Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9801276, 1 page
https://doi.org/10.1155/2023/9801276

Retraction

Retracted: Convolution-Based Design for Real-Time Pose Recognition and Character Animation Generation

Wireless Communications and Mobile Computing

Received 11 July 2023; Accepted 11 July 2023; Published 12 July 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] D. Wang and J. Lee, "Convolution-Based Design for Real-Time Pose Recognition and Character Animation Generation," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 6572420, 8 pages, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6572420, 8 pages
https://doi.org/10.1155/2022/6572420

Research Article

Convolution-Based Design for Real-Time Pose Recognition and Character Animation Generation

Dan Wang and Jonghan Lee

School of Department of Formative Convergence Arts, General Graduate Hoseo University, Asan 31499, Republic of Korea

Correspondence should be addressed to Dan Wang; 20215530@365.hoseo.edu

Received 13 January 2022; Revised 10 February 2022; Accepted 23 February 2022; Published 18 March 2022

Academic Editor: Zhiguo Qu

Human pose recognition and its generation are an important animation design key point. To this end, this paper designs new neural network structures for 2D and 3D pose extraction tasks and corresponding GPU-oriented acceleration schemes. The scheme first takes an image as input, extracts the human pose from it, converts it into an abstract pose data structure, and then uses the converted dataset as a basis to generate the desired character animation based on the input at runtime. The scheme in this paper has been tested on pose recognition datasets and different levels of hardware showing that 2D pose recognition can reach speeds above 60 fps on common computer hardware, 3D pose recognition can be estimated to reach speeds above 24 fps with an average error of only 110 mm, and real-time animation generation can reach speeds above 30 frames per second.

1. Introduction

Researchers in the field of artificial intelligence have realized that the simulation of the way human neurons work is one of the pathways to artificial intelligence through the analysis of human brain behavior. In the field of machine vision, the extraction and discovery of the visual characteristics of the human pose itself have been one of the popular areas of research [1–3]. The human's own gesture characteristics are the information that needs to be acquired in many fields. For example, monitoring systems want to be able to determine whether a person has fallen or not based on the posture, digital entertainment applications want to draw the corresponding screen output based on the human posture, and virtual reality applications want to capture the human posture and generate the same action of the virtual human in the virtual world [4]. Therefore using machine vision-related techniques to extract, compress, and apply the human pose itself is a topic that many researchers have been digging into. In this area, deep convolutional neural networks show great potential, so the introduction of related techniques to solve many complex problems in this process is in order [5, 6].

The process of extracting, analyzing, storing, and applying human pose is often split into several independent steps to be performed. A typical step can be divided into three steps: 2D pose extraction, 3D pose estimation, and real-time pose animation generation [7]. Among them, 2D pose extraction is the basis of the whole system, 3D pose is the most commonly used output, and real-time human animation generation is the subsequent step after 3D pose data is obtained by many applications [8].

Research on human pose extraction has started long ago. Many studios have adopted solutions based on visual marker points or sensors. Actors are dressed in special costumes with visual marker points or sensors, and then, special systems are used to capture the position of the body's articulation points to achieve pose capture of the human body. The visual marker-based solutions often have a large number of cameras in the venue, each of which estimates the 2D pose within the current camera frame based on the visual marker points; the estimates from multiple cameras are then

combined to produce 3D pose data [9, 10]. The 3D pose data is then used to generate the corresponding virtual character movements in the movie. However, the solution based on visual marker points and sensors requires special costumes, and such a technology requires long lead times and special requirements for the location. For homes and public places, such technology is difficult to be implemented [11].

Therefore, there have been attempts by researchers to accomplish pose estimation of the human body using a single common RGB camera. Traditional solutions are often based on artificially designed rules and features, such as skin color features. Such features are often heavily constrained by scene characteristics such as lighting. Therefore, similar techniques are often used for simple tracking. An example is hand tracking [12]. With the development of deep learning techniques, it has become possible to use neural networks to "learn" target features. Therefore, many researchers have also started to investigate the use of deep convolutional neural networks to extract features for human pose estimation. In recent times, several technical solutions have achieved good performance [13]. A typical example is the OpenPose scheme based on deep convolutional neural networks developed by CMU.

