Generating View-Changing Human Action Videos Using a Single Reference Image and Textual Prompting

by

Jyothi Sravan Kumar Akula

August 19, 2024

A thesis submitted to the

Faculty of the Graduate School of

the University at Buffalo, The State University of New York

in partial fulfilment of the requirements for the

degree of

Master of Science

Department of Computer Science and Engineering

Acknowledgments

I wish to express my sincere gratitude to Dr. Junsong Yuan for his invaluable guidance and support over the past one and a half years. I also thank Dr. Sreyasee Das Bhattacharjee for her feedback during my thesis defense. I would like to thank Sudhir and Liangchen for their inputs on this work. Finally, I am grateful to my family for their unwavering support and confidence in me, which empowers me to move forward.

Abstract

Human motion synthesis involves generating videos of a human performing specified actions from a reference image through pose inputs. Applications such as fitness and dance training apps, e-commerce virtual try-ons, and social media content creation would greatly benefit from human motion video synthesis that supports view changes and accepts motion directives through textual inputs. Current models are typically restricted to generating videos with a fixed camera view and require precise pose inputs to be given by users, limiting their applicability in various 3D application scenarios.

Traditionally, producing human action videos with view changes requires generating clothed 3D mesh models and rigging them to pre-scripted actions. While this process allows for full 3D view changes, it is complex, time-consuming, and not user-friendly. Moreover, many use cases do not require fully-fledged view changes; minor viewpoint adjustments from the front view often suffice to create the illusion of dynamic 3D videos.

This thesis presents a novel pipeline for generating human action videos from text with minor viewpoint changes. By leveraging a single reference image and textual pose instructions, our approach bypasses the need for manual 3D animation. Instead, we utilize a combination of off-the-shelf generation models to synthesize human action videos from given text instructions and employ depth-based view synthesis techniques to create dynamic, view-changing videos of moving humans with minimal production time and effort.

Our method offers a more efficient, cost-effective, and accessible alternative to traditional methods. The simplicity of our pipeline also facilitates further video editing and manipulation, including background modifications. By democratizing the creation of high-quality, dynamic video content, our solution bridges the gap between complex 3D simulations and practical content creation, making it accessible to a broader audience.

Table of Contents

List of Tables

List of Figures

Chapter1

Introduction

1.1 What is Human Motion Video Synthesis?

Human Motion Video Synthesis involves generating videos of a human performing specified actions from a reference image, offering a unique approach to content creation. This process relies heavily on pose inputs to generate the motion video. The pose sequences are extracted from reference videos of other humans using pose detectors or from motion capture systems. This heavy reliance on pose inputs gives control over the generation of long and complex motion but has severe limitations.

For example, the reliance on pose inputs restricts the use of more intuitive methods, such as text-based instructions for video generation. Additionally, these models typically lack camera control, operating from a fixed viewpoint due to being trained on datasets that do not account for changes in perspective or camera angles. These limitations significantly diminish the dynamic quality of the generated videos.

Previous works, such as those by Kim et al.[1] and Jiang et al. [2], have explored the use of text-based instructions for action generation. However, these methods do not incorporate reference-view images, relying instead on text alone to generate the entire video. Other models, like VideoComposer [3], allow for multimodal inputs, combining both reference images and text to generate human action videos. However, these models cannot generate long and intricate actions and lack full camera control.

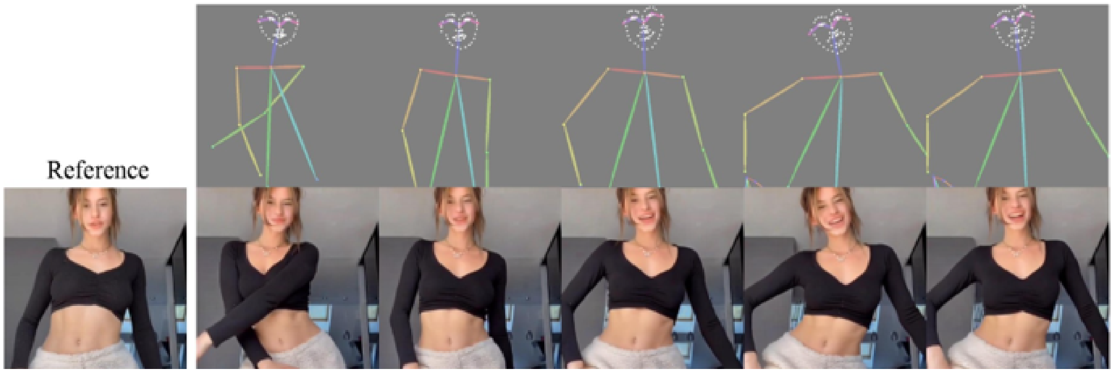

Figure 1.1: An example of Human Motion Video Synthesis.

Addressing these limitations by incorporating textual prompts and enabling camera or viewpoint changes within synthesized videos could unlock numerous applications. For instance, in fitness and dance training apps, users could receive personalized, dynamic demonstrations of exercises and dance moves from different angles, enhancing their learning experience. In e-commerce, virtual try-ons could be significantly improved by allowing customers to view themselves performing various actions in different outfits, providing a more comprehensive understanding of the product.

1.2 Traditional Methodology

Traditional methodology for generating human action videos with view changes is a highly complicated process. First, a fully clothed, textured 3D model resembling the human in the reference video is created. This model is then rigged to a skeleton, allowing the user to manipulate the 3D model using pose inputs. The desired actions are scripted into these pose inputs, which guide the model's movements frame by frame. Since the final product includes a complete 3D model, it enables full 3D camera view changes while the dynamic motion is performed.

However, this method has significant drawbacks. Generating 3D mesh models, rigging them, and scripting actions is complex and time-consuming, requiring substantial technical

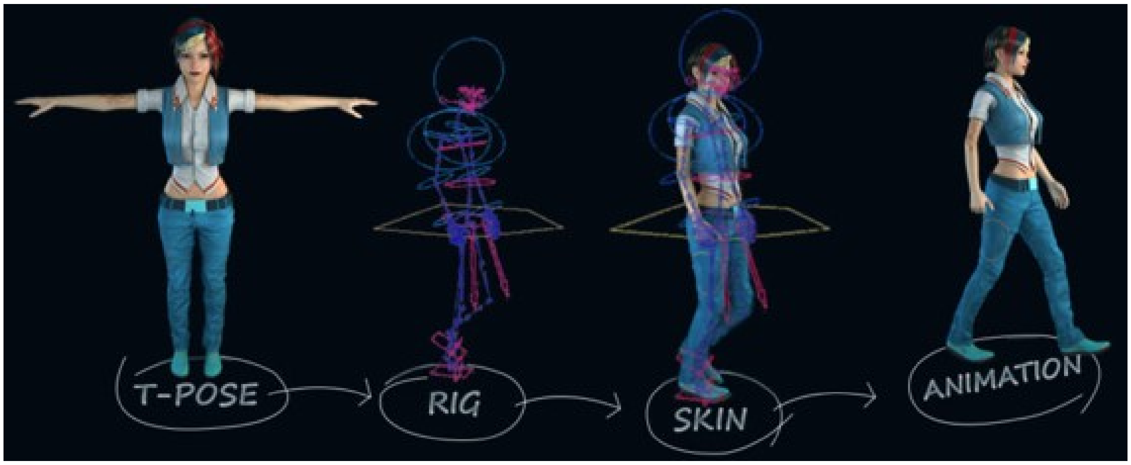

Figure 1.2: The process of Textured Mesh Rigging. [4].

expertise. Each step demands precise attention to detail, from accurately modeling clothing and body shapes to correctly implementing movement dynamics. Additionally, the need for specialized software and expertise in 3D modeling makes this approach less accessible to casual users, limiting its usability for broader applications.

1.3 Our Pipeline

To address the limitations of traditional methodologies, we propose an alternative pipeline designed to simplify the process of generating human action videos while still achieving a compelling illusion of full 3D. For many applications, full-fledged 3D view changes are unnecessary; minor adjustments to the viewpoint, especially from a front-facing perspective, can be sufficient to create the impression of a dynamic video. This insight forms the basis of our novel pipeline, which leverages text-based instructions and a single reference image to produce human action videos with slight viewpoint changes.

Our pipeline is divided into three key sections, each playing a specific role in the video generation process. The first section focuses on synthesizing pose sequences directly from text instructions. By translating textual descriptions into a sequence of poses, this section eliminates the need for pre-scripted inputs, offering a more intuitive and flexible approach to guiding human motion.

The second section is responsible for generating forward-facing human action videos. Using the synthesized pose sequences and the reference image, this section creates videos that capture the specified human motion sequence from a front-view camera.

Finally, the third section converts the forward-facing videos into view-changing videos. By applying subtle adjustments to the camera angle or perspective, this section creates the impression of a more dynamic 3D experience. While these viewpoint changes are minor, they are significant enough to provide a sense of depth and movement, making the videos more immersive without the need for complex 3D modeling.

Through this streamlined approach, our proposed pipeline offers a practical and efficient alternative to traditional methods, making the creation of dynamic human action videos more accessible and less time-consuming while still delivering high-quality results suitable for a wide range of applications.

Chapter2

Related Work

2.1 3D Avatar Generation Models

3D avatar generation models build on advanced 3D reconstruction techniques to create fully textured human assets from text or image inputs. These frameworks are capable of generating detailed full-body avatars in a short amount of time, but they typically require mesh rigging and pose inputs to animate the avatars. Moreover, they often lack the ability to model the background or other elements of the scene. For example, Kolotouros et al. [5] generates front and back views of an avatar, which are then combined into a textured mesh using pixel-aligned 3D reconstruction. Huang et al. [6] take a different approach by progressively optimizing a 3D model and its textures with texture guidance extracted through visual query models. Chen et al. [7] starts by estimating the mesh structure and then build the textures using IPAdapter, showcasing another method of integrating texture generation into the 3D reconstruction process.

2.2 Pose parameterized Volumetric Avatar Models

Pose-parameterized volumetric avatar models offer an alternative approach by eliminating the need for explicit mesh rigging. These models internally parameterize the avatar based on poses, making them more adaptable for dynamic animations. However, they typically require monocular video input for training. Jiang et al. [8] introduce a method that divides the full scene into a scene NeRF (Neural Radiance Field) model and a human NeRF model, which are jointly trained on a video to reconstruct both the scene and the avatar. Kocabas et al. [9] propose a Gaussian-based model that uses MLP networks to parameterize Gaussian splats in canonical space, simplifying the avatar generation process. Similarly, Hu et al. [10] improve upon this by employing an encoder model with an optimizable tensor to better capture pose-based variations in the avatar.

2.3 2.5 D Generation

2.5D generation models provide another layer of depth to image-based representations, offering potential for more realistic and dynamic visuals. These models typically operate on image data, making them simpler yet effective for certain applications. Shih et al. [11] introduce a method that uses Layered Depth Images (LDIs) instead of traditional meshes, leveraging their simplicity to predict edge and mesh inpainting for occlusion handling in nearby views. However, this approach is limited to still images and is not suitable for videos. Tucker et al. [12] take a different approach by using Multi-Plane Images (MPIs) to achieve similar 2.5D effects, offering a more robust solution for single-view view synthesis.

2.4 Image to 3D diffusion models

Image-to-3D diffusion models represent a cutting-edge approach to conditional generation tasks, where 3D assets are created from a single reference image using diffusion models. These models have even evolved to convert video sequences into 4D assets, but they often struggle with human generation due to their training on object-centric datasets and lack of input control. The advancements in this area are built upon foundational work such as Liu et al. [13], which pioneered zero-shot image-to-3D generation. Subsequent models, including Lin et al. [14], Sargent et al. [15], and Liu et al. [16], have focused on improving consistency in 3D generation. Ren et al. [17] push the boundaries further by extending 3D mesh generation to 4D for short video sequences, building upon the work of Tang et al. [18], who introduced the concept of score distillation sampling for Gaussian splatting for 3D models.

Chapter3

Methodology

3.1 Overview

Our proposed pipeline is structured into three distinct sections, each designed to leverage existing tools and models to streamline the process of generating human action videos from text instructions and a reference image. As depicted in Figure 2.1, the entire pipeline is built upon off-the-shelf generators and estimators, ensuring that the implementation is both practical and efficient.

The first section of the pipeline focuses on generating pose sequences directly from textual descriptions. This is followed by the second section, which uses these sequences to create forward-facing human action videos. Finally, the third section applies minor viewpoint changes to enhance the dynamic quality of the generated videos. Each stage plays a crucial role in ensuring that the final output is both visually appealing and functionally effective for a variety of applications.

