IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 March 2024, accepted 19 March 2024, date of publication 25 March 2024, date of current version 4 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381497

RESEARCH ARTICLE

3D Clothed Human Body Generation Method Based on Inter-Frame Motion Prediction of 2D Images

SHAOJIANG LIU, ZHIMING XU, ZHIJUN ZHENG, JINTING ZHANG, DANYU LI, AND ZEMIN QIU

Guangzhou Xinhua University, Dongguan 523133, China

Corresponding author: Zemin Qiu (qiuzemin@xhsysu.edu.cn)

This work was supported in part by Guangzhou Science and Technology Project with under Grant 202002030273, in part by the Young Innovative Talents in Guangdong Ordinary Colleges and Universities under Project 2023KQNCX124 and Project 2022KQNCX139, in part by the Faculty Members of Guangzhou Xinhua University under Project 2020KYYB03, in part by Guangdong Key Discipline Scientific Research Capability Improvement under Project 2021ZDJS144 and Project 2022ZDJS151, in part by Guangzhou Xinhua College Key Discipline under Project 2020XZD02, and in part by Guangzhou Science and Technology under Project 202201011731 and Project 202201011759.

ABSTRACT With the rapid progress of computer vision and deep learning techniques, accurately predicting continuous human motions from very few image inputs and generating high-quality 3D human models has become a cutting-edge research direction in this field. Despite the achievements in 2D to 3D conversion techniques, it is still a great challenge to capture coherent movements from limited image frames and generate texture-rich 3D models. In this paper, we propose a 3D clothed human body generation method based on Inter-Frame Motion Prediction (IFMP for short) of 2D images, which is capable of not only predicting a series of coherent human body motions, but also reconstructing a detailed textured 3D human body model from only two image frames. The method automatically focuses on key parts of the image through action coding and uses a conditional generative adversarial network to generate a series of consecutive intermediate frame images. A depth-aware implicit function representation is combined to map a 3D model from the 2D image, and high quality textures of the human body in a clothed state are obtained by texture mapping and model detail enhancement methods. Finally, the experimental results validate the advantages of the IFMP method in image action coherence prediction, as well as verifying the effectiveness of the human 3D model generated by the method in terms of geometric accuracy and texture quality.

INDEX TERMS Deep learning, inter-frame motion prediction, 3D human reconstruction, texture mapping.

I. INTRODUCTION

As the digital world continues to integrate into the real life of human beings, the capture and analysis of human motion has become a key technology in several fields, especially in the industries of virtual reality (VR), film production, video games, medical rehabilitation, and professional sports training. With the continuous advancement of technology, there is a growing demand from researchers and developers for more accurate, natural and smooth human motion capture techniques [1], [2]. Against this background, the technique of predicting continuous human movements and generating 3D models through inter-frame prediction has emerged, which is not only of great significance for academic research, but also lays the foundation for technological advancement and market competitiveness of related industries. For example, the ability to generate complete action sequences through the use of only minimal image inputs will greatly simplify the content creation process, reduce costs, and open up the possibility of high-quality content creation in low-resource environments [3], [4]. In addition, this technological advancement will facilitate the development of a new generation of

The associate editor coordinating the review of this manuscript and approving it for publication was Wanqing Zhao.

interactive applications that will be able to respond in real time to the user's actions and intentions, creating a more personalized and dynamic digital experience. As the technology matures and refines, entirely new application models may emerge in the future, such as virtual representatives in remote work, personalized virtual anchors on social media, and even real-time virtual fashion displays. These applications will further deepen the integration of the digital and real worlds, bringing changes to the way humans live and work.

Currently, research in human movement prediction focuses on how to capture and predict human movements more accurately in the short term. However, long-term, coherent motion prediction remains a challenge, especially under complex and variable environmental conditions [5]. In addition, although 2D motion capture technology is relatively mature, there are still many problems in converting from 2D data to 3D models, such as data incompleteness, noise interference, and inaccurate mapping from 2D to 3D. These problems not only affect the accuracy of 3D models, but also limit their application in a wider range of scenarios [6], [7].

In the field of 3D human model generation, although deep learning methods have made significant progress in enabling conversion from 2D images to 3D models, current techniques still face some significant challenges [8]. Most of the current methods focus on the mapping of a single image to a 3D model and lack a deep understanding of dynamic continuity and temporal coherence. This means that while structurally accurate 3D models can be created from a single 2D image, it is still an open challenge to capture and accurately predict a coherent series of actions from a very small number of frame images - e.g., only two. In addition, while existing 3D reconstruction techniques are capable of generating models that are relatively accurate in terms of shape structure, the fidelity in terms of texture, lighting, and materials still needs to be improved. These factors largely affect the realism of the generated models and limit their usability in a wider range of practical application scenarios.

In light of these challenges, this paper aims to explore a new approach that can accurately predict and generate a series of coherent 3D human action models from just two frame images. Our research not only aims to improve the accuracy of the 3D models in terms of dynamic continuity and temporal consistency, but also aims to enhance the realism of the generated models in terms of texture and detail, thus providing a richer and more realistic user experience for various application scenarios.

II. RELATED WORK

Deep learning has made significant progress in the field of motion prediction for 2D images. For example, some research scholars have dug deeper into the complexity of capturing and analyzing human movements using deep learning techniques [9].Cao et al. proposed a method for long-term human movement prediction using scene background, which fully considers the influence of the environment on human movement and achieves an accurate prediction of future human movements in complex scenes [10].Cui et al. broke away from the traditional static analysis method and introduced a framework capable of capturing dynamic interpersonal relationships and individual behavioral patterns [11]. A research scholar proposed an innovative neural time-series model that leverages the advantages of deep learning in time-series data processing to accurately predict the future movement trajectories and behaviors of individuals [12]. These works demonstrate the potential of utilizing deep learning for movement prediction.

