many applications in virtual and augmented reality, scene understanding and dynamic asset creation. Recent advances in reconstruction and novel view synthesis have shown impressive results for static scenes [13, 14] but moving objects increase the complexity and computational demands of the setup substantially. The core difficulty in dynamic reconstruction stems from the ill-posedness of the problem: recovering time-varying 3D structures from sparse viewpoints requires priors and regularization to disambiguate between changes caused by motion and viewpoint variations. Early methods relied on pre-defined templates and structured motion models. While these approaches work well on controlled datasets, they are limited by their reliance on explicit surface representations and hand-crafted motion priors, making them struggle with complex deformations and topological changes.

Neural Radiance Fields (NeRF) [2, 21, 22] have revolutionized static 3D scene representation by demonstrating that neural fields can capture view-dependent appearance effects. Despite the impressive novel-view synthesis results, the volume rendering approach neglects the representation of geometry, which makes it difficult to extract 3D meshes. As a solution to this, Neural SDFs [14, 30, 33] represent the geometry as a neural signed distance function (SDF) instead of a volume density, enabling the extraction of high-quality surfaces by applying marching cubes [15]. However, these methods are limited to static scenes and it remains challenging to extend them to dynamic scenes with the same quality of results.

Due to the overall success of neural radiance fields, several methods have extended them for dynamic settings by adding a temporal dimension or a deformation field [12, 24, 26, 29]. However, the same difficulties with extracting meshes from volume densities remain. Current trends in dynamic scene representation rely heavily on deformable 3D Gaussian splatting (3DGS) [32, 35] which are more efficient to optimize but also target mainly novel-view synthesis. Some methods represent the moving geometry as a neural SDF [27] or track 3DGS together with a mesh [36]. In these cases, the extracted geometry lacks fine geometric details and the reconstruction quality is not comparable with the quality of static 3D reconstruction methods. The reasons range from the template being implicit only which averages the geometry through the sequence making it overly smooth [27], to limited mesh resolution for optimization reasons [36].In summary, dynamic scene representation methods are achieving impressive novel-view synthesis results but lack the capability to extract high fidelity geometry.

In this work, we propose a novel framework for high-fidelity dynamic neural surface reconstruction. Our idea is based on taking NeuralAngelo [14], the current state-of-the-art in static reconstruction based on neural SDFs with highly detailed reconstructions, on the first frame, and then extending it to iteratively optimize neural deformation fields that track the movement of the template through time while jointly refining its geometry. Our project page is available at: https://mohamed-ebbed.github.io/moangelo

Contributions. Our main contributions are as follows:

- A novel framework for dynamic reconstruction that jointly deform the reconstructed template of an initial frame while tracking the movements of it and refining its geometry at the same time.
- Highly detailed dynamic reconstructions from multi-view videos that preserve surface details even in longer sequences with large movement.
- Our experimental results show that we outperform previous methods both qualitatively and quantitatively by a large margin.

2. Related Work

We focus this section on directly related work using RGB data but a more general survey including depth information can be found in [37].

2.1. Static 3D Reconstruction

Classical multi-view reconstruction algorithms used to rely on feature extraction and matching [8, 18] or photometric stereo by using lighting variations to recover surface normal information [1, 11]. These methods typically produce explicit surface representations such as point clouds or meshes, which provide a direct geometric interpretation but struggle with fine details. Recently, 3D Gaussian splatting [13] has been established as a powerful explicit representation that combines the benefit of point-based methods with differentiable rendering. Even though this representation was developed for novel view synthesis, countless extensions have been proposed to improve geometric fidelity and the quality of surface extractions [7, 10, 35].

Neural Field-based Methods A completely different approach is taken by neural field-based methods which implicitly define the scene properties like geometry or radiance by functions encoded in a neural network. This di-

rection became increasingly popular after the introduction of Neural Radiance Fields (NeRF) [21] for novel view synthesis, but implicit representations have been widely used for reconstruction before due to the flexibility in optimization [5, 20]. The quickly growing field addressed existing challenges with implicit representations through various architectural and algorithmic improvements: Instant-NGP [22] drastically accelerated training and inference through multi-resolution hashgrids. In the direction of pure geometry representation, NeuS [30] replace the NeRF density-based formulation with a pure signed distance function (SDF) to enable straightforward surface extraction, and [33] closed the gap between volume and surface rendering by learning the relationship between a density and signed distance function. The highest level of geometric detail has been achieved by combining hashgrids with SDFs, which was used in NeuS2 [31], MonoSDF [34], and NeuralAngelo [14]. Our dynamic reconstruction is based on extending Neuralangelo to work for dynamic scenes by initially reconstructing a high-quality template mesh, then jointly track its movement using deformation fields and furtherly tuning the geometry of the template mesh.