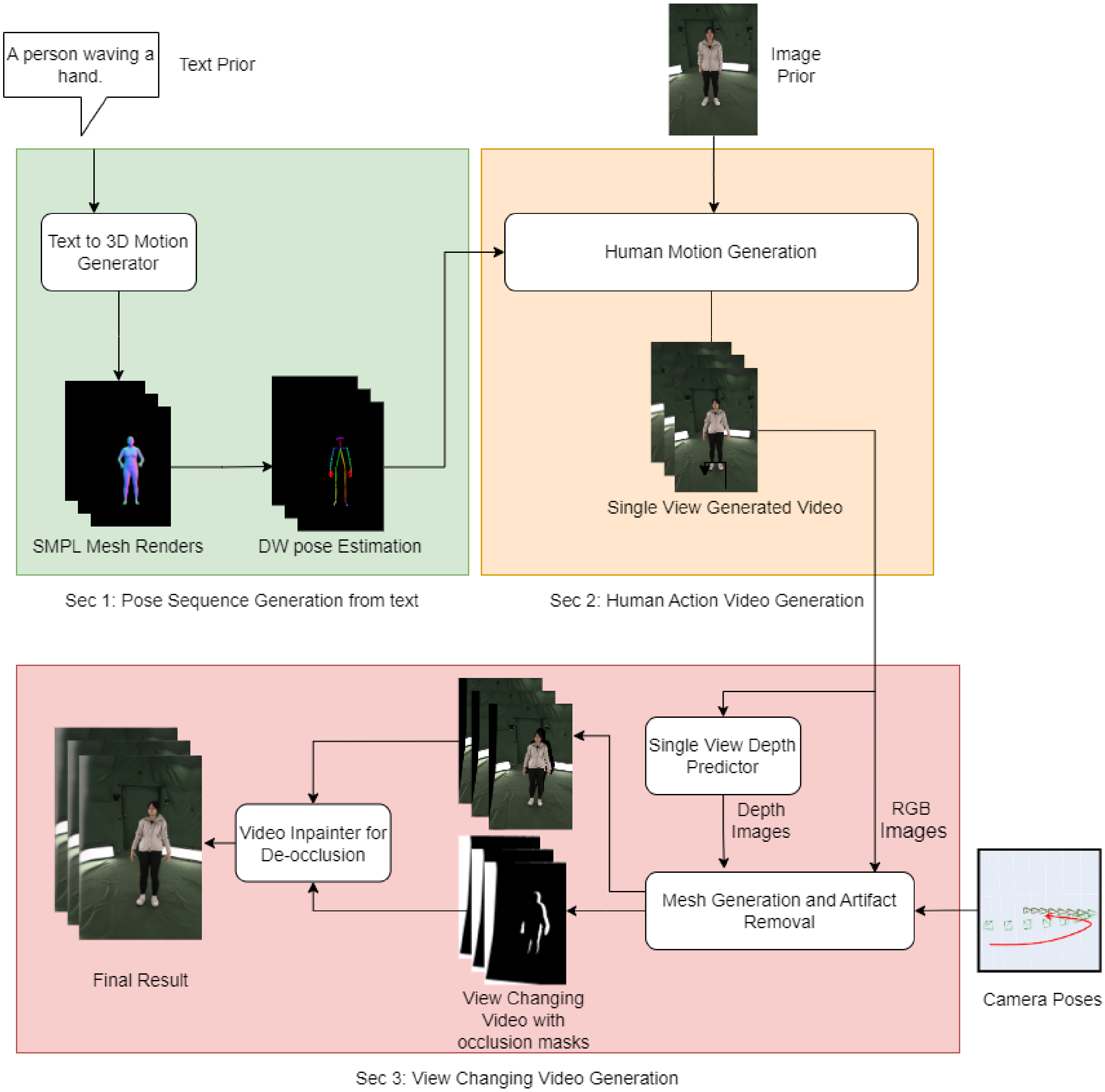
A person waving a hand.
Text Prior
Image Prior
Text to 3D Motion Generator
SMPL Mesh Renders
DW pose Estimation
Human Motion Generation
Single View Generated Video
Sec 1: Pose Sequence Generation from text
Sec 2: Human Action Video Generation
Single View Depth Predictor
Depth Images
RGB Images
Mesh Generation and Artifact Removal
Camera Poses
View Changing Video with occlusion masks
Video Inpainter for De-occlusion
Final Result
Sec 3: View Changing Video Generation

Figure 3.1: The overview of our pipeline.

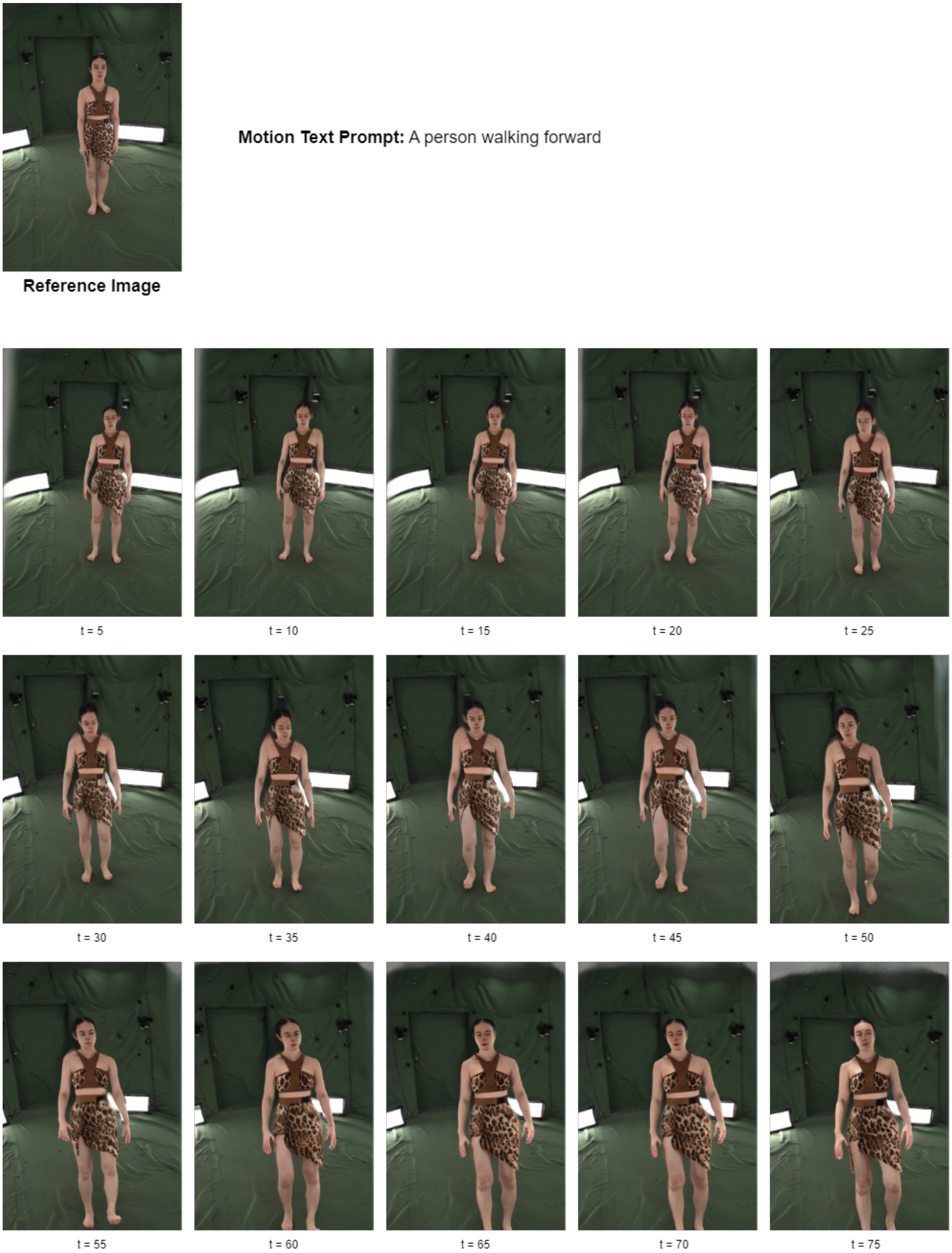

Figure 3.2: Demo 1 of the full pipeline.

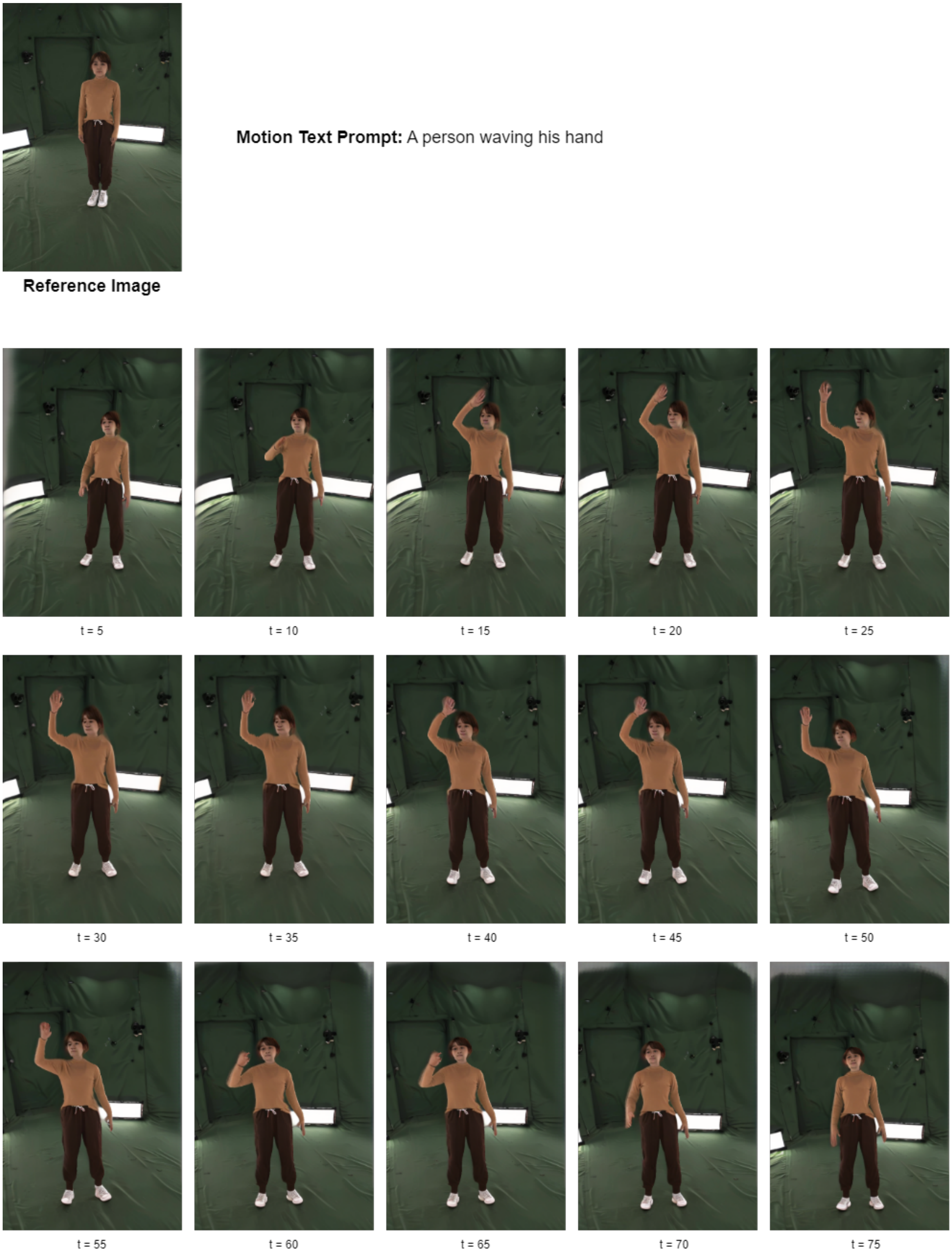

Figure 3.3: Demo 2 of the full pipeline.

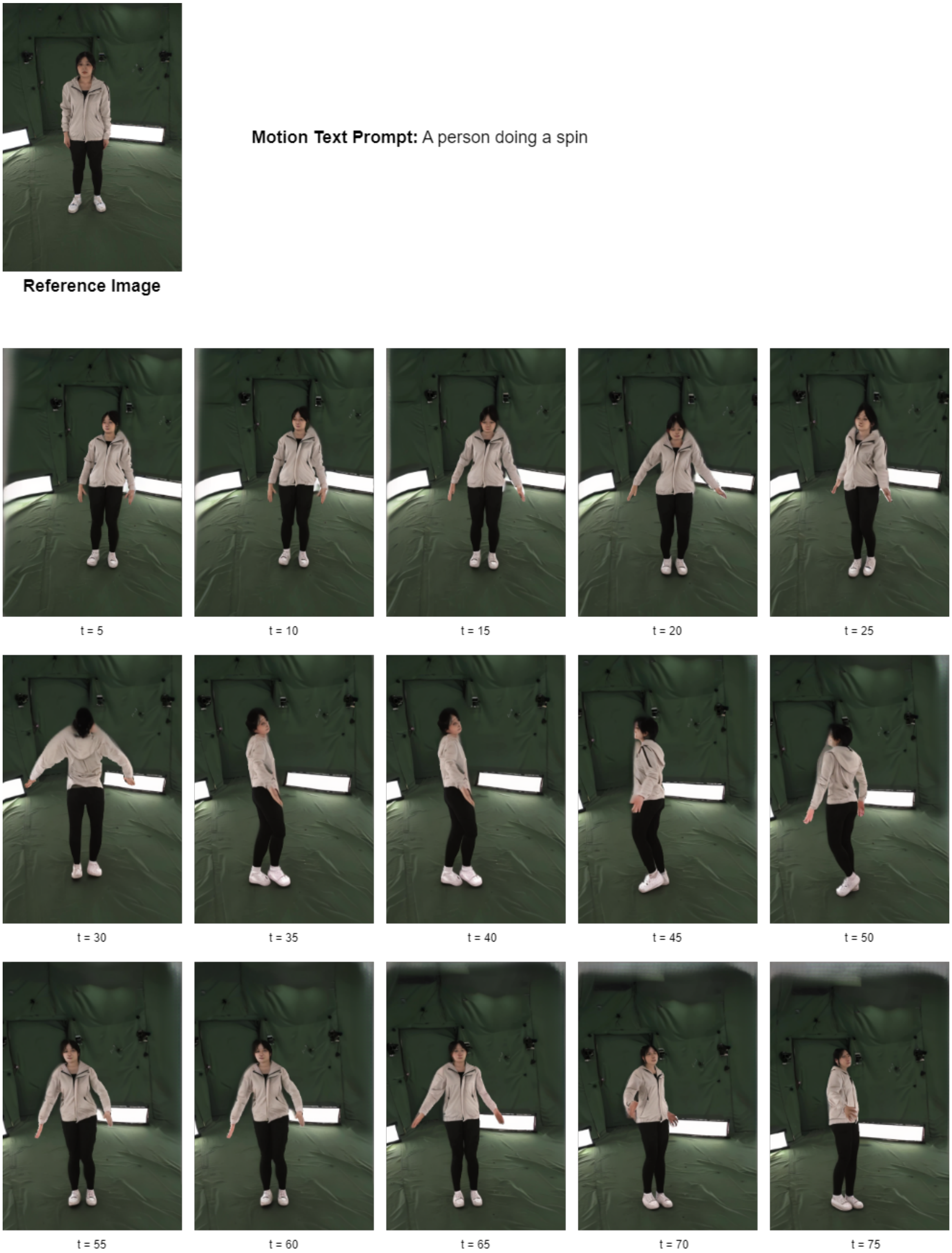

Figure 3.4: Demo 3 of the full pipeline.