With the rapid development in the field of computer vision, generating 3D models from 2D images has become a hot research topic. For example, the PaMIR method utilizes an implicit representation of parametric model conditioning for image-based human reconstruction [13].NeRF performs view synthesis by representing the scene as a neural radiation field [14].ICON utilizes implicitly clothed human bodies obtained from normals [15], while HumanNeRF implements free-viewpoint rendering of a moving figure from monocular video [16]. In addition, KeypointNeRF promotes image-based volumetric avatars through relative spatial coding of keypoints [17], while PIFu and PIFuHD use pixel-aligned implicit functions to digitize high-resolution clothed human bodies [18], [19], respectively. These studies show that the conversion from 2D images to 3D models requires not only advanced geometric and textural understanding, but also a deep understanding of object motion and morphology.

III. MOTION PREDICTION

Predicting intermediate actions from consecutive 2D images is a key challenge in research on motion capture and human pose estimation. The core goal of action prediction is to generate a series of consecutive intermediate frame images from two consecutive frame action images, as shown in Fig. 1. To achieve this goal, we employ an approach that combines an attention mechanism, deep learning coding, and generative adversarial networks.

A. ACTION CODING

The goal of action coding is to extract meaningful features from two consecutive framed action images and provide useful information for the subsequent generation process. To achieve this goal, we use an approach that combines a bidirectional long and short-term memory network (Bi-LSTM) and a self-attention mechanism.

First, we use Bi-LSTM [20] to capture the temporal dependency between two images. Bi-LSTM consists of two directional LSTMs, one from front to back and the other from back to front, which can capture the contextual information before and after. Given two images with feature sequences $X = \{x_1, x_2$, the forward and backward hidden states of Bi-LSTM are respectively:

$$\overrightarrow{h_t} = \mathrm{LSTM}_{forward}(x_t, \overrightarrow{h_{t-1}}) \quad (1)$$

$$\overleftarrow{h_t} = \mathrm{LSTM}_{backward}(x_t, \overleftarrow{h_{t-1}}) \quad (2)$$

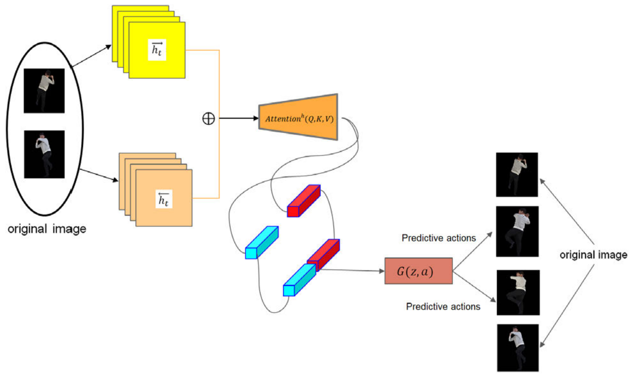


FIGURE 1. Predicting actions from the original two-frame image.

The final hidden state is a concatenation of hidden states in both directions:

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{3}$$

To further enhance the feature representation of these two images, we use the self-attention mechanism. This mechanism automatically focuses on key parts of the image and assigns higher weights to these parts. Given the output H $= \{h_1, h_2\}$ of Bi-LSTM, the output of the self-attention mechanism is:

$$Attention^h(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}} + M)V \tag{4}$$

where Q, K, and V are queries, keys, and values which are obtained from the output H of Bi-LSTM by different linear transformations, $Q = W_q\mathrm{H} + b_q$, $K = W_k\mathrm{H} + b_k$, $V = W_v\mathrm{H} + b_v$, M is a mask matrix used to make sure that only valid positions are focused on, and W and b denote are weight and bias, respectively.

To enhance the expressive power of the model, we introduced Multi-headed Self-attention [21]. In this mechanism, we run multiple self-attention mechanisms in parallel and concatenate their outputs, with the final output being a concatenation of the outputs of all the heads:

$$\begin{aligned} &MultiHeadAttention(Q, K, V) \\ &= Concat(Attention^1, Attention^2, \ldots, Attention^H)W_O \end{aligned} \tag{5}$$

where W_O is an output weight matrix. The pseudo-code for the model to perform action coding is as follows:

Algorithm 1 Action Encoding

```
**Input**: Two consecutive action frame images
**Output**: Enhanced feature representations
**A. Feature Extraction and Temporal Dependency Analysis**
Extract features from two consecutive action frame images.
Capture temporal dependencies between images using Bi-LSTM.
**Given**: Feature sequences $X - \{x_1, x_2\}$
$h_{\overrightarrow{t}} = \mathrm{LSTM}_{\mathrm{forward}}\ (x_t, h_{t-1})$
$h_t^- = \mathrm{LSTM}_{\mathrm{backward}}\ \left(x_t, h_{t-1}^+\right)$
$h_i = h_{\mathrm{t}}^+ \oplus h_t^+$
**B. Feature Enhancement with Self-Attention Mechanism**
Enhance feature representations using self-attention mechanism.
**Given**: $H = \{/\mathrm{ii},/\mathrm{i2}\}$
$Q = W_q\ H + b_q$,$K$= $W_k H + b_k$,$V = W_V H + b_v$
$M$ is a mask matrix for focusing on effective positions.
Attention $^h(Q, K, V) = \mathrm{sothmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right) V$
**C. Multi-Head Attention for Enhanced Model Expressiveness**
Introduce multi-head attention mechanism.
MultiHeadAttention($Q$, $K$, $V$)= Concat(Attention$^1$. Attention$^2$,..., Attention$^H$) Wo
```

B. PREDICTIVE IMAGE GENERATION

In the action encoding phase, we have extracted meaningful features from two consecutive frame action images. Next, we will use these features as conditions to generate a series of consecutive intermediate frame images via a conditional generative adversarial network.