However, the current schemes do not fully satisfy the needs of the applications. One of the reasons is that there are still many limitations in the real-time performance of these solutions. For example, the hardware of OpenPose that can achieve real-time is limited to the current high-end GPUs, and it cannot achieve the real-time requirement on the low-end GPUs [14]. Many 3D pose estimation techniques that claim to be real-time rely on high-quality 2D pose input, which takes time to obtain. The second reason is that there is no complete solution from 2D pose extraction, 3D pose, and estimation to real-time character animation generation. And many film and TV, game, and virtual reality companies need a complete solution more than anything else.

2. Related Work

Real-time pose recognition and animation generation are an important ongoing research direction in computer graphics, and the widely used method is deep learning. Real-time pose recognition and animation generation using deep learning are still a challenging task, and domestic and international research on this topic consists of the following three main areas.

2.1. 2D Posture Recognition. Currently, a typical scheme in the field of 2D pose recognition based on deep learning is the one based on Mask-RCNN [15]. In contrast to the two-stage scheme, all the limbs in the frame are extracted directly. Subsequently, all interconnected limbs are acquired to directly generate pose recognition results for all people, and such a scheme is called one-stage scheme. A typical implementation of this research direction is CMU's OpenPose [16], where OpenPose first selects VGG [17] to build a feature extraction network for backbone and then uses multiple iteratively corrected refine networks to achieve the final result extraction. Of course, there are special series of algorithms in addition to this. For example, DensePose [18] represents a recognition algorithm with limb region recognition as the core. This type of algorithm does not only identify the type of a "point" but also marks the whole range of the limb area. These solutions are more complex in their tasks, and therefore, it is difficult to achieve real-time.

2.2. 3D Posture Recognition. The idea in the field of 3D pose recognition is to build on the 2D pose recognition and further estimate the 3D pose. This scheme mainly relies on the high accuracy 2D pose recognition results. For example, [19] proposed a feedforward neural network-based scheme, which directly estimates the corresponding 3D pose using a neural network based on the already extracted 2D pose. Based on a similar scheme, Facebook AI introduces the information of time series to further improve the accuracy of 3D pose. The other one is to estimate 3D pose directly using images as input. For example, the scheme proposed in [3] prepares a parametric 3D human model and then constructs an encoder-decoder network directly based on the given image and then uses this network to predict each parameter of the parametric model to achieve the pose prediction of the human body. The scheme is able to predict both human pose and human body size.

2.3. Real-Time Character Animation Generation. Regarding the scheme of real-time character animation generation using feedforward neural networks based on a preextracted 3D pose database, the main research idea is to take the current environmental features near the character and the character's behavioral orientation (e.g., turning and jumping) data as input and train the neural network with matching pose as the desired output; then, the nearby virtual environmental features are continuously extracted and input to the neural network at runtime. Then, we extract the nearby virtual environment features into the neural network at runtime to build a highly realistic character animation. Among them, Phase-Functioned Neural Networks [8] further improve the realism of character animation by introducing forced periodicity, and this scheme is a representative research result in this direction.

3. Multitask Multilayer Neural Network for Real-Time 3D Pose Estimation

In this section, we first discuss the representation of the pose in the system that determines the neural network and postprocessing design.

Then, we discuss each step in the pipeline. A multitasking neural network is used to output 2D pose information and 3D information simultaneously. And a multistage detection structure is used to maintain speed and accuracy. The second postprocessing step involves detecting, linking, and automatically matching the pose in 2D image space with the depth in 3D world space. In contrast to the typical 3D pose estimation system simple 3D pose baseline [6], our work uses a specially designed intermediate representation to avoid losing depth information. This design allows our

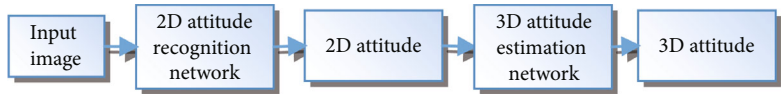

Figure 1: Typical 3D pose estimation algorithm flow.