2.2. Dynamic 3D Reconstruction

Dynamic reconstruction adds an additional time dimension to the problem, which increases the computational complexity significantly. Early methods relied on pre-defined templates and structured motion models. For example, non-rigid structure-from-motion extended classical bundle adjustment to handle deformable objects by assuming low-rank deformation [4]. Multi-view stereo techniques, on the other hand, were able to incorporate temporal consistency through optical flow and surface tracking [8, 23].

The success of neural fields in static reconstruction naturally led to extensions in the dynamic setting. Nerfies [24] and D-NeRF [26] decomposed the problem into optimizing for a canonical NeRF that represents the canonical scene properties, and a deformation field representing the non-rigid deformations of the canonical scene. We follow a similar idea, instead of implicitly optimizing for the canonical scene, we explicitly define it as the initial frame of the scene and initially reconstruct it and furtherly tune its geometry through time. We also iteratively optimize for a deformation field that tracks the movement of that template mesh. SceNeRFlow [29] also defines the template scene as the first frame and iteratively optimizes for a deformation field to track it, however their approach is mainly focused on novel-view synthesis because the geometry is represented as a volume density, therefore it is not straightforward to extract dynamic meshes from their representation. On the other hand, our geometry is represented as a neural SDF, making it easy to extract high-resolutional meshes using marching cubes. Additionally, the current trend is to use dynamic 3D gaussians that deform through time [32], however these approaches suffer from the same problem that they are only focused on novel-view synthesis, and suffer from the same problems that face gaussian splatting methods [35] when extracting a 3D mesh from them, thus all the current methods are limited only to novel-view synthesis. We refer the reader to [28] for an in-depth survey of recent non-rigid reconstruction methods.

Most related to our work, and thus used as competitors in the experiments section, are the following works: HumanRF [12] is able to produce high resolution 4D reconstructions by generating a 4D volume density, but the volume density leads to noisy mesh extractions. The method of Tensor4D [27] implicitly optimizes for a template scene and a deformation field represented by 4D feature grid followed by a shallow MLP. The feature grid is decomposed into low-rank tensors using tensor decomposition methods [6]. The template scene is implicitly optimized for, therefore the resulted template scene is very smooth, leading to reconstructed meshes that lack the fine geometric details. Not limited to neural fields, GauSTAR [36] combines dynamic 3D Gaussian splatting with a template reconstructed from HumanRF and deformed based on the movement of the Gaussians, however the resulted meshes are not high-resolutional, so they doesn't capture the fine geometric details.

3. Problem Setup and Notation

This section will introduce the problem setup, representations, and notation (Tab. 1) we used.

Symbol	Description
t	time step variable, frame index
f_{sdf}	signed distance function of geometry, also called template and canonical frame
f_{rgb}	the radiance field of the template frame
$x_{i,t}$	3D coordinate of point i in frame t indices are dropped when irrelevant
$\hat{x}_{i,t}$	$x_{i,t}$ deformed into canonical frame
s_i, d_i	SDF value and geometric features of x_i
f^t_{deform}	deformation field for frame t

Table 1. Overview of most important notation.

3.1. Objective

Given a sequence of multi-view RGB images of a scene captured from N_C cameras, each consisting of N_F frames, our objective is to reconstruct N_F high-quality triangular meshes, where each mesh M_i represents the detailed geometry of the scene at time step i. Optionally, segmentation masks may be provided for each frame to focus only on the object of interest.