The generator employs a deep residual network (ResNet) structure [22], which has been shown to perform well in image generation tasks. To enhance the detail and quality of the generated images, we also include dense connections in the network. The inputs to the generator are the random noise z and the action representation a obtained from the previous encoding. These two are first combined through a fully connected layer, then passed through multiple residual and densely connected blocks, and finally through a transposed convolutional layer to generate a series of consecutive intermediate frames of an image denoted as:

$$I_{ge} = G(z, a) \tag{6}$$

where G is the network structure of the generator, consisting of multiple deep residual blocks and densely connected blocks.

The discriminator employs a deep convolutional neural network that includes multiple convolutional layers, batch normalization, and a LeakyReLU activation function. To improve the discriminative power of the model, we also include spectral normalization in the discriminator. The inputs to the discriminator are an image and an action representation. These two are first combined through a convolutional layer, then through multiple convolutional layers, and finally through a fully connected layer that outputs a probability indicating the probability that the input image is true.

$$P_{real} = D(I_{real}, a) \tag{7}$$

$$P_{ge} = D(I_{ge}, a) \tag{8}$$

D is the network structure of the discriminator, consisting of multiple convolutional layers, batch normalization and LeakyReLU activation function, and I_{real} denotes the real image. The model employs a loss function designed to ensure that the generated intermediate frame image is as similar as possible to the real intermediate frame image and is consistent with the representation of the action obtained from the encoding. The loss function consists of the generator's loss and the discriminator's loss:

$$L_{ge} = -\log\left(D\left(G\left(z,a\right),a\right)\right) + \lambda \mathrm{MSE}(G\left(z,a\right), I_{target}) \tag{9}$$

$$L_{dis} = -\log\left(D\left(I_{real},a\right)\right) - \log(1 - D\left(G\left(z,a\right),a\right)) \tag{10}$$

where MSE is the mean square error, which is used to ensure that the difference between the generated image and the target image is minimized at the pixel level, and λ is a weighting parameter that balances the two loss terms.

To ensure that the generated images are consistent with the input image at different scales, we can also add multi-scale structures to the generator and discriminator. This allows us to capture and generate details of the image at different scales. By computing the losses at different scales and summing them up, we can ensure that the generated image agrees with the input image at all scales. The pseudo-code for the model to perform action prediction is as follows:

Algorithm 2 Predictive Image Generation

```
1: Input: Random noise $z$, action representation $a$
2: Output: Generated image $I_{\text{gen}}$
3: Generate images using conditional GAN.
4: $I_{gen} = G(z, a)$
5: Use deep convolutional neural network for discriminator.
6: $P_{\text{real}} = D\left(I_{\text{real}}, a\right)$
7: $P_{\text{gen}} = D\left(I_{\text{gen}}, a\right)$
8: Define loss function for image similarity.
9: $L_{\text{gen}} = -\log(D(G(z,a),a)) + \lambda\,\mathrm{MSE}\left(G(z,a), I_{\text{target}}\right)$
10: $L_{\text{dis}} = -\log\left(D\left(I_{\text{real}}, a\right)\right) - \log(1 - D(G(z,a),a))$
```

C. MULTI-TASK LEARNING WITH DEPTH ESTIMATION

In the process of action prediction, in addition to generating consecutive frames of action images, depth information is also a key component that can provide valuable contextual information for subsequent 3D model generation. Therefore, our model employs a multitask learning strategy for both action prediction and depth estimation.

Multi-task learning allows the model to learn multiple related tasks in a single framework, thus enabling knowledge sharing between tasks and improving the model's generalization ability. In our scenario, action prediction and depth estimation are two highly related tasks. Depth information can help the model better understand the motion and positional relationships of objects, while action information can provide context about the shape and structure of objects for depth estimation.

To perform depth estimation, we use an advanced network structure-Dense Depth, which is based on DenseNet [23] and incorporates the features of depth estimation. For image depth estimation, we use the following network structure:

(1) Initial convolutional layer: the input image first passes through an initial convolutional layer to obtain a set of preliminary feature maps.

(2) Dense blocks: these feature maps go into a series of dense blocks. Each dense block contains multiple convolutional layers, and the output of each convolutional layer is connected to the outputs of all previous layers of the block, ensuring that each layer has access to information from all previous layers.

(3) Transition layer: in order to reduce the resolution of the feature map, we add a transition layer between every two dense blocks containing a convolutional layer and an average pooling layer.

(4) Global average pooling: after all dense blocks, we use global average pooling to get a fixed size feature vector.

(5) Regression layer: finally the feature vector is converted to a depth map by a fully connected layer.

Given an image I, the output of the depth estimation network is:

$$D\left(I\right) = DepthNet(I) \tag{11}$$

where $DepthNet(\cdot)$ denotes the aforementioned DenseNet-based depth estimation network structure.