3.1.1 Section 1: Pose Sequence Generation from Text

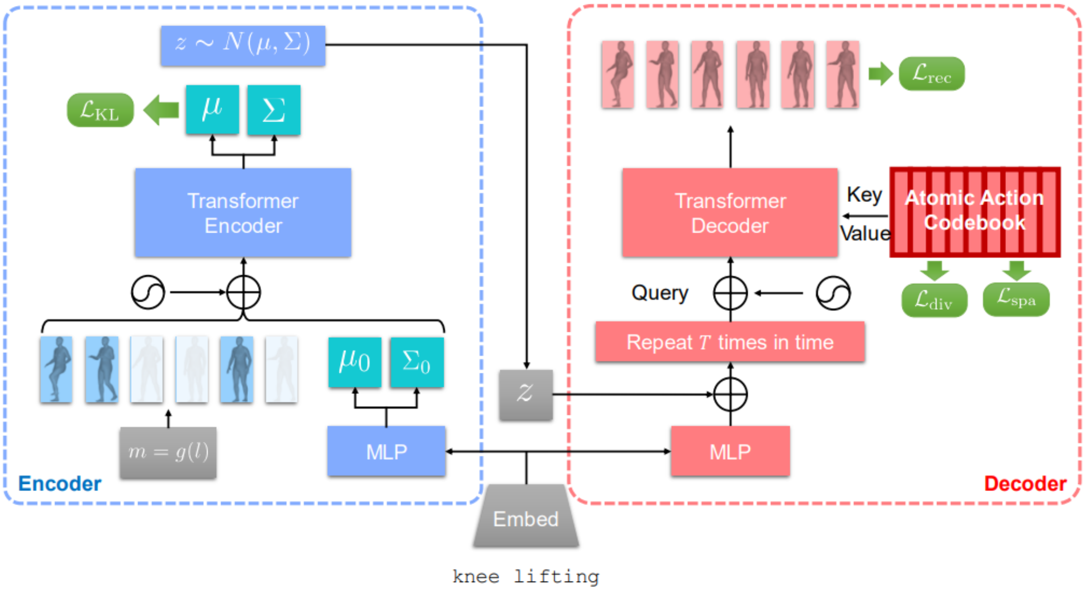

Figure 3.5: Pipeline of ATOM.

The initial stage of the pipeline is dedicated to generating pose sequences based on given textual instructions. To achieve this, we employ off-the-shelf language-guided motion synthesis models that are specifically designed for this task. This process significantly reduces the need for manual pose scripting, making the generation of human action sequences more accessible and less labor-intensive. In our experiments, we utilize the ATOM [19] model for this task.

ATOM, or ATomic mOtion Modeling, is a state-of-the-art model for generating motion sequences from text. Given a dataset of text-motion pairs $\{(y_i, M_i)\}$, where y_i represents text and M_i is a motion representation $M_i = [p_1, \ldots, p_T]$ of SMPL body poses at each time step t, the model utilizes a Conditional Transformer VAE (Variational Autoencoder) to align motion representations with text. The encoder captures the structure of the motion sequence and transforms it into a compact latent representation, while the decoder generates motion sequences from this latent representation and the corresponding text embeddings.

The learning objective of ATOM is consistent with standard CVAEs, including reconstruction loss and Kullback-Leibler (KL) divergence.

$$L_{\text{rec}} = \frac{1}{T} \sum_{t=1}^{T} \|p_t - \hat{p}_t\|_2^2$$

$$L_{\text{CVAE}} = L_{rec} + w_{KL} L_{KL}$$

A distinctive feature of ATOM is the integration of an atomic action codebook into the decoder as a key-value pair. The underlying principle is that motion sequences can be decomposed into a series of smaller, atomic actions. These actions are stored in a learnable matrix $A \in \mathbb{R}^{N \times D}$, where N represents the number of atomic actions, and D is the hidden dimension corresponding to the latent space of each action.

To ensure the effectiveness of this decomposition, ATOM introduces two additional loss functions: L_{div}, a diversity constraint that ensures the uniqueness of atomic actions.

$$L_{div} = \|AA^T - I\|_F$$

$\|.\|_F$ is the Frobenius norm. This objective forces the atomic action codebook matrix to be orthogonal, i.e, Unique from each other.

Another added constraint is,

$$L_{spa} = -\sum_{l} \sum_{h} max(H_{l,h})$$

$H_{l,h}$ is the attention map. This encourages sparsity by maximizing the attention weights in the cross-attention layer. This approach prevents over-segmentation of actions and ensures that the atomic actions relating to a sequence are unique and distinct.

ATOM is particularly well-suited for our needs because it works solely on text prompting, allowing it to generate nuanced and contextually appropriate pose sequences. Moreover, its new additions like atomic action codebook, allows model to produce smoother motion and motion transitions.

The output poses of ATOM are SMPL[20] poses. The Skinned Multi-Person Linear Model (SMPL) is a 3D human body model derived from body scans encompassing a wide range of poses. This model is represented by meshes containing approximately 7,000 3D points.

SMPL[20] parameterizes these points in terms of pose θ, shape/identity β, and soft tissue dynamics σ. To manage the high-dimensional shape parameters β, Principal Component Analysis (PCA) [21] is employed to obtain a lower-dimensional representation for each mesh identity.

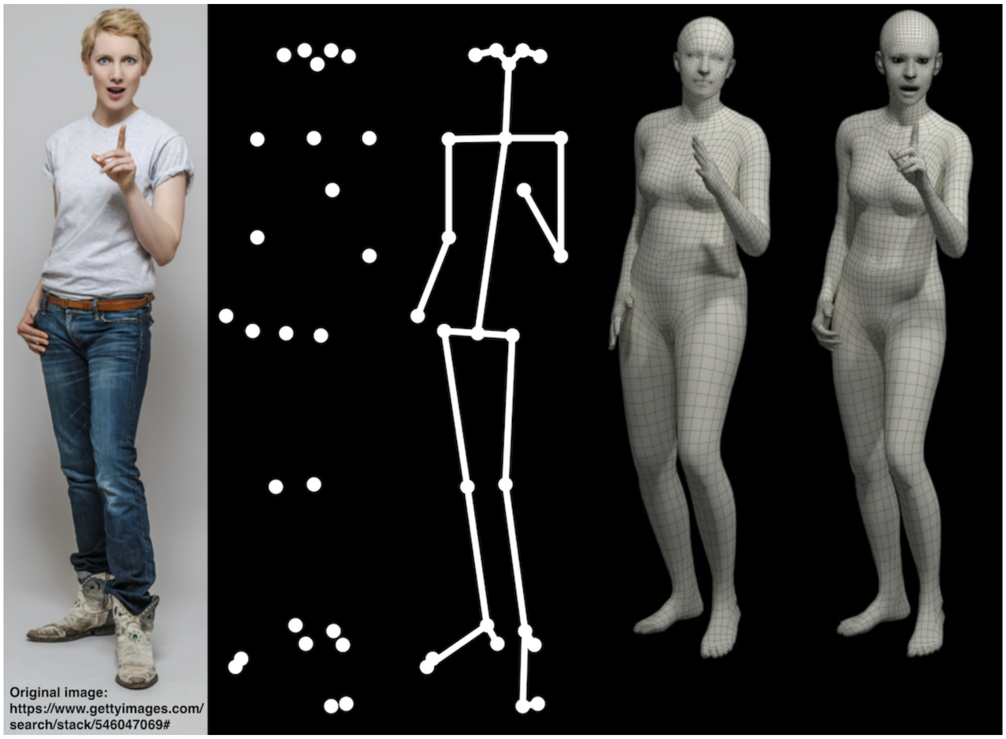

Figure 3.6: RGB Image, joints, skeleton, SMPL and SMPL-X [22]

The mesh is reconstructed using Linear Blended Skinning (LBS), which involves blending a template mesh T using joints $J \in \mathbb{R}^{3K}$, pose parameters $\vec{\theta} \in \mathbb{R}^{3K}$, blend weights $W \in \mathbb{R}^{N\times K}$, and a template $T \in \mathbb{R}^{3N}$. LBS works by linearly combining the transformed vertices of the template mesh based on their proximity to joints and the associated blend weights. The formula for this operation is:

$$W(T, J, W, \vec{\theta}) \rightarrow \text{Vertices}$$

While SMPL uses LBS, it incorporates a more complex model where the template T_F is a function of both $\vec{\beta}$ and $\vec{\theta}$. Additionally, the joints J are parameterized as a function of the shape β.

For 3D representation, SMPL is the most commonly used pose. These poses are low dimensional representation of the full mesh. An alternative for basic SMPL is SMPL-X [22] or expressive which includes detailed face and hand models in it. Number of joints in SMPL can vary based on datasets. HumanML3D has 22 joints and KIT-ML has 21 joints.

Once the pose sequences are generated, they need to be formatted for use in the subsequent stages of the pipeline. Human action video generation models in Section 2 typically accept pose data using 2D poses. Since most of these models were trained on static non changing cameras, they wernt built using 3D poses. Most commonly used 2D poses are OpenPose [23], DensePose [24] or DWPose [25], three widely-used 2D pose estimation frameworks.

OpenPose[23] is a notable 2D pose model that identifies 135 keypoints, which include body, face, hand, and foot keypoints. This model utilizes a Convolutional Neural Network (CNN) designed to detect keypoints in real-time for multiple individuals simultaneously. An alternative variant, known as the Body-25 model, focuses solely on the 25 body keypoints, omitting the face, hand, and foot keypoints when such detailed information is not required.

To convert SMPL (Skinned Multi-Person Linear) models to OpenPose format, there are three main approaches:

Using a Body Model Regressor: A joint regressor can be trained to map SMPL/SMPL-X joints to OpenPose joints. This method offers high accuracy and speed. However, literature typically provides regressors only for body-18 and body-25 keypoints. While facial keypoints can be derived from face contours in models like SMPL-X, such conversions are not feasible for the basic SMPL model. It might be because face, hand and foot keypoints are hard to regress due to their close proximity with nearby points.

Matching Corresponding Keypoints Across Conventions: This method involves mapping keypoints from one convention to another, but it may result in missing values or inaccuracies, particularly depending on the training specifics of the model. While direct conversion from SMPL to OpenPose 135 keypoints is not feasible, conversion from SMPL-X to OpenPose 135 keypoints is possible. In our case, we tested SMPL-X to Openpose using MMHuman3D's [26] inbuilt converter for DNA Rendering dataset with good success. ATOM's outputs however are SMPL. To get this working, SMPL first needs to be converted into an SMPL-X model that can be mapped to Openpose (like SMPLify-X[22] version or MMHuman3D version). This can be done by regressing SMPL mesh with SMPL-X mesh which is significantly time consuming. For this project we tested converting to Body-25 model this way but full Openpose has not been tested.

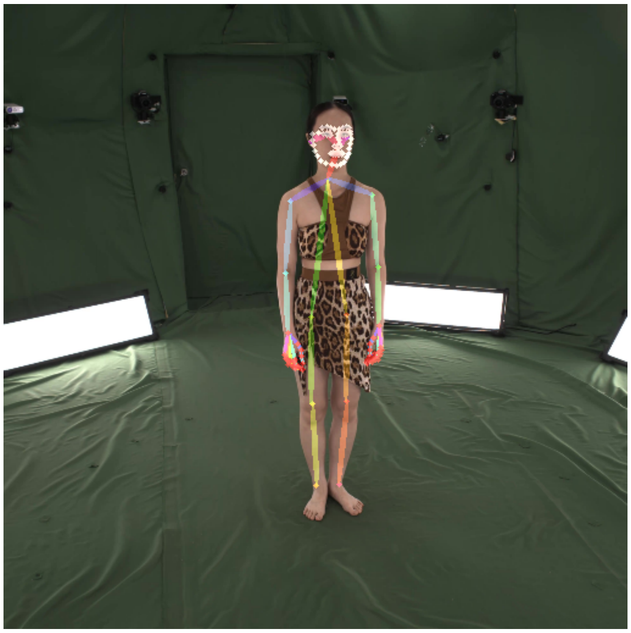

Figure 3.7: SMPL converted to openpose and overlayed on RGB image using HumanMM3D [26]

Rendering the Mesh Per Time Step and Running the Detector Directly: This approach involves rendering the SMPL mesh for each time step and then applying the detector. Although this method may not be as accurate as using a model regressor, it can still yield satisfactory results. However, it may struggle with occlusions.