3.2. Geometry Representation

The scene geometry is represented as signed distance function (SDF) $f_{\text{sdf}} : \mathbb{R}^3 \rightarrow \mathbb{R}$ in a multi-resolution hash grid [22] followed by a multi-layer perceptron (MLP) with a single hidden layer of 64 neurons. The SDF returns the distance to the closest surface, with the sign indicating whether the point lies inside or outside the surface. A triangular mesh can be extracted from the SDF using the marching cubes algorithm [16]. The hash grid comprises 16 resolution levels, ranging from 32^3 to 2024^3. Each level outputs a 4-dimensional feature vector, which are concatenated to form a total feature vector of dimension 64. This feature vector is then concatenated with the input 3D position x, and the resulting vector is passed to the MLP to predict both the SDF value s at that point and an additional feature vector d. The surface normal at x is computed as the gradient of the SDF with respect to the input position, i.e., $n = \nabla_x f_{\text{sdf}}(x)$. This gradient can be efficiently calculated using PyTorch's automatic differentiation [25]. At the beginning, f_{sdf} is initialized by running NeuralAngelo [14] on the first frame, then the template is tracked and gradually refined during our optimization.

3.3. Appearance Representation

The appearance of the scene is represented using a radiance field $f_{\text{rgb}}(x, v, n, d)$, which maps the 3D position x, viewing direction v, surface normal n, and the SDF feature vector d to a color value c.

4. Methodology

We propose a novel framework for dynamic reconstruction based on explicit modeling of the template reconstructed from the canonical frame, and joint optimization of motion and template for the rest of the sequence. Our method starts with a static reconstruction of the first (or canonical) frame using NeuralAngelo [14] (see Sec. 4.3). In contrast to previous work, we use this reconstruction as an explicit representation that is regularized and refined during the optimization of the motion (see Sec. 4.1) morphing it into later frames, see Sec. 4.2. This combination results in a high fidelity reconstruction of fine details.

4.1. Motion Representation

We represent the scene motion as a collection of deformation fields, each mapping points from the coordinates of the current time step t (observation frame) into the canonical frame. The geometry at time t can then be computed as

$$f^t_{\text{sdf}}(x) = f_{\text{sdf}}(f^t_{\text{deform}}(x)), \tag{1}$$

where $x \in \mathbb{R}^3$, $f_{\text{deform}} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, and $f_{\text{sdf}} : \mathbb{R}^3 \rightarrow \mathbb{R}$ is the signed distance representation of the template. Each f^t_{deform} consists of a multi-resolution hash grid with the same number of levels, feature dimensions, and resolutions as the hash grid as f_{sdf}. The hash grid is followed by single-layer MLP with 64 neurons. (see Sec. 3.2). Specifically, the deformation fields predict an $\mathfrak{so}(3)$ transformation from the observation frame to canonical frame, then the transformation is converted to a rotation matrix R_i and a translation vector T_i using exponential mapping [19]. We apply the transformation to $x_{i,t}$ as follows:

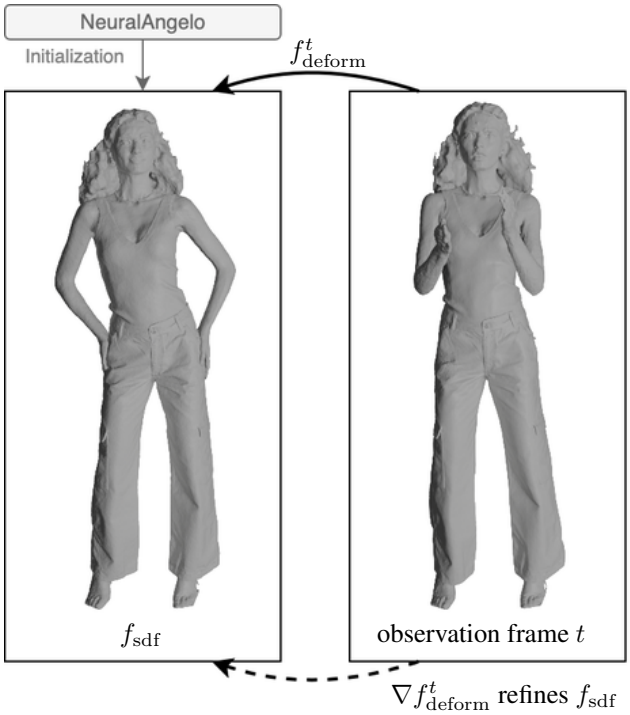

Figure 2. Overview of the joint optimization of the template f_{sdf} and the deformation to each observation frame f^t_{deform}. The movement for frame t is stored in a deformation field f^t_{deform} and optimized based on the current state of the template f_{sdf}. However, the gradient information of this update is propagated to f_{sdf} which is refined in turn.

$$R_i, T_i = f^t_{deform}(x_{i,t}) \tag{2}$$

$$\hat{x_i} = R_i x_{i,t} + T_i \tag{3}$$

In contrast to previous work [26, 27], we model and deform the template f_{sdf} explicitly, and can optimize its geometry and appearance to high fidelity and without over-smoothing. However, this also requires explicit tracking of the template in all frames which we describe in the next section.