In order to learn action prediction and depth estimation simultaneously, we design a multi-task loss function. This loss function is a weighted sum of the action prediction loss and the depth estimation loss:

$$L_{total} = \alpha L_{action} + \beta L_{depth} \tag{12}$$

where L_{action} is the loss of the action prediction, which consists of the generator's loss L_{ge} and the discriminator's loss L_{dis}. α and β are two weight parameters used to balance the two loss terms. L_{depth} is the loss of the depth estimation network, expressed as the mean square error between the predicted depth map and the true depth map:

$$L_{depth} = \mathrm{MSE}(DepthNet\left(I\right), I_{depth\_true}) \tag{13}$$

where I_{depth_true} is the true depth map. With this multi-task learning strategy, our model can perform both action prediction and depth estimation, thus improving the overall performance of the model.

The pseudo-code for the model to perform multi-task learning and depth estimation is shown below.

Algorithm 3 Multi-task Learning and Depth Estimation

```
1: Input: An image $I$
2: Outputs: Action prediction, Depth estimation
3: Employ multi-task learning for action prediction and depth estimation.
4: Use Dense Depth network for depth estimation.
5: $D(I) = \text{DepthNet}(I)$
6: Design a multi-task loss function.
7: $L_{\text{total}} = \alpha L_{\text{action}} + \beta L_{\text{depth}}$
8: $L_{\text{depth}} = \mathrm{MSE}\left(\text{DepthNet}(I), I_{\text{depth\_true}}\right)$
```

IV. 2D IMAGE TO 3D MODEL CONVERSION

In this chapter, we take an in-depth look at the conversion process from 2D images to 3D models, especially after motion prediction has been completed. In order to ensure high

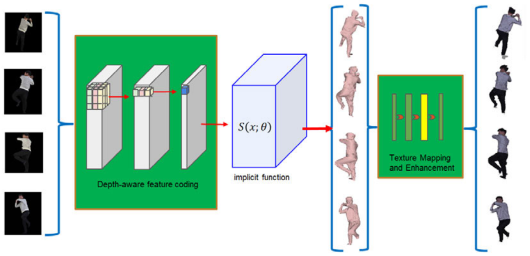


FIGURE 2. Convert predicted 2D images to 3D models.

resolution of the 3D model as well as texture mapping and optimization, we introduce innovative technical strategies. First, by using implicit functions and depth-aware modules, we are able to obtain 3Dmesh surfaces from 2D images. Second, in the texture mapping stage, we employ a texture fusion method that ensures a high degree of texture consistency and accuracy under various lighting and viewing angle conditions. Finally, to further enhance the details of the model, we improve the quality of the texture by generating an adversarial network. The process is shown in Fig. 2.

A. COMBINING DEPTH-AWARE IMPLICIT FUNCTION REPRESENTATIONS

The implicit function representation provides a continuous, mapping from 2D images to 3D models for 3D reconstruction. The 3D model is represented in PIFuHD using the implicit function $S(x;\theta)$, where x is a point in 3D space and θ is a parameter of the network. The output of this function is a scalar value indicating whether point x is occupied by the 3D model or not. Specifically, if $S(x;\theta) > 0$, the point is occupied by the model; otherwise, point x is outside the model.

Traditional implicit function representations typically use fixed-resolution images, which limits the model's ability to capture high-frequency details, especially in texture-rich or complexly shaped regions. Our approach introduces adaptive resolution extraction. The system automatically adjusts the resolution of the input image based on the content and complexity of the image, ensuring higher resolution in detail-rich or shape-complex regions, thus capturing details more accurately. This process can be achieved by the following optimized objective function:

$$min_{\theta}L_{res} = \sum\nolimits_{x\in X} \left\| I_{high}(x) - I_{gen}(x;\theta) \right\|^2 \tag{14}$$

where I_{high} is the high resolution image, I_{gen} is the generated image, X is all the pixel points, θ is the network parameter and L_{res} is the resolution loss.

We introduce a depth-aware module that utilizes depth information obtained from a multi-task learning framework to enhance the 3D perception of 2D features. This way of feature encoding incorporating depth information helps the implicit function to more accurately infer the occupation state of points in 3D space. The depth-aware feature encoding can be described by the following function:

$$E(I, D(I);\theta_E) = f_{enc}(I;\theta_{enc}) \oplus g_{depth}(D(I);\theta_{depth}) \tag{15}$$

where f_{enc} and g_{depth} are the coding functions for image and depth information, respectively, $D(I)$ denotes the output of the depth estimation network, $\oplus$ denotes the fusion of features, and θ_{enc} and θ_{depth} are the corresponding network parameters.

In standard implicit function representations, each 3D point is treated independently without considering point-to-point relationships, which ignores the importance of the distribution and interaction of points in 3D space for shape understanding. We introduce the Point Transformer layer, which captures and understands the complex relationships between 3D points. This layer dynamically adjusts the feature representation of each point by considering the state of each point and its neighborhood. The Point Transformer is implemented by the following function:

$$T(x;\theta_T) = \mathrm{Transformer}(E(x), E(N(x));\theta_T) \tag{16}$$

where $T(x)$ is the transformed feature of point x, $E(x)$ is the original feature of point x, and $N(x)$ denotes the set of points in x's neighborhood, θ_T denotes the network parameters of the point Transformer layer, including weights and biases.

Finally, we combine the feature $E(I, D(I);\theta_E)$ with depth information and the processed 3D point $T(x;\theta_T)$ into the implicit function $S(x;\theta)$. The implicit function will output a scalar value representing the occupation state of the point x in 3D space:

$$S(x;\theta) = \sigma(E(I, D(I);\theta_E), T(x;\theta_T);\theta) \tag{17}$$

In this way, the implicit function is enabled to accurately represent the 3D model, making it possible to utilize not only the visual and depth information of the 2D image, but also to understand the contextual relationships of the points in the 3D space, thus enabling an accurate mapping from the 2D image to the 3D model.