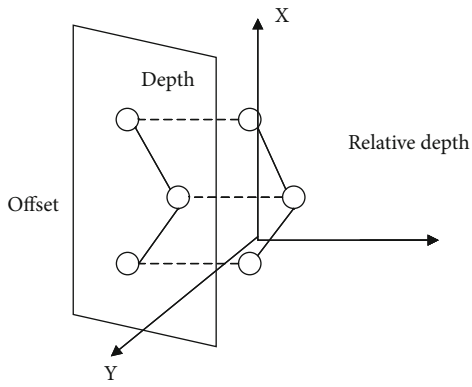

Figure 2: Encoding of a 3D off-node J.

system to perform both 2D and 3D detections in a single network, which means that we are able to achieve higher speedups and the neural network is better able to convey potentially useful information.

A typical 3D pose estimation system tends to go through the following process as shown in Figure 1.

The input image is first processed by a 2D pose recognition network to produce a 2D pose. The 2D pose is then passed to the 3D pose estimation network to be processed one by one to produce a 3D pose for each person. This process involves both 2D and 3D networks, and the second 3D pose estimation network will run more than once. This leads to the following problems.

(1) The 2D pose network converts the input image into an abstract 2D pose, a process that results in the loss of the image information carried in the original image. Therefore, it is not possible to improve the accuracy of the 3D pose estimation network based on information such as pixel values

(2) In the current case, there are two networks to run, and in a practical deployment, it is often the case that the first network is started first. The first network, after running to get the output results, offloads the first network and then starts the second network. This results in a significant performance loss and limits the possibility of performance improvement

In summary, traditional 3D pose estimation systems have bottlenecks in terms of further accuracy and operational speedup, so we propose our own 3D pose estimation network scheme. Instead of having two steps, the scheme outputs all the information needed to assemble the 3D pose in a single run.

This section first describes how to design a 3D pose representation that is more suitable for the output of the neural network, followed by a discussion of the specific construction of the network.

3.1. 3D Pose Representation Based on Relative Depth. The pose representation is like a bridge between the neural network and the postprocessing system. Therefore, the definition of the representation has a great impact on both neural network design and postprocessing system design. We use CNN as the basic module block, which means that the output is an image. Therefore, we define an image-based representation for all human poses. Continuing from the 2D pose recognition system, we consider the human pose as a directed Figure 2. In this directed graph, each node in this directed graph corresponds to a human node j. Each node contains the following information: $(\text{Class}_j, X_j, Y_j, D_j, \text{Offset}\, X_j, \text{Offset}\, Y_j)$. This information is assembled into a 3D pose in the manner of Figure 2.

Class_j denotes the type of the current node. X_j, Y_j denotes the 2D coordinates of the current node. D_j denotes the distance from the current node to the camera. $\text{Offset}\, X_j, \text{Offset}\, Y_j$ denotes the relative offset from the parent node of the current node to the current node. This encoding continues our research in 2D pose recognition systems.

In order to simplify and speed up the process, we want to make the network task as simple as possible. Therefore, in this example, we encode the 3D joint position in two parts:

(1) 2D position in image space X_j, Y_j

(2) Relative depth in world space D_j

The 2D position can be reused from the pose linking process. Therefore, depth D_j is the only additional information needed. Based on the network output, the postprocessing execution mapping F will convert the 2D limb nodes to 3D space in the following way:

$$F(X_j, Y_j, D) \longrightarrow (X_{3D}, Y_{3D}, Z_{3D}). \quad (1)$$

We encode the depth value as the relative depth in the world space, which is to make the probability distribution of the depth values of the nodes as homogeneous as possible, thus facilitating the learning of the neural network. We consider the human pose as a tree structure, with the head as the root node. For each joint, the connected joint near the head will be the parent node, while the distant joints will be the child nodes. Then, the relative depth is encoded as the difference value of the depth of the current joint to the depth of the parent joint. We make sure that each joint has only one parent; otherwise, the relative depth is not unique. The reason why relative depth is easier to learn than absolute depth is that it is independent of body position and requires only local information. This is in line with the local field of perception characteristic of convolutional neural networks.