4.2. Template Mesh Tracking

Each deformation field f^t_{deform} needs to map the geometry of the corresponding frame t exactly to the right position of the template geometry f_{sdf}, which will be optimized

through volume rendering.

Volume Rendering. For each frame t, we need to render the 3D neural fields into 2D to compare them to the input images. To that end, we shoot N_r rays at random pixels and 80 many 3D points x_i along these rays. The signed distance value s_i and feature vector d_i for x_i can be computed in the canonical frame by taking $s_i, d_i = f_{sdf}(\hat{x}_i)$ and the normal vector as $n_i = \nabla_x f_{\text{sdf}}(\hat{x}_i)$. R_i is applied to the normals to move them from canonical space back to observation frame. Finally, the color c_i at x_i is computed as $c_i = f_{rgb}(\hat{x}_i, v_i, d_i, n_i)$ where v_i is the viewing direction of the ray.

To render the correct RGB value C_r for the pixel, we need to integrate all points on the ray shooting through it.

$$C_r = \sum_{i=1}^{N_s} T_i \alpha_i \cdot c_i \tag{4}$$

where α_i is the opacity of x_i and $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$ the transmittance which represents the probability a ray will reach the sample without being occluded, the standard way to compute the weighting [14, 30].

$$\alpha_i = \max\left(\frac{\Phi_s(s_i) - \Phi_s(s_{i+1})}{\Phi_s(s_i)},\ 0\right),$$

Where Φ_s is the derivative of the sigmoid function, known as the logistic density distribution.

4.2.1 Optimization

The deformation field f^t_{deform} for each frame t is optimized separately but initialized with f^{t-1}_{deform}, or identity for $t = 0$. The total loss function is

$$\mathcal{L}_{total} = L_{render} + \lambda_{mask}\mathcal{L}_{mask} + \lambda_{eik}\mathcal{L}_{eik}, \tag{5}$$

definitions of each term follow below. The loss function is very similar to a static single frame reconstruction using NeuralAngelo [14] (see Sec. 4.3), In each observation frame t, we optimize for the current f^t_{deform}. Not all geometric details were necessarily visible during initialization, due to (self-)occlusions or topology changes, therefore we refine the template scene to handle these changes by allowing the gradient signal back to f_{sdf} and f_{rgb}. The joint optimization of the template and each deformation field enables a refinement of the geometry and appearance which leads to much more accurate and temporal consistent reconstructions than a static template or implicit template frame can.

While reconstructing the template, we run the optimization for $250k$ steps with AdamW optimizer [17] which take 2 hours, then for the following time steps we run it for $25k$ steps for each time step t which take 30 minutes for each time step. The number of iterations is treated as a hyperparameter to balance quality and runtime. We also optimize the hash grid of each deformation field in a coarse-to-fine manner to prevent overfitting. Initially, we start the optimization with 4 active levels, and every 1000 steps we activate an additional level until all 16 levels are active.

Rendering Loss. To optimize the template scene, we sample a minibatch of N_r pixels from a random image with ground truth RGB value of $\hat{C}_r$ and compute the L1 re-rendering loss between the rendered RGB colors of the shooted rays to these pixels and their ground truth values as follows:

$$\mathcal{L}_{\text{render}} = \frac{1}{N_r}\sum_{r=1}^{N_r}\left\|C_r - \hat{C}_r\right\|_1 \tag{6}$$

Masking Loss. Furthermore, to encourage the template scene represent only the object of interest, we use 2D segmentation masks to guide the optimization to the region where the object of interest lies. Specifically, we sum the weights along the ray and apply a binary cross-entropy loss between this sum and the corresponding mask value. In our experiments, we set λ_{mask} as 0.1 when optimizing for the template scene, and increase it to 1.0 when tracking the template.

$$\mathcal{L}_{\text{mask}} = \frac{1}{N_r}\sum_{r=1}^{N_r} BCE(W_r, M_r) \tag{7}$$

Where W_r is the weights sum of samples on ray r and M_r is the mask value.