B. TEXTURE MAPPING

Texture mapping is a critical step in 3D model creation that involves accurately mapping the texture information of a 2D image onto the surface of a 3D model. This process requires not only geometric accuracy but also a high degree of consistency with the original image in terms of color, texture, and lighting. In this section, we employ an innovative texture mapping and optimization framework that focuses on handling high-resolution textures and is able to adapt to lighting and viewing angle variations.

We use the ResNeSt network [24] to extract texture features from 2D images, and ResNeSt is able to capture richer contextual information and fine-grained features by introducing segmented transforms and attention mechanisms:

$$T_f = \mathrm{ResNeSt}(I;\theta_{texture}) \tag{18}$$

where T_f is the texture feature extracted from the 2D image I, $\mathrm{ResNeSt}(\cdot)$ is the feature extraction network, and $\theta_{texture}$ is the network parameters.

We introduce a Lighting-viewing consistency module (LVCM for short) that handles both lighting variations and viewing angle variations to maintain texture consistency under different conditions. Based on the estimated lighting and viewing angle parameters, the LVCM adjusts the texture of the 3D model to ensure visual consistency under different conditions. This involves adjusting attributes such as color, brightness, and contrast of the texture to match the predicted lighting and viewing angle conditions. The loss function for texture consistency is defined as follows:

$$L_{texture} = \left\| T_{original} - T_{adjusted}(P_{light}, P_{view}) \right\|^2 \tag{19}$$

$T_{original}$ is the original texture, $T_{adjusted}$ is the texture adjusted according to the lighting and perspective, P_{view} is the real perspective parameter, P_{light} is the real lighting parameter.

The LVCM estimates the lighting conditions in the current view, which is achieved by analyzing the input image and predicting the lighting parameters such as the light source direction, intensity, and ambient light conditions. The LVCM also evaluates the viewing angle conditions of the current view, including the position and orientation of the camera. The loss function for illumination estimation and the loss function for viewpoint estimation are defined as follows, respectively:

$$L_{light} = \left\| P_{light} - \hat{P}_{light} \right\|^2 \tag{20}$$

$$L_{view} = \left\| P_{view} - \hat{P}_{view} \right\|^2 \tag{21}$$

$\hat{P}_{light}, \hat{P}_{view}$ denote the predicted light parameter and the predicted view parameter, respectively. The ultimate goal of LVCM is to minimize the weighted sum of all loss functions:

$$L_{LVCM} = w_1 L_{light} + w_2 L_{view} + w_3 L_{texture} \tag{22}$$

C. MODEL DETAIL ENHANCEMENT

After completing the texture mapping, we further enhance the visual quality of the 3D model, especially in terms of texture details. To achieve finer texture details, we propose a deep learning-based detail enhancement method that combines the advantages of Asymptotic Feature Pyramid Network (AFPN) [25] and Generative Adversarial Network (GAN) to generate high-resolution, high-quality textures.

After the texture mapping is complete, we further extract and enhance the texture features obtained from the original 2D image using an AFPN network, which enhances the important texture information by adaptively re-weighting the feature maps at different levels while suppressing irrelevant details. This can be achieved by the following optimization objective:

$$\min_{\theta_f} L_{AFPN} = -\sum_{i=1}^{N} \lambda_i log P(Y_i | X; \ \theta_f) \tag{23}$$

where L_{AFPN} denotes the loss function of the AFPN network, P is the prediction of the network, Y_i is the real texture label of the i-th scale, X is the input low-resolution texture, θ_f is the parameter of the AFPN network, and λ_i is the adaptive weight.

Based on the AFPN network, we further introduce a texture refinement generative adversarial network to refine and enhance the texture details of 3D models. The network uses a generator and a discriminator to enhance the texture details. The generator uses the output of the AFPN network and further refines the texture. The discriminator then evaluates the realism of the generated texture. The objective function of the network consists of two parts: one part evaluates the realism of the generated texture and the other part penalizes the Euclidean distance between the generated texture and the real texture, which can be expressed as:

$$\begin{aligned}\min_{\theta_g} \max_{\theta_d} L_{Gan} = \mathbb{E}_{x\sim p_{data}(x)} \left[logD\left(x; \theta_d\right)\right] \\ + \mathbb{E}_{z\sim p_z(z)} \left[\log\left(1 - D(G\left(z; \theta_g\right); \theta_d)\right)\right] \\ + \mu \left\| G\left(z; \theta_g\right) - x \right\|_2^2\end{aligned} \tag{24}$$

where x denotes the original texture data extracted from the surface of the 3D model, z is the random noise, and μ is the weight of the Euclidean distance penalty. By this method, we further enhance the texture details of the 3D model to achieve a higher degree of visual fidelity and aesthetics.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASETS AND EVALUATION METRICS

Self-filmed video sequences: To validate the accuracy of our IFMP method in predicting human movements, we used self-filmed 30-minute video sequences that contain about 54,000 frames. These videos captured diverse human movements under a variety of lighting and background conditions.

THuman2.0 dataset [26]: In order to validate the accuracy of our IFMP method in converting 2D images to 3D models, we used the widely recognized THuman2.0 dataset. This dataset contains human images and corresponding high-accuracy 3D models under a wide range of poses, clothing and lighting conditions.