DWPose [25] is another comprehensive pose estimation framework designed to track body, hand, and face poses across multiple individuals in images. It employs a two-stage distillation process to enhance pose detection accuracy. Additionally, DWPose is optimized for integration with ControlNet models in diffusion-based generation frameworks.

Unlike Openpose, generating DWPose from SMPL pose is not straightforward and contains a different set of challenges. Since there is no direct conversion method available, the DWPose detector must be applied to SMPL-generated meshes. This involves rendering normal maps from SMPL using a static camera setup and then feeding these maps into the DWPose detector. While this approach is faster than the SMPL-to-SMPL-X conversion, it can introduce artifacts and distortions, necessitating careful handling during the subsequent video generation phase.

In our experiments, we use `Open3D` [27] to generate and render the SMPL meshes. For the SMPL mesh corresponding to pose p_t at time step t, we employ a static camera with a large focal length to render the mesh using a normal renderer. Since the models discussed in Section 2 are designed for static cameras, we opted for this approach. However, SMPL-generated meshes often exhibit significant movement. Therefore, using a camera with a large focal length positioned at a greater distance allows us to cover a substantial area and accommodate the mesh's motion effectively.

In the second stage of our pipeline, we focus on generating forward-view videos of human actions using off-the-shelf Human Action Video Generators. For this purpose, we utilize UniAnimate [28].

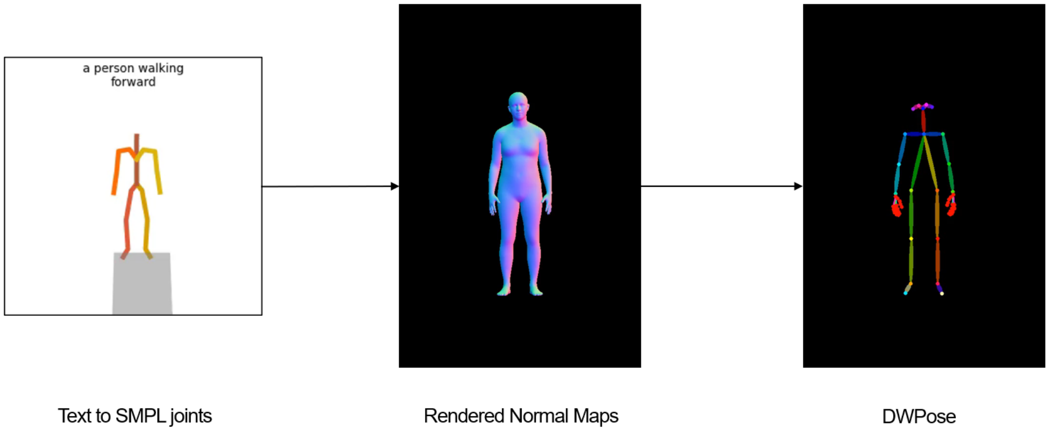

Figure 3.8: DWPose Generation Process

3.1.2 Section 2: Human Action Video Generation

UniAnimate is a state-of-the-art model designed specifically for generating realistic human action sequences. The driving pose sequence for UniAnimate is provided by DWPose, as detailed in the previous section. This integration requires the application of the DWPose generation process described earlier.

Previous methods in this domain, such as MagicAnimate [29], MagicPose [30], and DisCo [31], employ separate 3D U-Net networks to encode and retain appearance information. Unlike these methods, UniAnimate does not use this approach. Instead, UniAnimate proposes using a unified video diffusion architecture. This architecture encodes all relevant information—including the reference image, reference pose, and driving pose—into a single input for the diffusion models. By concatenating or stacking this information, UniAnimate enables joint modeling of appearance and motion, improving the cohesiveness of generated sequences.

Specifically, reference information is concatenated and broadcasted to the noised input shape, which is then stacked with noised input which was concatenated with driving pose.

Additionally, UniAnimate introduces a robust technique for generating long video sequences. While previous methods often rely on a sliding window approach for generating long sequences, this can lead to discontinuities between windows. Unianimate shows that using just last frame from previous window, and conditioning it with noise inputs for next window, we can get rid of discontinuity issues among different windows.

Furthermore, UniAnimate demonstrates that replacing temporal transformers with temporal mamba significantly reduces computational complexity while achieving comparable results. This modification streamlines the process without compromising the quality of the generated video sequences.

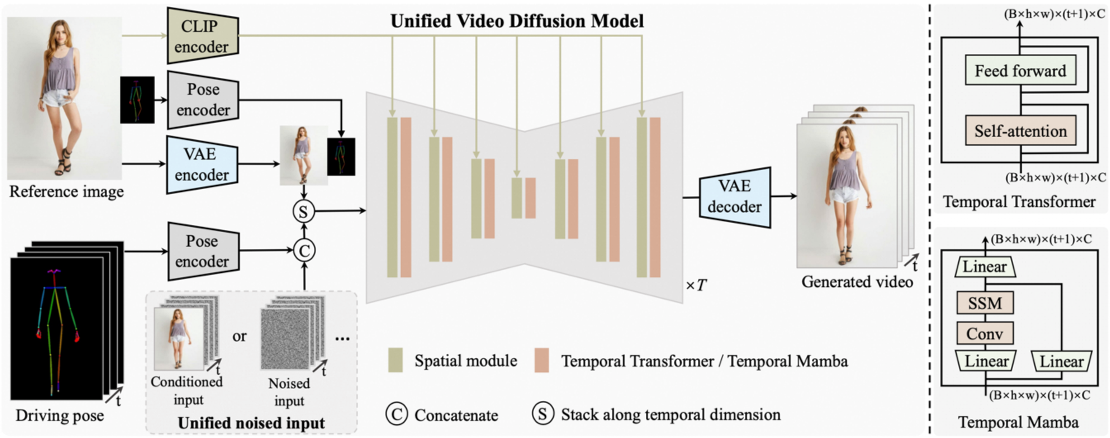

Figure 3.9: Pipeline of UniAnimate

By leveraging UniAnimate, our pipeline can efficiently produce high-quality forward-view videos based on the pose sequences generated in Section 1. The model's ability to handle large sequences of frames makes it particularly effective for generating videos that require a high degree of temporal continuity. This capability is crucial for applications where fluid and realistic motion is essential, such as in fitness demonstrations or dance tutorials.

3.1.3 Section 3: View Changing Video Generation

The third and final stage of our pipeline is dedicated to transforming the generated forward-view videos into dynamic, view-changing videos. This is achieved through depth-based view synthesis techniques, which allow for the creation of novel perspectives from the orig-

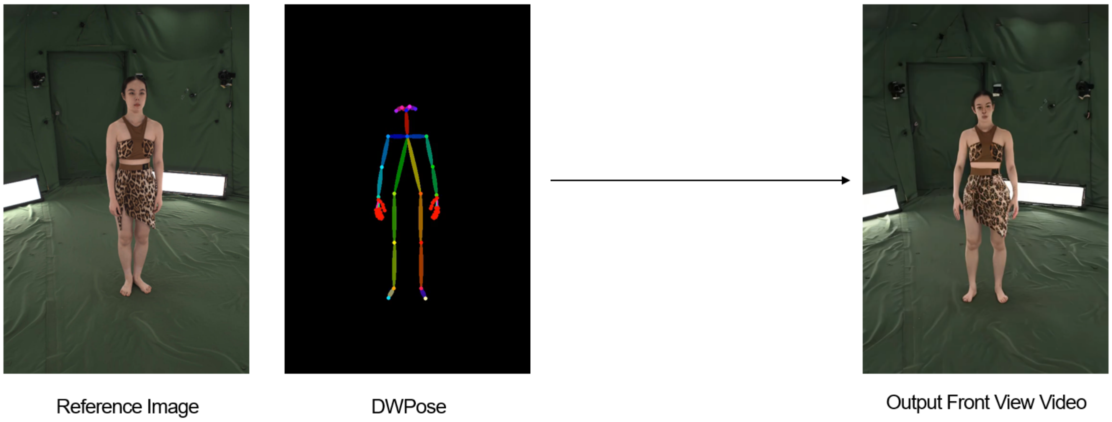

Figure 3.10: Front View Output Generated using DWPose and UniAnimate

inal video frames. To accomplish this, we first generate metric depth for each frame using DepthAnything [32], a tool designed to produce accurate depth maps from standard video frames.

DepthAnything [32] represents the state-of-the-art in monocular depth estimation. It utilizes a Dense Prediction Transformer architecture and is trained with a universal training model, enabling it to effectively leverage data from diverse sources. This approach allows DepthAnything to outperform previous models such as MiDaS [33] in depth accuracy and robustness.

In our experiments, DepthAnything's metric depth video exhibits significantly less flickering compared to MiDaS. For our evaluations, we employ metric depth rather than relative depth to get better background modeling. However, because metric depth values are not inherently bounded, the final depth values are not normalized. This is because normalizing these values would produce results comparable to those obtained with relative depth, rather than absolute depth.

Once the depth maps are generated, we explore two distinct methods for synthesizing new views. The first method involves generating a point cloud from the RGBD images and rendering this point cloud from new viewpoints. This technique effectively simulates

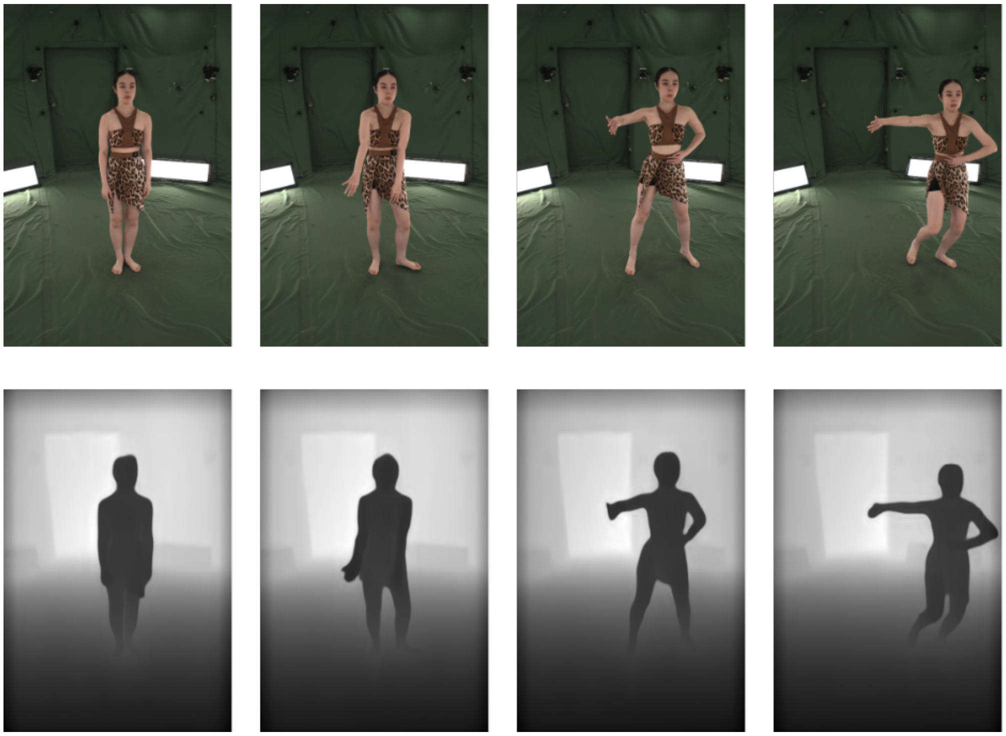

Figure 3.11: Video and Depth Images

different perspectives but introduces occlusions—areas where information is missing due to the projection of 3D points into 2D space.