Eikonal Loss. Additionally, to encourage the SDF to converge to a plausible SDF, we apply the eikonal regularization [9] on the normal vectors by penalizing the deviation of their L2-norm from one as follows:

$$\mathcal{L}_{\text{eik}} = \frac{1}{N_r N_s}\sum_{r=1}^{N_r}\sum_{i=1}^{N_s}\left(\|\nabla_x f_{\text{sdf}}(x_{r,i})\|_2 - 1\right)^2 \tag{8}$$

λ_{eik} is set to be 0.1 while optimizing for the template and while tracking it.

4.3. Template Reconstruction

The optimization in Sec. 4.2.1 requires an initialization for the explicit template f_{sdf} and appearance f_{rgb}. We use the state-of-the-art static reconstruction method NeuralAngelo [14], which achieves state-of-the-art performance among Neural SDF-based methods, on the multi-view images of the first frame to get this initialization. While the

reconstruction of NeuralAngelo is very accurate, it cannot include information from later frames which might lead to small inaccuracies in the template due to occlusions or ambiguities, but also topological inconsistencies due to larger movement in later frames. Our refinement during optimization, see Sec. 4.2.1, allows us to increase the fidelity, remove mistakes and adapt the topology of the template without relying on a perfect reconstruction from NeuralAngelo. With this, we can achieve a high-quality reconstruction for each frame in only $25k$ steps, which amounts to 10% of the optimization time for the template scene.

5. Evaluation

In the following section, we present the experimental evaluation of our proposed method against several baselines on dynamic reconstruction based on a quantitative evaluation (Sec. 5.2), qualitative results (Sec. 5.3), and an ablation study (Sec. 5.4). Additional qualitative examples can be found in the supplementary material. Experiments were done on a single Nvidia A100 GPU with 80GBs memory.

5.1. Experimental Setup

5.1.1 Dataset

To evaluate our method and compare it with the other baselines, we use the ActorsHQ dataset [12]. It consists of multi-view videos of moving humans captured from 160 high-resolutional cameras with a total of 39,765 frames. We experiment on 6 scenes from the dataset with sequences where the actor is moving ranging from 30 to 40 time steps due to computational demands. We choose ActorsHQ because it the only multi-view video dataset that contains ground-truth meshes for all frames which we use to quantify and compare the reconstruction accuracy.

5.1.2 Baselines

To show that our method achieves the best geometric quality for dynamic scenes, we compare with different recent dynamic approaches:

Tensor4D [27] represents the geometry of the dynamic scene using a neural SDF defined by a 3D feature grid followed by a shallow MLP. They implicitly optimize for the template scene by having a deformation field aligning the positions of the observation frame to the coordinates of the template scene. The 3D and 4D hashgrids are decomposed into low-level tensors using tensor decomposition methods [6].

HumanRF [12] uses a neural field representation that maps the 4D input (x, y, z, t) to a volume density and an RGB value. The neural field consists of a 4D grid followed by a shallow MLP. The 4D grid is decomposed into lower rank multi-resolution hash grids using tensor decomposition. Since the geometry is encoded in a volume density, the extracted 3D meshes are low quality because the exact surface intersection is unknown.

GauSTAR [36] combines 3D Gaussians splats with a 3D mesh. The initial template mesh is initialized from HumanRF[12], and while the 3D Gaussians are tracked through time, the template mesh is deformed accordingly.

Scene	Tensor4D [27]	GauSTAR [36]	HumanRF [12]	Ours
Actor1	0.0150	0.00833	0.0091	**0.0057**
Actor5	0.0187	0.2200	0.1100	**0.0037**
Actor6_1	0.0160	0.0077	0.0087	**0.0068**
Actor6_2	0.1830	0.0078	0.0088	**0.0058**
Actor7	0.0110	0.0067	0.0046	**0.0035**
Actor8	0.3590	0.0068	0.0119	**0.0059**
Average	0.1005	0.04288	0.0255	**0.0052**

Table 2. **Reconstruction Accuracy.** We report L_1-Chamfer distance (lower is better) for all methods. Our method provides the most accurate reconstruction on all scenes.