For 2D action prediction, we used the following key metrics: (1) Prediction accuracy. Evaluated by comparing the Euclidean distance between the predicted keypoint locations and the true keypoint locations. (2) Action coherence. The smoothness of the action is assessed by analyzing the frame-to-frame differences in the predicted action sequences, which can be quantified by calculating the rate of change of the motion velocity of the keypoints.

For 3D model generation, we used the following metrics: (1) Geometric accuracy. This is evaluated by calculating the point cloud distance between the generated 3D model and the real model. (2) Texture quality. The structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) were used to quantitatively assess the similarity between the generated texture and the real texture.

B. IMPLEMENTATION DETAILS

Our IFMP model is implemented based on PyTorch. For action prediction, we use the Adam optimizer on selfie video sequences with an initial learning rate set to 1e-4 and a

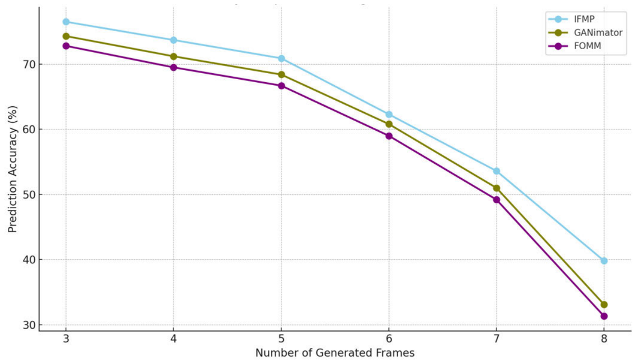


FIGURE 3. Predicting the accuracy of key points.

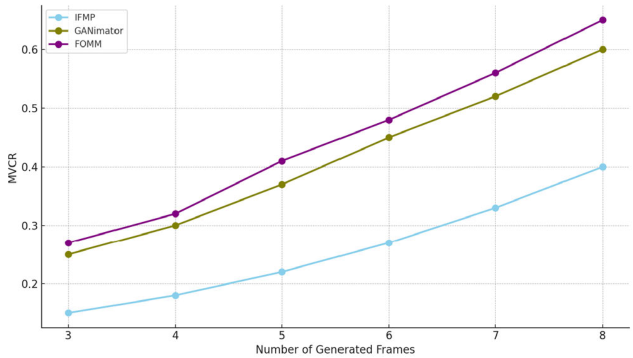


FIGURE 4. MVCR comparison results.

learning rate decay strategy. The model is trained on two NVIDIA GTX 3080 Ti GPUs for a total of 60 epochs. data enhancement includes random cropping, level flipping, and normalization. For 2D to 3D conversion, we trained and validated on the THuman 2.0 dataset. We employed multi-view geometry and depth estimation, where the depth estimation module uses a fully connected layer with 512 neurons. For the texture mapping and detail enhancement phases, we used a deep learning based approach combining Asymptotic Feature Pyramid Network and Generative Adversarial Network.

In order to validate the state-of-the-art of the methods in this paper, for 2D action prediction, the comparison algorithms we used are the GANimator [27] method, and the method from the literature [28] (hereinafter referred to as FOMM), which can create animations from a single 2D image. For the comparison of 3D model generation, the comparison methods we used were PIFu, PIFuHD, and PaMIR. To ensure a fair comparison, we evaluated all methods on the same experimental platform, dataset, and evaluation metrics.

C. COMPARISON AND ANALYSIS

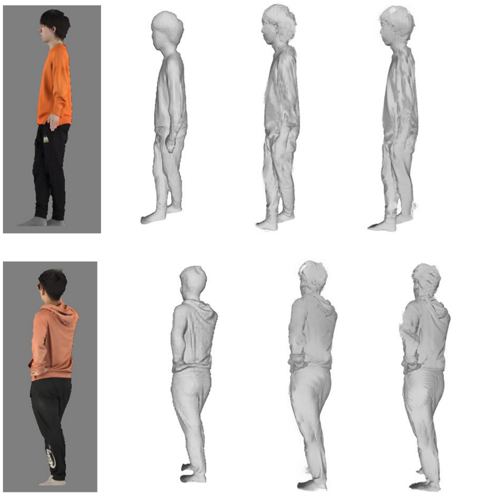
In order to validate the performance of our IFMP method for 2D action prediction, we used the accuracy of predicted keypoint locations as an evaluation metric. This is achieved by calculating the Euclidean distance between the predicted keypoints and the real keypoints. Our experiments were performed on a 30-minute video sequence shot by ourselves. We compared IFMP with state-of-the-art GANimator and FOMM methods. The experiments were performed given two initial frame images, with the aim of generating three to eight frame images with consecutive actions. The Euclidean distance between all the keypoints of each generated image and the true keypoint is calculated, and then a distance greater than 0.2 mm is judged to be a non-conformity of the keypoint with the true keypoint. Then the average accuracy of all the keypoints of all the generated images is counted and the experimental results in Fig. 3 are obtained.

As can be seen from the above figure, IFMP slightly outperforms the GANimator and FOMM methods in all cases. It is worth noting that the accuracy of all methods shows a decreasing trend as the number of frames to be predicted increases. However, the performance degradation of IFMP is smaller relative to both GANimator and FOMM, suggesting that our method is more robust when dealing with longer sequences. Specifically, when generating 8 images from 2 initial frame images, the accuracy of IFMP decreases by 36.7%, while the accuracy of GANimator decreases by 41.2% and that of FOMM decreases by 40.4%. One possible reason for this difference in performance is that IFMP employs a more efficient time-continuum modeling approach that is better able to capture and predict actions in future frames. In addition, IFMP employs an improved loss function and finer feature extraction for keypoint detection and action generation, which further improves the prediction accuracy.