Given a 2D homogeneous pixel coordinate $(u, v, 1)$ and depth Z, the 3D homogeneous world coordinate point $(X, Y, Z, 1)$ can be recovered as follows:

1. Apply depth to the pixel, Specifically, we will assume that the point is Z distance from camera center on the ray.

$$points = \begin{bmatrix} (u_{\text{pixel}} + 0.5) * z \\ (v_{\text{pixel}} + 0.5) * z \\ z \end{bmatrix}$$

2. Use inverse of intrinsic matrix K to back-project the 2D pixel coordinates to a ray in

camera space.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} (u_{\text{pixel}} + 0.5) * z \\ (v_{\text{pixel}} + 0.5) * z \\ z \end{bmatrix}$$

3. Apply inverse of extrinsic matrix to these points to get world coordinates

$$P_{world} = [R|T]^{-1}.P_{camera}$$

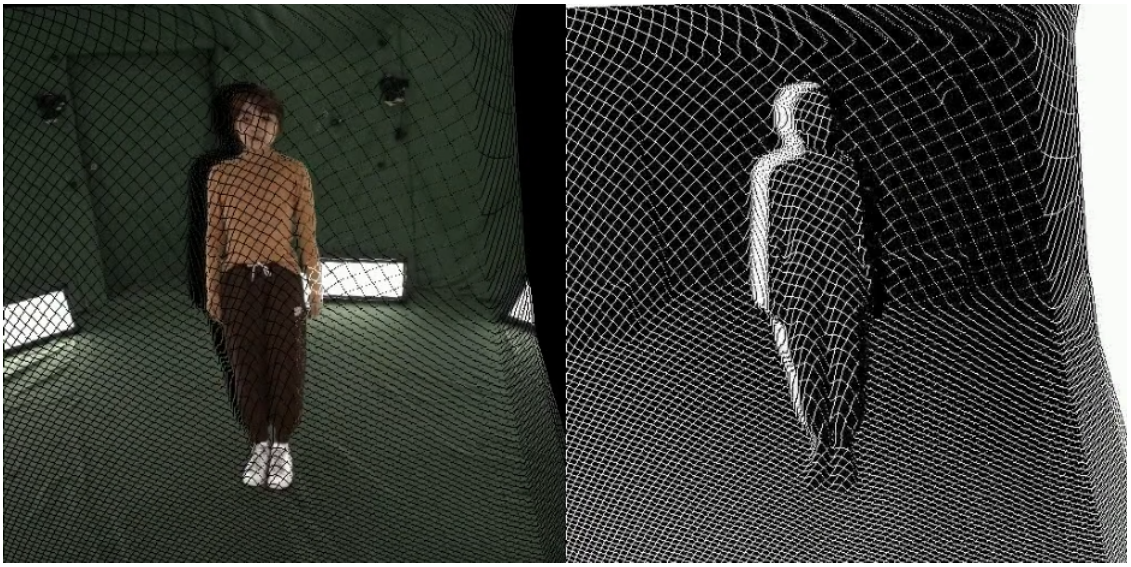

Figure 3.12: backprojected pointcloud and its occlusion mask

To address these occlusions, we employ an infilling process. A common approach is to use bilinear interpolation to fill the missing areas. In this method, bilinear interpolation is applied to the generated pointcloud at each timestep of the video. However, the results reveal that point cloud warping causes two types of occlusion areas: small/thin occlusions and large occlusions. Bilinear interpolation struggles to fill small or thin occlusions, especially in objects with low thickness. Although it can address larger areas, the absence of a temporal context leads to significant discontinuities between timesteps, making the filled areas appear

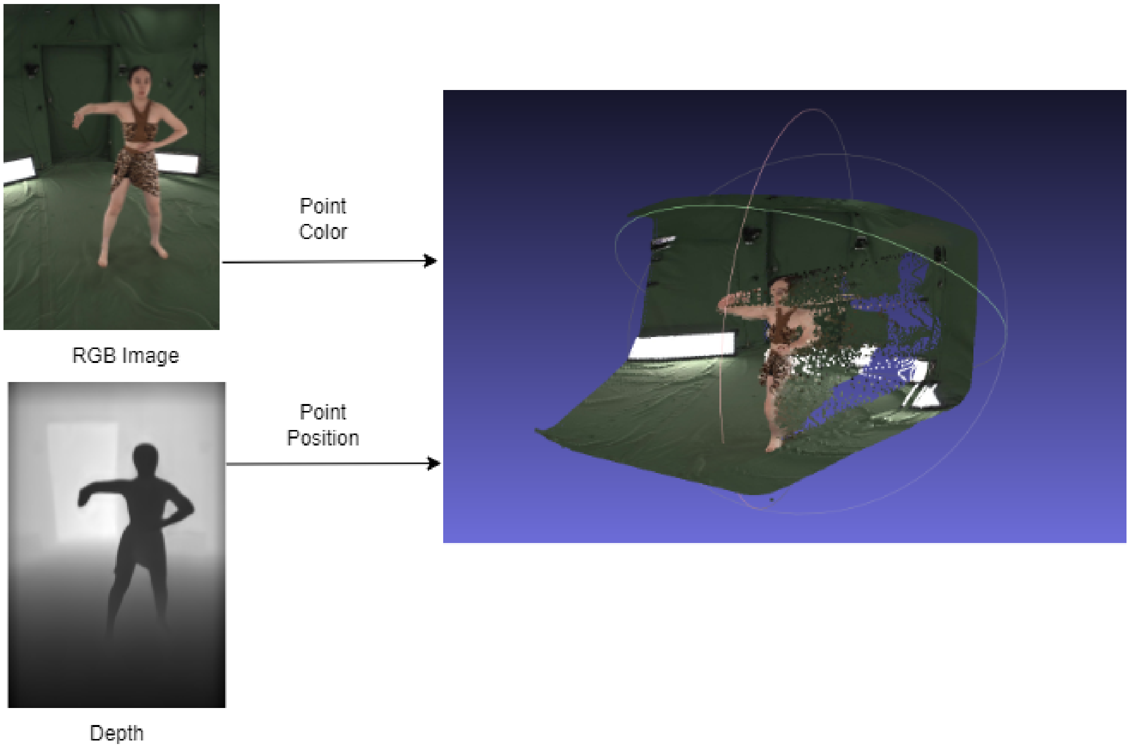

Figure 3.13: Pointcloud Generation

unnatural.

To overcome these limitations, we explore a second method that involves generating meshes from the depth maps. Unlike point clouds, meshes consist of interconnected triangles that interpolate colors from nearby vertices, effectively smoothing out many of the small holes that occur during view synthesis.

Meshes are typically generated using one of two main approaches:

Poisson Surface Reconstruction[34]**:** This technique is often used for building watertight surfaces and is more appropriate for datasets with dense, multi-view point clouds.

Ball Pivoting Surface Reconstruction[35]**:** It works by pivoting a ball around the points and connecting them to form a mesh. This approach is suitable for our scenario, as our points are generated from a single view.

The process is pretty straight forward. However, given that our data consists of an RGB image with corresponding depth information, such a complicated process is not necessary.

Since our data is an RGB image with corresponding depth, we can directly use neighbourhood pixel property to generate a good mesh.

This method involves forming triangles for each pixel by connecting it with its two nearest neighbors based on depth information.

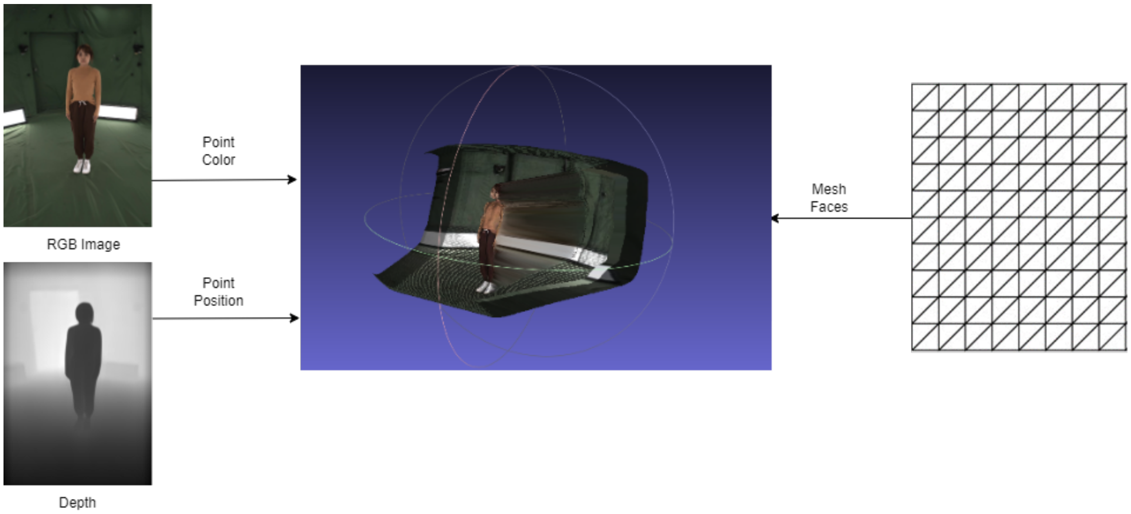

Figure 3.14: Mesh Generation

However, depth predictions from single-image depth estimation models are not always precise, and the generated triangle meshes inadvertently connect foreground objects with the background, leading to unnatural artifacts. To mitigate this, we use a filtering process that identifies and removes triangles with overly stretched edges/pixels, by thresholding the length of the triangle. By refining the mesh in this way, we reduce the occurrence of artifacts, leading to cleaner and more realistic view-changing videos.

Specifically,

1. Points are projected to world space like before. Each point is identified by its pixel location.

2. Two faces can be formed with each pixel. For pixel $[u, v]$, a triangle is formed for following tuples

$$face1 = ([u, v], [u + 1, v], [u + 1, v + 1])$$

$$face2 = ([u, v], [u + 1, v + 1], [u, v + 1])$$

The order of points in the faces matter since normals are generated based on the it.

3. Artifact removal: Edge distances are distances between two points of the triangle.

$$EdgeDistances = (\|p2 - p1\|_2, \|p3 - p1\|_2, \|p3 - p2\|_2,)$$

if any of the edge distances are greater than threshold, do not generate that face. These are the faces that usually connect background to foreground.

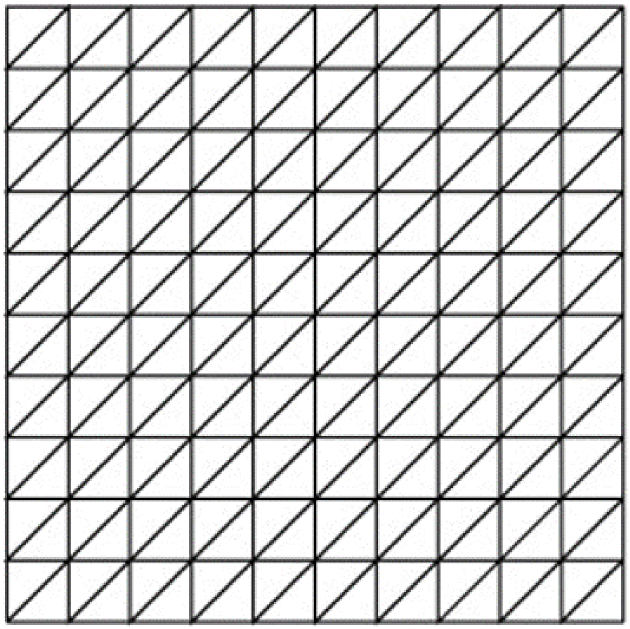

Figure 3.15: Regular Triangle Mesh[36]

After generating the meshes, any remaining occlusions or artifacts are addressed using video inpainting techniques. For this task, we utilize ProPainter [37], a state-of-the-art video inpainting model that excels in extrapolating missing flow information in masked areas before inpainting them. This approach significantly improves the visual quality of the final view-changing video, producing results that are both more accurate and visually coherent.

ProPainter operates with three main components. First, it includes a recurrent flow completion module that identifies and addresses corrupted flow in masked areas, completing the flow both forward and backward across frames. Once the flow is completed, the model fills the masked regions by using the flow information from the neighboring frames—specifically, pixels from the previous frame (t-1) and the subsequent frame (t+1) are used to fill in the gaps at the current frame (t).

After this initial filling, the remaining unfilled areas, known as residual flow areas, along with the already filled regions, are refined and completed using mask-guided flow transformer blocks. These blocks propagate features effectively across the video sequence, ensuring a coherent and continuous visual flow. ProPainter's efficiency is further enhanced by its sparse strategy, which processes only a subset of tokens, making it both fast and robust, particularly in handling long sequences.

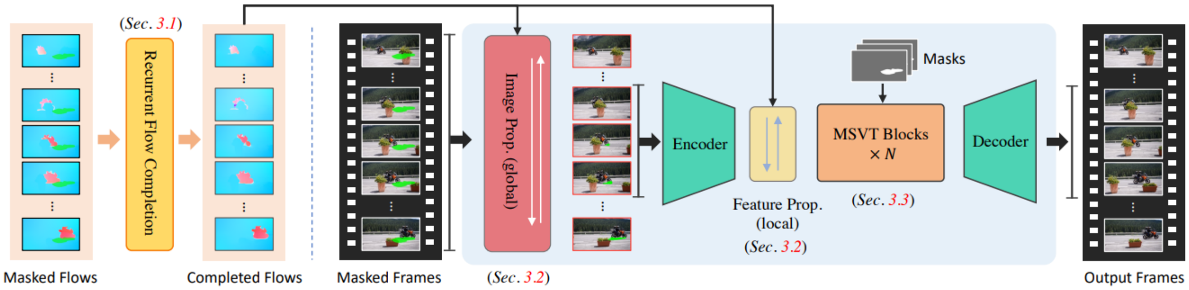

Figure 3.16: ProPainter Architecture

Together, these methods allow our pipeline to produce compelling and dynamic videos that simulate minor viewpoint changes, enhancing the overall visual experience without the need for complex 3D modeling or extensive manual intervention.

3.1.4 Optional Component: Changing background

Incorporating background changes during the generation of view-changing videos is an optional but impactful component of our pipeline. This process involves replacing the background after generating the forward-view video but before initiating view-changing video synthesis. After Section 2, once the forward-facing video has been generated, the background can be changed using segmentation techniques. By employing models such as the Segment Anything Model (SAM) [39] or the Grounded Segment Anything Model (Grounded SAM) [40] with ”person” as the prompt, we can isolate the human figure from each frame of the video. These segmented frames can then be composited onto a new background, allowing for a seamless transition into the view-changing video generation process in Section 3.