5.2. Quantitative Evaluation

To evaluate the quality of our reconstructions quantitatively and to compare them with the baselines, we compute the Chamfer distance [3] between the reconstructed meshes and the ground truth meshes by sampling a point cloud of 1M points from each surface and computing the L_1 Chamfer distance between them. Results are reported in Tab. 2 and we outperform all competitors on all scenes. To extract meshes from HumanRF, we use the postprocessing proposed in GauSTAR [36]. Specifically, we apply marching cubes on the density field and use the provided 3D occupancy grids computed from the segmentation masks in the dataset to zero the non-occupied regions in the volume density grid. For this reason, it is not be possible to extract a mesh from HumanRF if segmentation masks are not available. Our method does not suffer this limitation since the zero level set of an SDF is always well defined.

5.3. Qualitative Evaluation

Fig. 3 presents a direct qualitative comparison which shows that our method reconstructs much finer details than competitors and even provides accurate results in complex cases. A video of the reconstruction of all sequences in comparison to the baselines on ActorsHQ can be found on the project page. Additional qualitative figures can be found in the supplementary.

5.4. Ablation Study

In this section, we evaluate the different design choices of our framework, we run a quantitative comparison between the different variants and our full version using the same experimental settings mentioned before. We show in Tab. 3, that our full version achieves the best average L_1-Chamfer