In addition to prediction accuracy, action coherence is also a key metric for evaluating action prediction algorithms. To quantify action fluency, we calculated the rate of change of motion velocity at each key point in the predicted action sequence. Ideally, a coherent action sequence should have a small rate of change of velocity because in the real world, human actions are usually smooth and coherent. We use a standardized metric called Mean Velocity Change Rate (MVCR).MVCR is obtained by calculating the velocity change at each key point in the sequence and averaging it. A lower MVCR indicates better action coherence. The following is a comparison of the MVCR of the IFMP method with GANimator and FOMM for different sequence lengths:

From the results, it can be seen that both in shorter and longer sequences, the IFMP method exhibits a lower MVCR compared to both methods, GANimator and FOMM, which means that our method generates smoother and more coherent action sequences. Especially in longer sequences (e.g., 8 frames), the MVCR of IFMP is only 0.40, while the MVCR of GANimator reaches 0.60 and the MVCR of FOMM reaches 0.65, which indicates that the latter two have significantly lower performance in terms of action coherence.

In terms of verifying the effect of 3D conversion of 2D images, we tested the IFMP method proposed in this paper with PIFuHD and PaMIR methods for 3D image generation using the same 2D images under the same training dataset and simulation environment, and obtained the results in Fig. 5. As can be seen from the following experimental results, PaMIR and IFMP perform better in 3D model generation compared to PIFuHD, especially in model details, capturing



(a) 2D Image (b) PIFuHD (c) PaMIR (d) IFMP

FIGURE 5. Experimental results of PIFuHD, PaMIR and IFMP methods for 3D model generation.

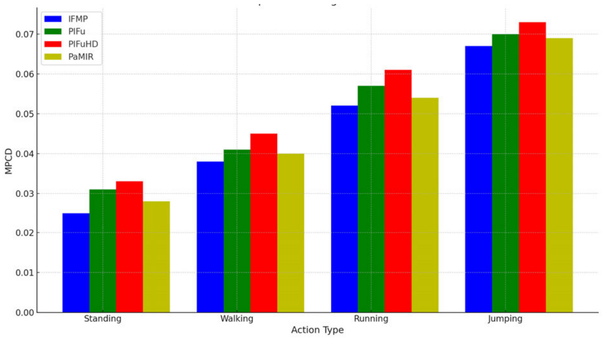


FIGURE 6. MPCD results for different action types.

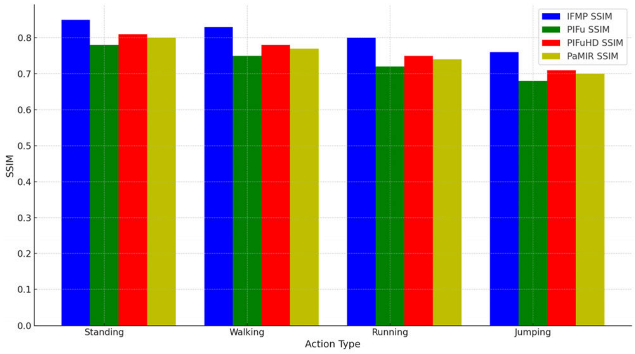


FIGURE 7. SSIM results for different action types.

the fine details of the model more accurately, including surface fine textures and small geometric changes.

In order to comprehensively evaluate the performance of our IFMP method under different conditions, we chose different movement types to test the geometric accuracy of the model. These movement types include standing, walking, running, and jumping, each with its own unique challenges such as different body dynamics and pose changes. We calculated the Mean Point Cloud Distance (MPCD) between IFMP and PIFu, PIFuHD, and PaMIR for these specific movement types. Figure 6 below shows the results of the experiment:

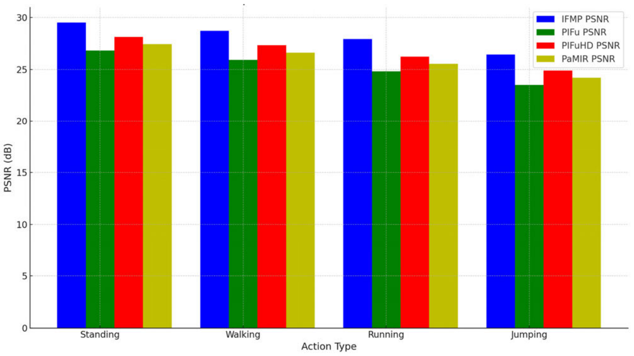


FIGURE 8. PSNR results for different action types.

As can be seen from the above figure, different movement types have a significant effect on the geometric accuracy. In the relatively static ‘‘standing’’ action, all methods show high accuracy, with IFMP reaching the lowest MPCD value of 0.025, demonstrating its accuracy in capturing static poses. However, the geometric accuracy starts to decrease with the addition of movements such as ‘‘walking’’ and ‘‘running’’ because these movements introduce more dynamics and uncertainty, which increases the difficulty of 3D reconstruction. In more dynamic and complex actions, such as ‘‘jumping’’, the MPCD values of all methods increase, but the IFMP still maintains a relatively low value. This shows the effectiveness of IFMP in dealing with complex dynamic scenes.

Texture quality is another important metric for the realism of 3D mannequins. In order to quantitatively evaluate the performance of our IFMP method in texture generation, we use the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) as evaluation metrics. SSIM is used to measure the visual similarity between two images, whereas PSNR is a commonly used metric to characterize the quality of an image reconstruction, especially when detail and texture preservation of an image are taken into account. Figures 7 and 8 show the experimental results.