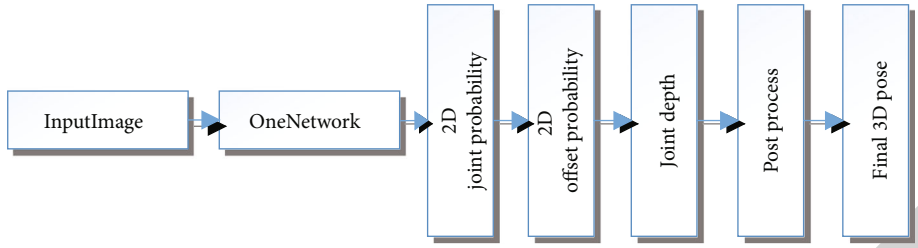

Figure 3: Basic framework of 3D pose estimation algorithm.

Obviously, there is no unique representation for 3D pose. A typical 3D pose representation is the following matrix:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}. \tag{2}$$

Each row represents the 3D coordinates of an articulation, and n is equal to the number of articulations of the current figure. This matrix representation is used by networks such as simple 3D baseline. However, this representation means that the network F can only perform the following mapping:

$$F\left(\begin{bmatrix} U_1 & V_1 \\ \vdots & \vdots \\ U_n & V_n \end{bmatrix}\right) = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}, \tag{3}$$

where UV represents the 2-dimensional coordinates of the nodes in the image space.

However, our desired output is as follows:

$$F(\text{Image}) = \left\{ P_1 \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}, P_2, \cdots P_n \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} \right\}. \tag{4}$$

In this case, the length of the output sequence is uncertain. Therefore, it is difficult to use neural networks for modeling. An optional way is to run the network F multiple times to output all sequences:

$$F(\text{Image}, \{P_1, P_2, \cdots P_{n-1}\}) = P_n \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}. \tag{5}$$

The problem with this scheme is that the number of runs of the network is uncertain, and the length of the input vector becomes longer as the length of the sequence grows. This is not conducive to the design of the network.

Another option is to divide the regions of each body in advance and then perform one-by-one mapping. This is the scheme adopted by simple 3D baseline [20–22].

In contrast to the design of these schemes, we have adopted a different idea. We want the mapping of the network itself F is a simple mapping:

$$F(\text{Image}) = \left\{ \text{Map}_{\text{class}}, \text{Map}_{\text{offset}}, \text{Map}_{\text{depth}} \right\}. \tag{6}$$

Then, add a new fast linking algorithm C that outputs the result as an infinitely long sequence:

$$C\left(\left\{ \text{Map}_{\text{class}}, \text{Map}_{\text{offset}}, \text{Map}_{\text{depth}} \right\}\right) = \left\{ P_1 \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}, P_2 \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}, \cdots P_n \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} \right\}. \tag{7}$$

This solution effectively avoids running the neural network repeatedly and ensures the speed of operation.

3.2. Multitasking and Multilevel 3D Detection Network. The design goal of our network is to maintain accuracy and achieve high operational speed. With this in mind, we propose and provide a new multitasking and multilevel 3D detection network architecture. Our network architecture consists of three parts' feature pyramid network, a 2D detection branch, and a depth detection branch. The depth detection branch looks almost identical to 2D detection, except for the map generation module. In each detection branch, we perform detection at different feature level and then connect them together. Finally, the map detection module analyzes the concatenated results and outputs out the final map.

Figure 3 illustrates the structure of our 3D pose detection network. First, a ResNet34-like backbone processes the input image and then comes the deconvolution and connection layers. These two structures build a U-Net structure. Then, there are two separate branches. One is used for 2D detection to output 2D probability maps and 2D offset maps. The other is used for the 3D depth map. Each detection branch consists of a multilevel detection module, a connectivity layer, and a map generation module.