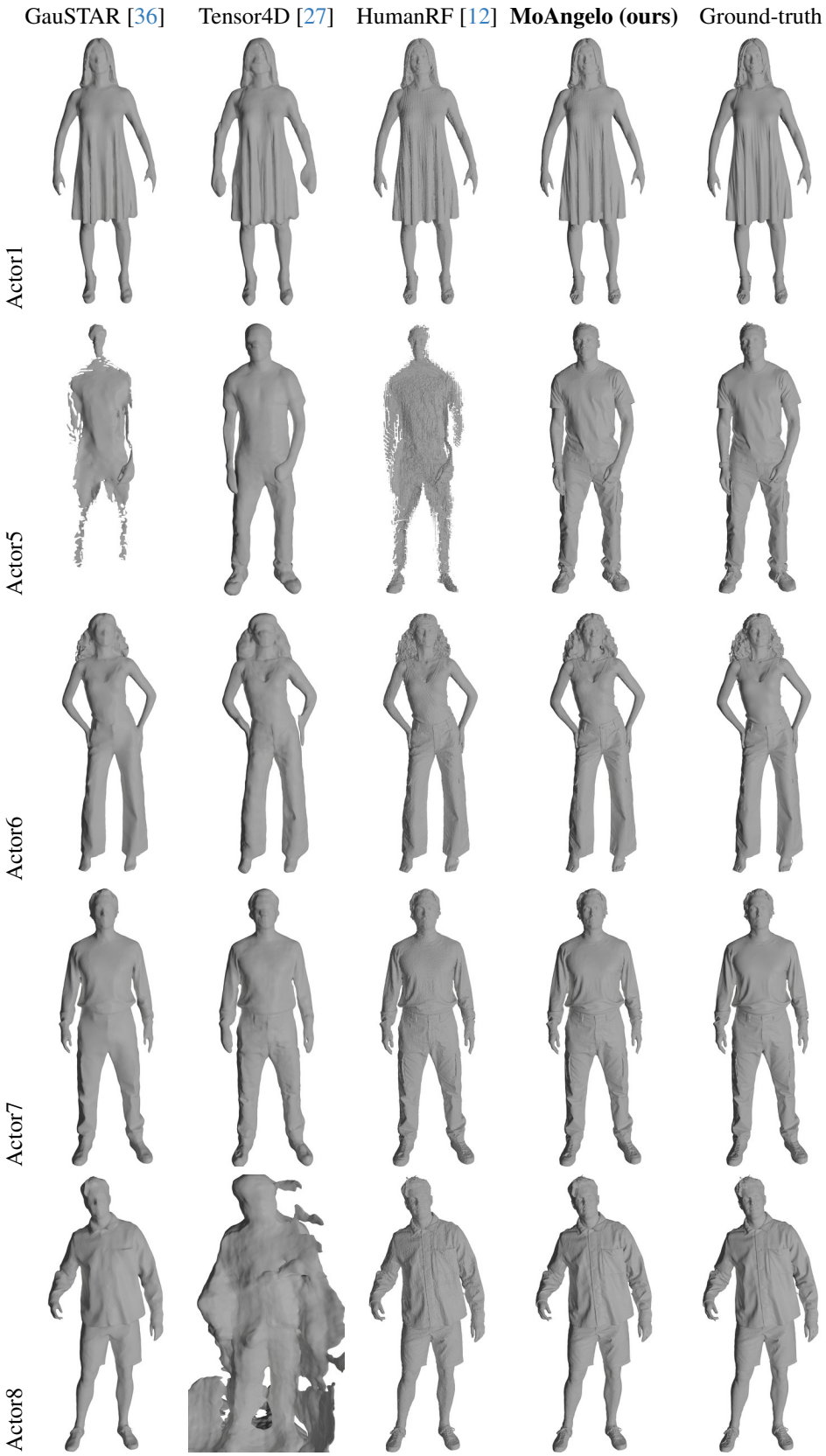

Figure 3. Qualitative results of competitors and MoAngelo (our method) from four different scenes in the ActorsHQ dataset. Our method reconstructs the highest fidelity geometry without unnecessary noise and does not have major failure cases.

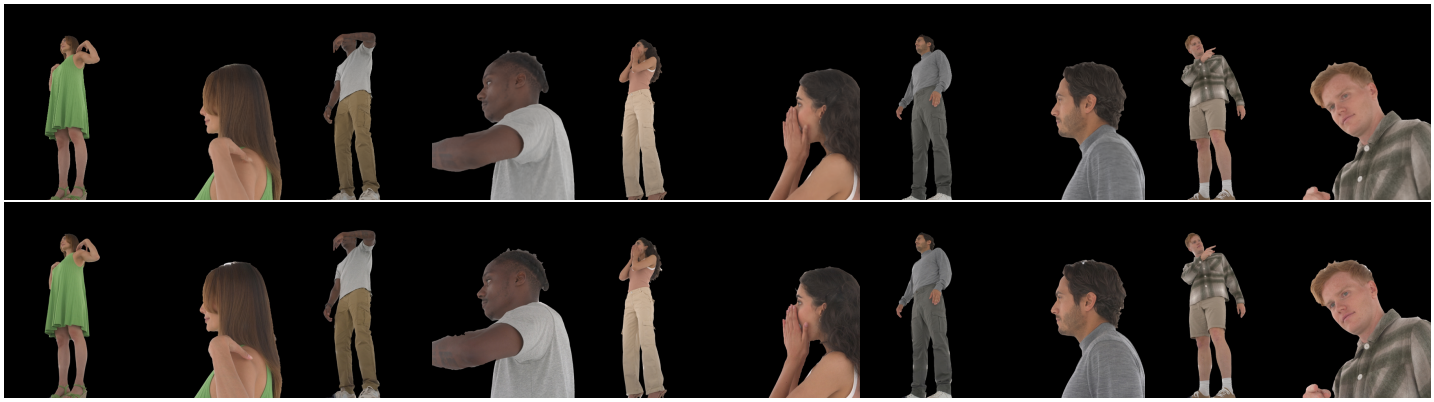

Figure 4. Visualization of renderings for novel-view synthesis. **Top:** Our renderings. **Bottom:** Ground truth, We demonstrate that our approach is capable not only of producing highly accurate meshes but also of generating high-quality renderings for novel-view synthesis

w/o refinement | w/o deform init | w/o C2F deform | **Ours (full)**

Figure 5. Ablation study on the effect of different design choices. Our full model achieves the best reconstruction quality.

Scene	w/o refine.	w/o deform init.	w/o C2F deform	**Ours (full)**
Actor1	0.0973	0.0057	0.0074	**0.0057**
Actor5	0.0359	0.0042	**0.0037**	**0.0037**
Actor6_1	0.0088	0.0073	0.0070	**0.0068**
Actor6_2	0.0062	0.0068	0.0062	**0.0058**
Actor7	0.0043	0.0039	**0.0035**	**0.0035**
Actor8	0.0089	0.0071	0.0080	**0.0059**
Average	0.0269	0.0058	0.0059	**0.0052**

Table 3. **Ablation Study.** We report L_1-Chamfer distance (lower is better) for all the variants of our framework, our full version achieves the best chamfer distance on most scenes and on average distance on the scenes.