In the reconstruction of the jumping action, the IFMP reached 0.76 on SSIM, showing its ability to maintain the image structure. This is mainly due to the fact that IFMP synthesizes information from multiple perspectives and is able to effectively deal with lighting and viewpoint changes. However, despite IFMP’s advantage in texture quality, the gap with other methods is not significant. This suggests that existing 3D human reconstruction methods have been able to achieve high texture reconstruction quality. Future work can explore more texture enhancement techniques and data fusion methods to further improve the visual realism of 3D models. It is worth noting that PSNR values are usually associated with background noise and detail loss in images. Although IFMP obtained 26.4 dB in PSNR in jumping action, this score still has room for improvement when dealing with complex backgrounds and different lighting conditions. These factors may affect the quality of texture reconstruction, so future research needs to further explore noise suppression and detail enhancement techniques to improve texture quality.

VI. CONCLUSION

This research is dedicated to predicting intermediate actions from consecutive 2D images and generating them into 3D models. We propose an integrated framework that combines an attention mechanism, deep learning coding, and generative adversarial networks to generate a series of consecutive intermediate frame images, and explore the transformation from 2D images to 3D models, especially high-resolution 3D model reconstruction and texture mapping after action prediction is complete. For action coding, we employ a bi-directional long and short-term memory network and a self-attention mechanism to capture temporal dependencies between images and automatically focus on critical parts of the image. In addition, we introduce Multi-headed Self-attention to enhance the expressive power of the model. For predictive image generation, we use a conditional generative adversarial network to generate a series of consecutive intermediate frame images. The generator employs a deep residual network structure and incorporates dense connections to enhance the detail and quality of the generated images. The discriminator, on the other hand, employs a deep convolutional neural network and incorporates spectral normalization to improve the discriminative power of the model. For 2D image to 3D model conversion, we introduce innovative technical strategies such as incorporating depth-aware implicit function representation, texture mapping and model detail enhancement. We propose a deep learning-based detail enhancement method that combines Asymptotic Feature Pyramid Network and Generative Adversarial Network to generate high-resolution, high-quality textures. Future work will focus on further improving the accuracy and real-time performance of the model, as well as exploring more application scenarios.

REFERENCES

[1] J. Xie, Y. Xu, Z. Zheng, S.-C. Zhu, and Y. N. Wu, "Generative PointNet: Deep energy-based learning on unordered point sets for 3D generation, reconstruction and classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14971–14980.

[2] X. Zheng, Y. Liu, P. Wang, and X. Tong, "SDF-StyleGAN: Implicit SDF-based StyleGAN for 3D shape generation," *Comput. Graph. Forum*, vol. 41, no. 5, pp. 52–63, Aug. 2022.

[3] F. Hong, Z. Chen, Y. Lan, L. Pan, and Z. Liu, "EVA3D: Compositional 3D human generation from 2D image collections," 2022, *arXiv:2210.04888*.

[4] J. Gu, L. Liu, P. Wang, and C. Theobalt, "StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis," 2021, *arXiv:2110.08985*.

[5] L. Zhou, Y. Du, and J. Wu, "3D shape generation and completion through point-voxel diffusion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5806–5815.

[6] Y. Deng, J. Yang, D. Chen, F. Wen, and X. Tong, "Disentangled and controllable face image generation via 3D imitative-contrastive learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5153–5162.

[7] B. Li, Y. Zhang, B. Zhao, and H. Shao, "3D-ReConstnet: A single-view 3D-object point cloud reconstruction network," *IEEE Access*, vol. 8, pp. 83782–83790, 2020.

[8] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich, "Score Jacobian chaining: Lifting pretrained 2D diffusion models for 3D generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12619–12629.

[9] J. Bütepage, M. J. Black, D. Kragic, and H. Kjellström, "Deep representation learning for human motion prediction and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1591–1599.

[10] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik, "Long-term human motion prediction with scene context," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 387–404.

[11] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3D human motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6518–6526.

[12] A. Gopalakrishnan, A. Mali, D. Kifer, L. Giles, and A. G. Ororbia, "A neural temporal model for human motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12108–12117.

[13] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, "PaMIR: Parametric model-conditioned implicit representation for image-based human reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3170–3184, Jun. 2022.

[14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Jan. 2022.

[15] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, "ICON: Implicit clothed humans obtained from normals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13286–13296.

[16] C. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "HumanNeRF: Free-viewpoint rendering of moving people from monocular video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16189–16199.

[17] M. Mihajlovic, A. Bansal, M. Zollhoefer, S. Tang, and S. Saito, "KeypointNeRF: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 179–197.

[18] S. Saito, Z. Huang, R. Natsume, S. Morishima, H. Li, and A. Kanazawa, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2304–2314.

[19] S. Saito, T. Simon, J. Saragih, and H. Joo, "PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 81–90.

[20] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Bi-LSTM network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors J.*, vol. 20, no. 3, pp. 1191–1201, Feb. 2020.

[21] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9638–9644.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[24] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-attention networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 2736–2746.

[25] G. Yang, J. Lei, Z. Zhu, S. Cheng, Z. Feng, and R. Liang, "AFPN: Asymptotic feature pyramid network for object detection," 2023, *arXiv:2306.15988*.

[26] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, and Y. Liu, "Function4D: Real-time human volumetric capture from very sparse consumer RGBD sensors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5742–5752.

[27] P. Li, K. Aberman, Z. Zhang, R. Hanocka, and O. Sorkine-Hornung, "GANimator: Neural motion synthesis from a single sequence," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–12, Jul. 2022.

[28] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 7137–7147.