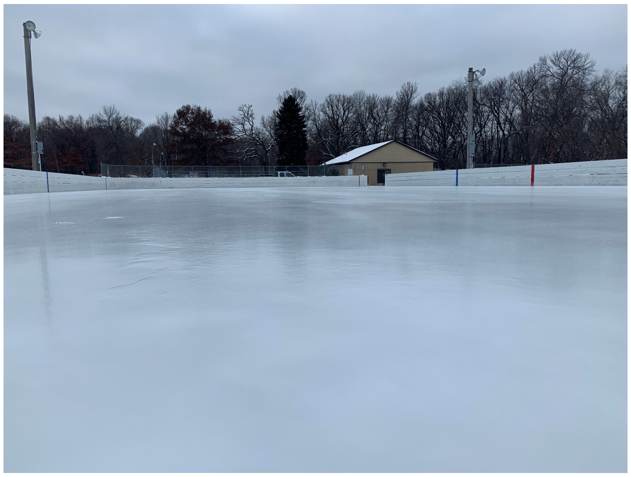

Figure 3.17: Image of a real Ice Rink used as Background [38]

Initially, we experimented with extracting and compositing the depth maps alongside the RGB images. However, this approach led to suboptimal results, as the composited depths introduced significant artifacts and distortions. Instead, we found that passing the composited images directly into the DepthAnything [32] model yielded much better outcomes, as it allowed for more accurate depth estimation and integration with the new background.

The results, as shown in the accompanying figures, demonstrate that while the compositing process effectively integrates the human figure with the new background, it can also introduce minor distortions relative to the original video generation. Despite these challenges, the final output still successfully maintains the coherence of the scene, with the new background blending smoothly into the view-changing video.

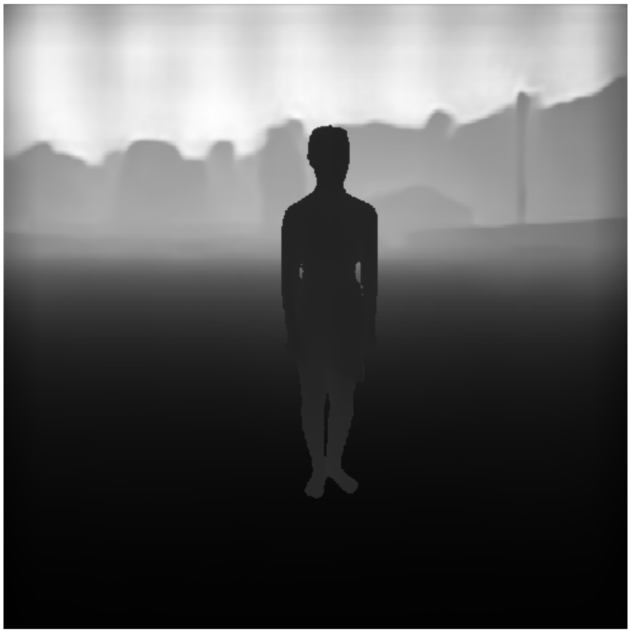

Figure 3.18: Composited Depth

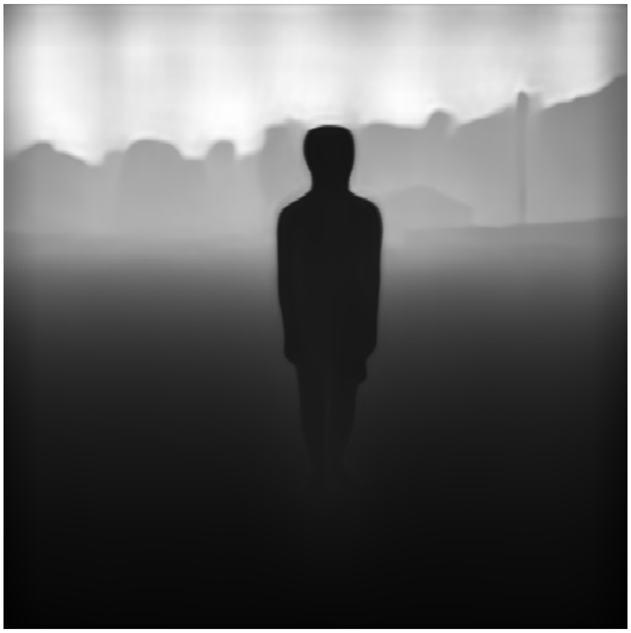

Figure 3.19: Generated Depth

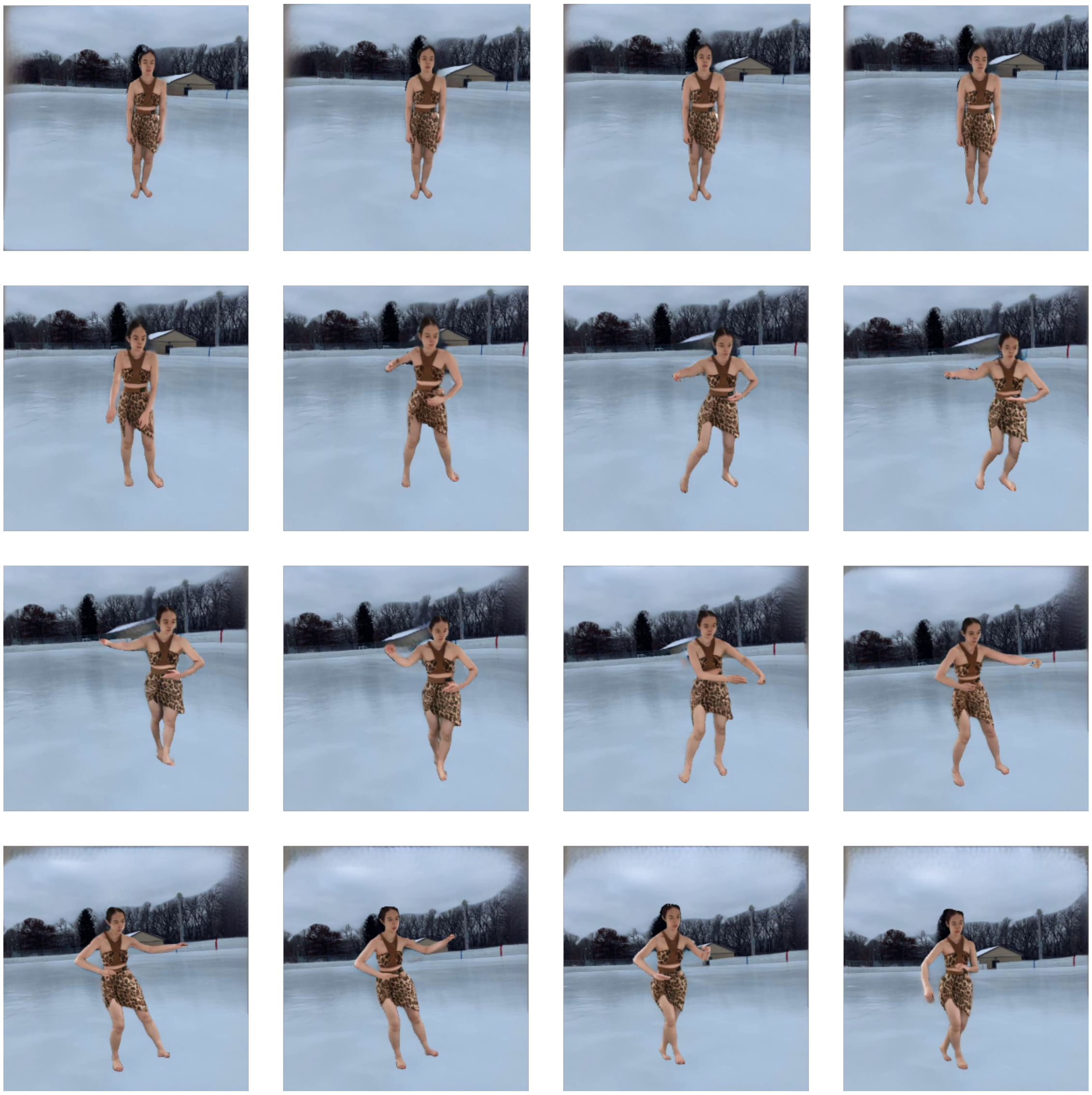

Figure 3.20: Rendering new background

Chapter4

Experiments and Results

4.1 Dataset

For Section 1, 5 text,motion pairs were selected from HumanML3D [41] dataset. Qualitative metrics are calculated for this subset.

For our experiments, we utilized the DNA Rendering dataset [42], which provides a diverse set of video sequences well-suited for evaluating components of our proposed pipeline. The dataset comprises of more than 500 sequences, each consisting of 224 frames. Of these, 5 sequences are selected for evaluation of the pipeline. These sequences are captured from multiple viewpoints, however we will only be using the main/front view, henceforth called as reference view.

The DNA Rendering dataset features subjects with varied clothing, body shapes, and motion patterns, providing a comprehensive basis for testing the robustness and versatility of our pipeline. This variability ensures that our evaluation covers a wide spectrum of scenarios, reflecting real-world applications and challenges. Additionally, the availability of 3D optimized SMPL and SMPL-X models, along with object masks, facilitates accurate pose extraction and depth map generation, further enhancing the reliability of our experiments.

4.2 Evaluation Process

Given the uniqueness of our approach, particularly its reliance on single-view depth prediction, we lack ground truth information and a baseline to perform a direct evaluation. To address this, we utilize the DNA Rendering dataset to compare the outputs of Section 2 and Section 3 of our pipeline. This comparison allows us to assess the effectiveness of our forward-view video generation and view-changing techniques. Additionally, we conduct ablation studies to investigate the impact of various components within the pipeline, providing insights into the contributions of each stage and the overall performance of the proposed methods.

4.3 Quantitative results for Section 1

Table 4.1 presents the FID (Frechet Inception Distance) [43] and R Precision scores for the text-motion pairs generated using the HumanML3D model. The results indicate that the performance of this subset closely aligns with the scores reported in the original paper.

The FID score, calculated between the ground truth motion and the generated motion, serves as a key metric in this evaluation. Specifically, a contrastive model [44] was trained to map text to motion, and the features extracted from this model were used in the FID calculation. A higher FID score indicates better alignment between the generated and real motion sequences.

For R Precision, the evaluation involved selecting each motion sequence's ground truth text and comparing it against 30 other random texts from the dataset. Motion vectors were extracted, and the Euclidean distance between these vectors was calculated. If the ground truth text ranked within the top k results, it was considered a successful retrieval. The average accuracy was then computed for the top three positions.

Table 4.1: Quantitative results for Section 1

Method	**FID**	**R Precision**
Real Motion	0.002	0.797
In paper	1.691	0.569
Our Subset	3.284	0.482

4.4 Quantitative results on Reference View

To evaluate the performance of Sections 2 and 3 of our pipeline, we leverage the ground truth data provided by the DNA Rendering dataset. This allows us to generate outputs for both sections and compare them against the reference view using qualitative metrics. For Section 2, we assess the forward-view videos generated from the pose sequences. For Section 3, we generate a mesh, remove artifacts and inpaint any holes using video inpainting. This allows us to understand how well the pipeline is able to replicate the ground truth data.

Table 4.2: Evaluation results on Sec 2 and Sec 3 using reference view videos.

Seq Name	**Section**	**PSNR[45]**	**SSIM[45]**	**FVD[46]**
0034_04	Sec2	25.57	0.894	209.75
	Sec2 + Sec3	24.94	0.875	342.92
0088_09	Sec2	27.33	0.897	174.11
	Sec2 + Sec3	26.39	0.880	212.81
0092_11	Sec2	29.72	0.936	328.98
	Sec2 + Sec3	28.21	0.917	385.13
0190_11	Sec2	27.53	0.913	145.75
	Sec2 + Sec3	26.46	0.894	200.93
0813_01	Sec2	28.56	0.900	293.76
	Sec2 + Sec3	27.2	0.882	395.26

The results show that Section 3 outputs have minimal impact on PSNR and SSIM but the increase in FVD is significant. This indicates that artifact removal and inpainting process tends to distort the output video making it slightly more unrealistic and dissimilar from ground truth.

4.5 Quantitative results on Novel View

To conduct a quantitative evaluation of the generated view-changing videos, a consistent novel view path is essential. For our analysis, we generate a spherical path within the same plane as the reference camera. This setup involves positioning new cameras which are facing at target which is 2 units along the forward axis of the reference camera, with a radius of

0.2 units from the reference camera center. Due to the absence of ground truth data for these novel views, we employ no-reference quality metrics to both quantify the quality of the generated videos and facilitate comparative analysis.

4.5.1 No Reference Image Quality Assessment

For assessing image quality without reference, we use several parameters. First, we employ the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE)[47]. BRISQUE generates a quality score by measuring potential losses in ”naturalness” due to distortions. It analyzes the statistical features of locally normalized luminance coefficients and compares them to a model based on images of known quality. The resulting score ranges from 0 to 100, where a lower score indicates better image quality.

Additionally, we use the Blur Score, computed via the Variance of Laplacian. The Laplacian operator, a differential operator representing the divergence of the gradient, highlights rapid intensity changes and performs edge detection. By calculating the variance, we assess the level of blurriness in the image. A higher variance indicates better sharpness and less blurriness. These metrics are typically calculated per frame and averaged to provide an overall quality score.