5.4.1 Refining the Template Scene

In our framework, gradients are propagated back to the template scene, allowing it to adapt to topology changes and occlusions that occur after the first frame. When this gradient flow is disabled and only the deformation field is optimized, the quality of the reconstruction is significantly degraded as shown in Fig. 5. This degradation happens because the deformation field alone cannot capture topology changes or occlusions, as it can only model non-rigid transformations from the current frame to the template's coordinate space.

5.4.2 Initializing the Deformation Field

In our experiments, we show that it is necessary to initialize the deformation field f^t_{deform} with the weights of the previous deformation field f^{t-1}_{deform}, the experiments show that initializing the deformation field every time step from scratch makes it challenging for the deformation field to represent the correct deformation, the higher t, the larger is the transformation back to the template scene, leading to a noisy reconstruction as demonstrated in Fig. 5.

5.4.3 Coarse-To-Fine Optimization of the Deformation Field

In our setup, we optimize the deformation field in a coarse-to-fine manner, we initially start with 4 active hashgrid levels, and gradually activate a new level every 1000 steps. Initiating the optimization with the complete 16 hashgrid levels, lead to overfitting and the inability to accurately represent the deformation as shown in Fig. 5.

6. Conclusion

We presented MoAngelo, a multi-view dynamic reconstruction method, based on deforming a template from a static reconstruction method to later frames while updating and refining the template during optimization. This joint update of geometry and deformation prevents oversmoothing, can update previously noisy or occluded parts and is flexible enough to incorporate later topology changes. In experiments, our method outperforms the previous state-of-the-art in reconstruction accuracy and provides significantly more visually pleasing results.

References

[1] Jens Ackermann and Michael Goesele. A survey of photometric stereo techniques. *Foundations and Trends® in Computer Graphics and Vision*, 9(3-4):149–254, 2015. 2

[2] 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, 2021. 2

[3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, page 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc. 6

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 690–696 vol.2, 2000. 3

[5] Fatih Calakli and Gabriel Taubin. Ssd: Smooth signed distance surface reconstruction. In *Computer Graphics Forum*, pages 1993–2002. Wiley Online Library, 2011. 3

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields, 2022. 3, 6

[7] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 2

[8] Yasutaka Furukawa and Carlos Hernández. 2015. 2, 3

[9] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes, 2020. 5

[10] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 2

[11] Bjoern Haefner, Zhenzhang Ye, Maolin Gao, Tao Wu, Yvain Queau, and Daniel Cremers. Variational uncalibrated photometric stereo under general lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[12] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2, 3, 6, 7

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

[14] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 4, 5

[15] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 2

[16] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987. 4

[17] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 5

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

[19] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017. 4

[20] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 3

[21] 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. *CoRR*, abs/2003.08934, 2020. 2, 3

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2, 3, 4

[23] Armin Mustafa, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton. Temporally coherent 4d reconstruction of complex dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[24] 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 (ICCV)*, pages 5865–5874, 2021. 2, 3

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 4

[26] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes, 2020. 2, 3, 4

[27] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2, 3, 4, 6, 7

[28] Edith Tretschk, Navami Kairanda, Mallikarjun B R, Rishabh Dabral, Adam Kortylewski, Bernhard Egger, Marc Habermann, Pascal Fua, Christian Theobalt, and Vladislav

Golyanik. State of the art in dense monocular non-rigid 3d reconstruction. *Computer Graphics Forum*, 42(2):485–520, 2023. 3

[29] Edith Tretschk, Vladislav Golyanik, Michael Zollhöfer, Aljaz Bozic, Christoph Lassner, and Christian Theobalt. Scenerflow: Time-consistent reconstruction of general dynamic scenes. In *International Conference on 3D Vision (3DV)*, 2024. 2, 3

[30] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, 2023. 2, 3, 5

[31] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[32] 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 (CVPR)*, pages 20310–20320, 2024. 2, 3

[33] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2, 3

[34] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[35] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. 2, 3

[36] Chengwei Zheng, Lixin Xue, Juan Zarate, and Jie Song. Gaustar: Gaussian surface tracking and reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 16543–16553, 2025. 2, 3, 6, 7

[37] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. *Computer Graphics Forum*, 37(2):625–652, 2018. 2