4.5.2 No Reference Video Quality Assessment

For video quality assessment, we use the Disentangled Objective Video Quality Evaluator (DOVER) [48]. This model evaluates video quality from both technical and aesthetic perspectives, providing an overall score between 0 and 1. Trained on the DIVIDE-3k [48] dataset, DOVER’s score reflects how well the video performs compared to other videos in the dataset. This assessment helps gauge the overall visual and perceptual quality of the generated view-changing videos.

4.5.3 3D Correspondences

To evaluate the accuracy of novel view generation, we analyze high-confidence correspondences between the reference view and the novel views. We use the "LoFTR: Detector-Free Local Feature Matching with Transformers" [49] model to identify and match features across frames. By thresholding based on confidence scores, we determine the number of high-confidence correspondences. A higher number of these correspondences indicates a better alignment and accuracy of the generated novel views relative to the reference. We use a threshold of 0.8 confidence score.

Table 4.3: Novel view evaluation results on 0034_04 .

0034_04	**BRISQUE**	**Blur**	**DOVER**	**Corrs**
Reference Video	35.71	334.80	0.759	–
Depth Warping with Bilinear Interpolation	34.51	652.89	0.690	3111.91
Mesh Rendering with artifacts	35.91	226.65	0.698	3144.92
Mesh Rendering with video inpainting	34.69	193.27	0.746	3154.25

Table 4.4: Novel view evaluation results on 0088_09 .

0088_09	**BRISQUE**	**Blur**	**DOVER**	**Corrs**
Reference Video	35.69	228.40	0.787	–
Depth Warping with Bilinear Interpolation	36.24	465.59	0.734	3140.89
Mesh Rendering with artifacts	38.73	155.77	0.706	3156.78
Mesh Rendering with video inpainting	37.10	124.36	0.772	3184.45

Depth warping with bilinear interpolation consistently yields better scores on pooled single-image quality metrics but shows lower performance on video quality metrics and correspondences. This is consistent with our modeling, as bilinear interpolation relies solely on data from the current frame to interpolate information. Interestingly, even meshes with artifacts tend to score higher on DOVER and correspondences. However, video inpainting after

Table 4.5: Novel view evaluation results on 0092_11 .

0092_11	**BRISQUE**	**Blur**	**DOVER**	**Corrs**
Reference Video	41.64	247.26	0.843	–
Depth Warping with Bilinear Interpolation	42.49	361.04	0.787	3132.57
Mesh Rendering with artifacts	45.01	160.52	0.780	3135.83
Mesh Rendering with video inpainting	42.89	125.04	0.825	3143.63

Table 4.6: Novel view evaluation results on 0190_11 .

0190_11	**BRISQUE**	**Blur**	**DOVER**	**Corrs**
Reference Video	38.56	265.97	0.782	–
Depth Warping with Bilinear Interpolation	38.67	421.99	0.716	3257.54
Mesh Rendering with artifacts	41.11	175.01	0.725	3266.88
Mesh Rendering with video inpainting	39.06	139.04	0.756	3264.97

Table 4.7: Novel view evaluation results on 0813_01 .

0813_01	**BRISQUE**	**Blur**	**DOVER**	**Corrs**
Reference Video	40.71	205.46	0.827	–
Depth Warping with Bilinear Interpolation	38.91	360.08	0.777	3062.93
Mesh Rendering with artifacts	41.42	141.77	0.772	3078.33
Mesh Rendering with video inpainting	39.64	104.77	0.810	3083.21

artifact removal significantly improves the scores, with all metrics showing enhancements over sequences containing artifacts.

4.6 Qualitative outputs on Novel View

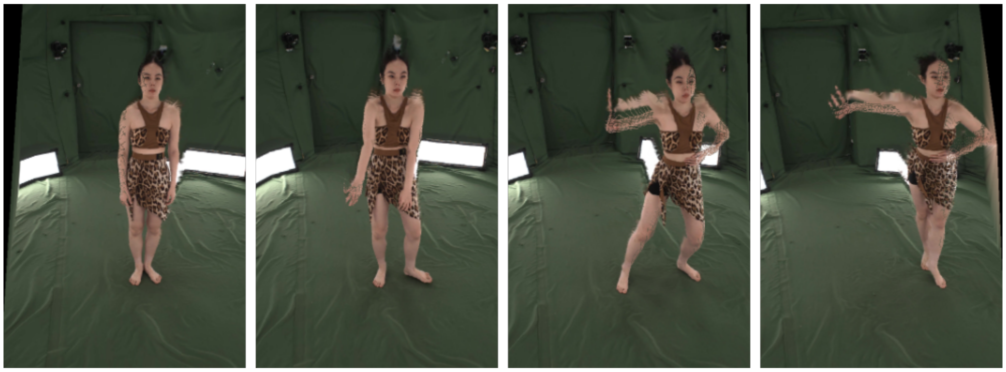

Figure 4.1: Depth Warping and Bilinear Interpolation

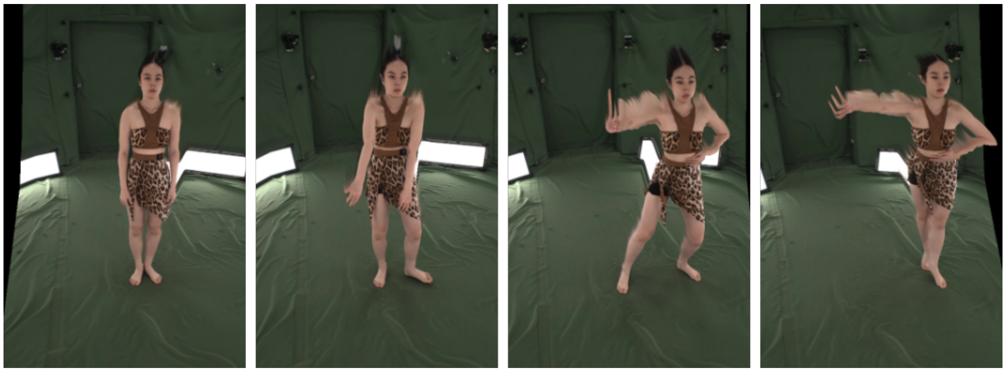

Figure 4.2: Mesh With Artifacts

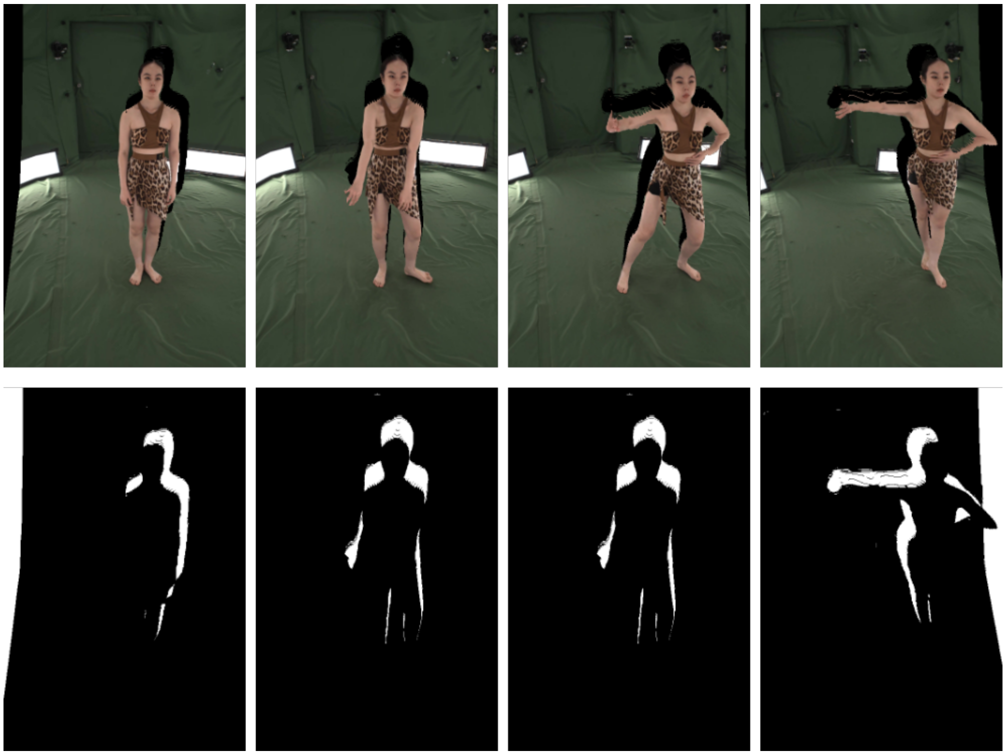

Figure 4.3: Mesh Without Artifacts and with occlusion masks

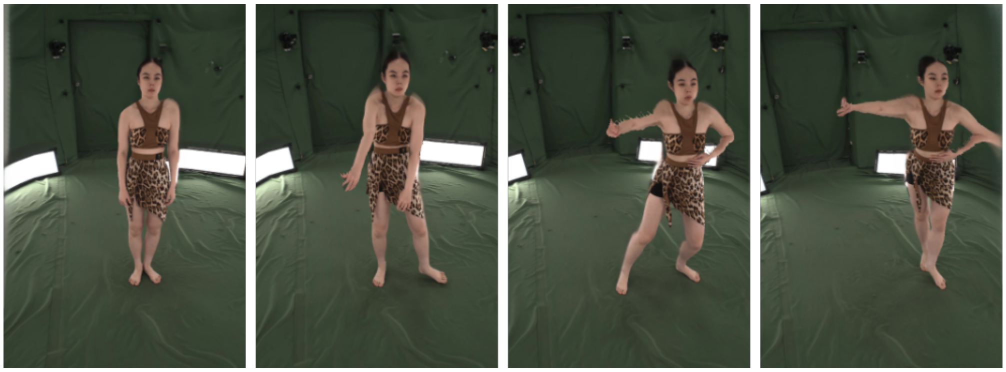

Figure 4.4: Mesh after video inpainting

4.7 Quantitative results on Viewpoint Change

We also need to evaluate how the output videos are affected by changes in the viewpoint. Specifically, we will vary the radius of novel view spherical path and obtain quantitative results to assess the impact.

Table 4.8: Quantitative results on Viewpoint Change .

Radius from center (units)	**BRISQUE**	**Blur**	**DOVER**	**Corrs**
0.1	35.59	170.62	0.765	3834.22
0.2	34.69	193.27	0.746	3154.25
0.3	33.34	201.68	0.725	2499.54
0.4	32.08	208.71	0.720	1913.45
0.5	30.75	219.73	0.676	1438.21

The results indicate that the output quality tend to degrade rapidly the further the viewpoint. The change in correspondences indicate that the model fails to figure out missing data using inpainting if the holes are too big.

4.7.1 Comparison between Reference Image and Final Output

We can evaluate how closely the generated final output frames resemble the reference image by leveraging the capabilities of the CLIP (Contrastive Language-Image Pretraining) [50] model. CLIP is a network trained on image-text pairs, and we utilize its powerful encoding capabilities for this comparison.

First, we extract the CLIP image features of the reference image. Then, for each frame in the generated sequence, we similarly extract the CLIP image features. The similarity between the reference image and each frame is quantified by calculating the cosine similarity between their respective features. All cosine similarity scores are then pooled and averaged to produce a final similarity score.

Sequence Name indicates the Sequence whose first frame was used as the reference image. Text indicates what text was used to generate the final video.

The results of these similarity scores are reported here. The closer the scores are to 1 the more similar their resemblance.

Table 4.9: Evaluation results on Sec 2 and Sec 3 using reference view videos.

Seq Name	**Section**	**Text**	**Clip-I Score**
0034_04	Sec 2	” a person walking forward”	0.891
	Sec 2 + Sec 3	” a person walking forward”	0.884
0092_11	Sec 2	” a person waving his hand”	0.823
	Sec 2 + Sec 3	” a person walking forward”	0.825
0190_11	Sec 2	” a person doing a spin”	0.911
	Sec 2 + Sec 3	” a person doing a spin”	0.894
0813_01	Sec 2	” a person kicking the air”	0.873
	Sec 2 + Sec 3	” a person kicking the air”	0.868

The clip-I scores from section 2 and section 3 results indicate that there are indeed deformations formed from the process.

4.7.2 Comparison between Text and Final Output

We can assess how closely the final output matches the input text description. Similar to the comparison with the reference image, this process involves using text features instead of image features.

However, since CLIP operates on individual frames, it lacks the capability to evaluate temporal dynamics. As a result, CLIP treats the motion described by the text as if it were occurring in each frame independently. To better account for the temporal aspect, we can employ a contrastive model designed for video analysis.

For this purpose, we use X-CLIP [51], an extension of CLIP that is adapted for video. X-CLIP samples 8-16 frames from the entire video and compares the features between the text and the video. It then outputs a probability indicating how well the text matches the video.

Limitations:

- X-CLIP only analyzes a subset of frames (8-16), which may not fully capture the entire video's content.

- X-CLIP is trained on generalized text, whereas our input is focused solely on motion descriptions. This is why the output probability of the model is always 1 irrespective of any action specified.

Chapter5

Limitations and Future Work

While the pipeline developed in this work shows promising results, there are several limitations that need to be addressed.

5.0.1 Human Intervention

One significant limitation is the need for human intervention when rendering meshes with a static camera, as discussed in Section One. To achieve optimal results, the camera must be manually positioned. This is particularly important because the mesh can move significantly within the world space, potentially causing it to drift out of the frame. Additionally, manual adjustment is often required to ensure that the mesh correctly aligns with the floor, further complicating the process.

Eliminating the need for user intervention and automating the pose sequence generation in this context is a potential direction for future work. Developing methods to automate camera positioning or adapting the mesh to stay within the frame and properly aligned could significantly enhance the efficiency and usability of the pipeline.

5.0.2 Dependence on 2D Pose

The reliance on 2D pose estimation models, such as OpenPose and DWPose, stems from the fact that most available data on the internet is inherently 2D rather than 3D. While these models are effective for standard video generation tasks, they fall short when it comes to 3D applications. Specifically, 2D poses lack critical information about the human body's shape and depth within 3D space, limiting their utility in more complex 3D scenarios.

Future work could focus on developing models that utilize 3D poses, enabling the creation of 3D camera-parameterized diffusion models for human motion video generation. Such advancements would allow for the direct generation of 3D videos of human motion, overcoming the limitations imposed by the current dependence on 2D pose data.

5.0.3 Mesh Generation and Distortions

The pipeline currently relies on single-view metric depth prediction, which is susceptible to various distortions and artifacts. Because depth prediction depends solely on individual images rather than video sequences, the generated data often exhibits abrupt and abnormal changes between consecutive frames, leading to visible flickering in the depth image video.

To address these issues, future work could explore the use of video-based depth predictions, which may provide more consistent results across frames. Additionally, incorporating SMPL meshes into the depth generation process could further improve depth consistency. This approach not only mitigates the challenges posed by limited data but also produces more reliable and coherent depth information.

5.0.4 Lack of Data for Complete Pipeline

A significant challenge in developing the complete pipeline is the scarcity of comprehensive datasets that include paired text, camera poses, and video data, which are essential for both training and evaluation. Acquiring such data is inherently difficult, and even when available, establishing a reliable correspondence between these elements poses an additional challenge.

One potential solution is to start with available text-video pairs and then infer the camera poses using structure-from-motion (SfM) pipelines. This approach could help bridge the data gap and facilitate the development and evaluation of more robust models.

Chapter6

Conclusion

In this work, we presented a novel pipeline for generating human action videos with minor viewpoint adjustments using a single reference image and textual pose instructions. Our pipeline addresses several limitations of current human motion video synthesis models, particularly their inability to handle textual prompts and generate view-changing videos. By leveraging off-the-shelf tools and integrating depth-based view synthesis techniques, our pipeline offers a more accessible and efficient solution for creating dynamic video content without the need for complex 3D modeling or animation.

Through a series of experiments using the DNA Rendering dataset, we evaluated the performance of our pipeline across different stages/sections. The dataset's availability of reference view video allowed us t evaluate the results of section 2 and section 3. In the absence of ground truth for novel views for section 3, we employed no-reference qualitative metrics, such as BRISQUE, Blur Score, and DOVER, to quantify the visual quality of the synthesized videos. These metrics, along with high-confidence correspondence analysis, demonstrated the effectiveness of our pipeline in producing visually coherent and realistic view-changing videos.

While our results indicate that the proposed methods can generate compelling and realistic human action videos, we also identified areas for improvement. The reliance on single-view depth prediction, for instance, introduces certain challenges due to deformations and inaccuracies. Combining camera control directly in diffusion process and building modules to directly correlate text with image are possible future directions to go for mitigating these issues.

Overall, our pipeline offers a significant step forward in the field of human motion video synthesis, particularly in applications where full 3D view changes are not required. By simplifying the video generation process and enabling the use of text prompts, our approach opens up new possibilities for content creation in areas such as fitness training, virtual try-ons, and social media. Future work could focus on further refining the depth prediction and view synthesis techniques, as well as combining segments presented in pipeline into a single generation task.

Bibliography

[1] Taehoon Kim et al. “Human Motion Aware Text-to-Video Generation With Explicit Camera Control”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 5081–5090.

[2] Yuming Jiang et al. *Text2Performer: Text-Driven Human Video Generation*. 2023. arXiv: `2304.08483 [cs.CV]`. URL: `https://arxiv.org/abs/2304.08483`.

[3] Xiang Wang et al. *VideoComposer: Compositional Video Synthesis with Motion Controllability*. 2023. arXiv: `2306.02018 [cs.CV]`. URL: `https://arxiv.org/abs/2306.02018`.

[4] Arash Naghdi and Payam Adib. *What is 3D Rigging*. URL: `dreamfarmstudios.com/blog/what-is-3d-rigging/`.

[5] Nikos Kolotouros et al. *Instant 3D Human Avatar Generation using Image Diffusion Models*. 2024. arXiv: `2406.07516 [cs.CV]`. URL: `https://arxiv.org/abs/2406.07516`.

[6] Yangyi Huang et al. *TeCH: Text-guided Reconstruction of Lifelike Clothed Humans*. 2023. arXiv: `2308.08545 [cs.CV]`. URL: `https://arxiv.org/abs/2308.08545`.

[7] Mingjin Chen et al. *Ultraman: Single Image 3D Human Reconstruction with Ultra Speed and Detail*. 2024. arXiv: `2403.12028 [cs.CV]`. URL: `https://arxiv.org/abs/2403.12028`.

[8] Wei Jiang et al. *NeuMan: Neural Human Radiance Field from a Single Video*. 2022. arXiv: `2203.12575 [cs.CV]`. URL: `https://arxiv.org/abs/2203.12575`.

[9] Muhammed Kocabas et al. *HUGS: Human Gaussian Splats*. 2023. arXiv: `2311.17910 [cs.CV]`. URL: `https://arxiv.org/abs/2311.17910`.

[10] Liangxiao Hu et al. *GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians*. 2024. arXiv: `2312.02134 [cs.CV]`. URL: `https://arxiv.org/abs/2312.02134`.

[11] Meng-Li Shih et al. *3D Photography using Context-aware Layered Depth Inpainting*. 2020. arXiv: `2004.04727 [cs.CV]`. URL: `https://arxiv.org/abs/2004.04727`.

[12] Richard Tucker and Noah Snavely. *Single-View View Synthesis with Multiplane Images*. 2020. arXiv: `2004.11364 [cs.CV]`. URL: `https://arxiv.org/abs/2004.11364`.

[13] Ruoshi Liu et al. *Zero-1-to-3: Zero-shot One Image to 3D Object*. 2023. arXiv: 2303.11328 [cs.CV]. URL: https://arxiv.org/abs/2303.11328.

[14] Chen-Hsuan Lin et al. *Magic3D: High-Resolution Text-to-3D Content Creation*. 2023. arXiv: 2211.10440 [cs.CV]. URL: https://arxiv.org/abs/2211.10440.

[15] Kyle Sargent et al. *ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Image*. 2024. arXiv: 2310.17994 [cs.CV]. URL: https://arxiv.org/abs/2310.17994.

[16] Yuan Liu et al. *SyncDreamer: Generating Multiview-consistent Images from a Single-view Image*. 2024. arXiv: 2309.03453 [cs.CV]. URL: https://arxiv.org/abs/2309.03453.

[17] Jiawei Ren et al. *DreamGaussian4D: Generative 4D Gaussian Splatting*. 2024. arXiv: 2312.17142 [cs.CV]. URL: https://arxiv.org/abs/2312.17142.

[18] Jiaxiang Tang et al. *DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation*. 2024. arXiv: 2309.16653 [cs.CV]. URL: https://arxiv.org/abs/2309.16653.

[19] Yuanhao Zhai et al. *Language-guided Human Motion Synthesis with Atomic Actions*. 2023. arXiv: 2308.09611 [cs.CV]. URL: https://arxiv.org/abs/2308.09611.

[20] 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. DOI: 10.1145/2816795.2818013.

[21] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis". In: *Chemometrics and Intelligent Laboratory Systems* 2.1 (1987). Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, pp. 37–52. ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(87)80084-9. URL: https://www.sciencedirect.com/science/article/pii/0169743987800849.

[22] Georgios Pavlakos et al. *Expressive Body Capture: 3D Hands, Face, and Body from a Single Image*. 2019. arXiv: 1904.05866 [cs.CV]. URL: https://arxiv.org/abs/1904.05866.

[23] Zhe Cao et al. *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2019. arXiv: 1812.08008 [cs.CV]. URL: https://arxiv.org/abs/1812.08008.

[24] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. *DensePose: Dense Human Pose Estimation In The Wild*. 2018. arXiv: 1802.00434 [cs.CV]. URL: https://arxiv.org/abs/1802.00434.

[25] Zhendong Yang et al. *Effective Whole-body Pose Estimation with Two-stages Distillation*. 2023. arXiv: 2307.15880 [cs.CV]. URL: https://arxiv.org/abs/2307.15880.

[26] MMHuman3D Contributors. *MMHuman3D: OpenMMLab 3D Human Parametric Model Toolbox and Benchmark*. Dec. 2021. URL: https://github.com/open-mmlab/mmhuman3d.

[27] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. *Open3D: A Modern Library for 3D Data Processing*. 2018. arXiv: 1801.09847 [cs.CV]. URL: https://arxiv.org/abs/1801.09847.

[28] Xiang Wang et al. *UniAnimate: Taming Unified Video Diffusion Models for Consistent Human Image Animation*. 2024. arXiv: 2406.01188 [cs.CV]. URL: https://arxiv.org/abs/2406.01188.

[29] Zhongcong Xu et al. *MagicAnimate: Temporally Consistent Human Image Animation using Diffusion Model*. 2023. arXiv: 2311.16498 [cs.CV]. URL: https://arxiv.org/abs/2311.16498.

[30] Di Chang et al. *MagicPose: Realistic Human Poses and Facial Expressions Retargeting with Identity-aware Diffusion*. 2024. arXiv: 2311.12052 [cs.CV]. URL: https://arxiv.org/abs/2311.12052.

[31] Tan Wang et al. *DisCo: Disentangled Control for Realistic Human Dance Generation*. 2024. arXiv: 2307.00040 [cs.CV]. URL: https://arxiv.org/abs/2307.00040.

[32] Lihe Yang et al. *Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data*. 2024. arXiv: 2401.10891 [cs.CV]. URL: https://arxiv.org/abs/2401.10891.

[33] René Ranftl et al. *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer*. 2020. arXiv: 1907.01341 [cs.CV]. URL: https://arxiv.org/abs/1907.01341.

[34] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. "Poisson surface reconstruction". In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, 61–70. ISBN: 3905673363.

[35] F. Bernardini et al. "The ball-pivoting algorithm for surface reconstruction". In: *IEEE Transactions on Visualization and Computer Graphics* 5.4 (1999), pp. 349–359. DOI: 10.1109/2945.817351.

[36] H.R. Hiester et al. "Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange". In: *Ocean Modelling* 73 (Jan. 2014), 30–44. DOI: 10.1016/j.ocemod.2013.10.003.

[37] Shangchen Zhou et al. *ProPainter: Improving Propagation and Transformer for Video Inpainting*. 2023. arXiv: 2309.03897 [cs.CV]. URL: https://arxiv.org/abs/2309.03897.

[38] Anonymous. *Eagan municiple center*. URL: cityofeagan.com/rinks.

[39] Alexander Kirillov et al. “Segment Anything”. In: *arXiv:2304.02643* (2023).

[40] Tianhe Ren et al. *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*. 2024. arXiv: 2401.14159 [cs.CV]. URL: https://arxiv.org/abs/2401.14159.

[41] Chuan Guo et al. “Generating Diverse and Natural 3D Human Motions From Text”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5152–5161.

[42] Wei Cheng et al. *DNA-Rendering: A Diverse Neural Actor Repository for High-Fidelity Human-centric Rendering*. 2023. arXiv: 2307.10173 [cs.CV]. URL: https://arxiv.org/abs/2307.10173.

[43] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG]. URL: https://arxiv.org/abs/1706.08500.

[44] Chuan Guo et al. “Generating Diverse and Natural 3D Human Motions from Text”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5142–5151. DOI: 10.1109/CVPR52688.2022.00509.

[45] Alain Horé and Djemel Ziou. “Image Quality Metrics: PSNR vs. SSIM”. In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.

[46] Thomas Unterthiner et al. *Towards Accurate Generative Models of Video: A New Metric Challenges*. 2019. arXiv: 1812.01717 [cs.CV]. URL: https://arxiv.org/abs/1812.01717.

[47] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. “No-Reference Image Quality Assessment in the Spatial Domain”. In: *IEEE Transactions on Image Processing* 21.12 (2012), pp. 4695–4708. DOI: 10.1109/TIP.2012.2214050.

[48] Haoning Wu et al. *Exploring Video Quality Assessment on User Generated Contents from Aesthetic and Technical Perspectives*. 2023. arXiv: 2211.04894 [cs.CV]. URL: https://arxiv.org/abs/2211.04894.

[49] Jiaming Sun et al. *LoFTR: Detector-Free Local Feature Matching with Transformers*. 2021. arXiv: 2104.00680 [cs.CV]. URL: https://arxiv.org/abs/2104.00680.

[50] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[51] Yiwei Ma et al. *X-CLIP: End-to-End Multi-grained Contrastive Learning for Video-Text Retrieval*. 2022. arXiv: 2207.07285 [cs.CV]. URL: https://arxiv.org/abs/2207.07285.