3.3. Multilayered 3D Inspection Network Skeleton. Convolutional neural networks usually have the best detection target size. Therefore, in order to accommodate different target

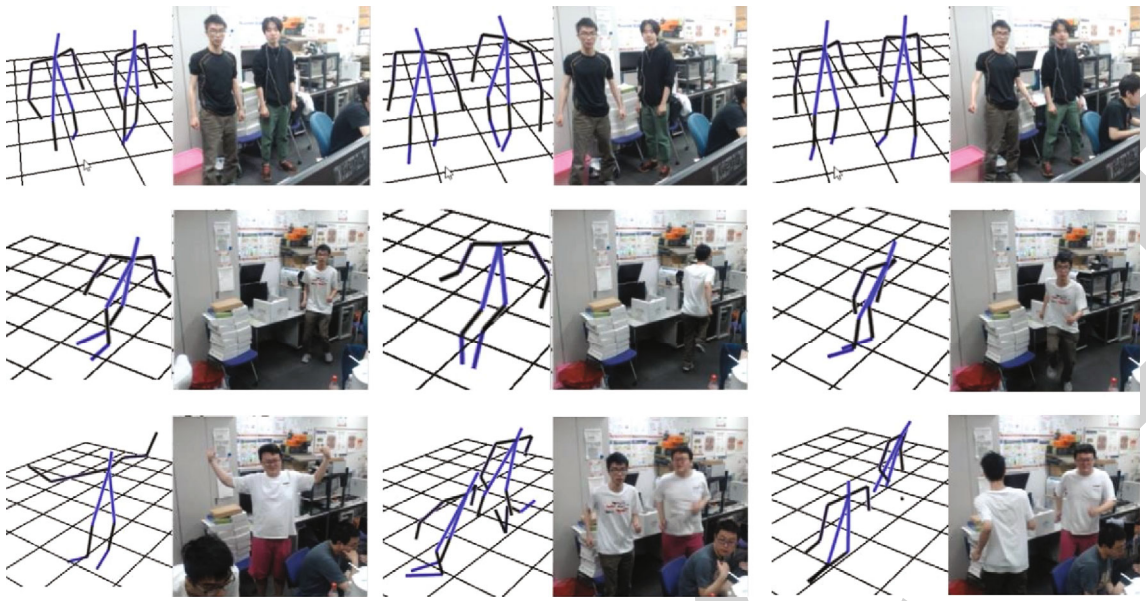

Figure 4: Generation results of different models trained at different epochs.

Figure 5: Pattern collapse in model training.

Table 1: Number of pattern collapses for different models at 300 iterations of epoch.

Model name	Mode collapse times	Epoch with the earliest mode collapse
DCGAN (no label)	10	26
WGAN (no label)	4	51
LSGAN (no label)	10	24
CGAN (labeled)	4	48
ACGAN (labeled)	2	63
LMV-ACGAN (labeled)	0	—

sizes, the usual approach is to process the input image multiple times with different scaling. However, in our case, processing the input image multiple times would take a lot of time. Instead, we use a multilayer architecture based on a feature pyramid network to perform detection at multiple scales through the network structure. This structure can adapt to the size of the target without adding too much to the computational cost. Since the computational cost savings are redundant, this data can be preserved without causing accuracy loss. For each region in the input image, there is only one optimal detection level, which means that no other detection levels need to be computed. The network structure of the multiscale feature pyramid has already been discussed in the context of the 2D pose recognition algorithm, so again we will not dwell too much on it [16].

Here, we only describe the parts of the 3D pose estimation network structure that differ significantly from the 2D

Table 2: Mean values of FID scores for different models.

Model name	FID score
WGAN (no label)	153.64 ± 1.44
CGAN (labeled)	99.83 ± 1.13
ACGAN (labeled)	92.23 ± 1.21
LMV-ACGAN (labeled)	79.42 ± 0.92

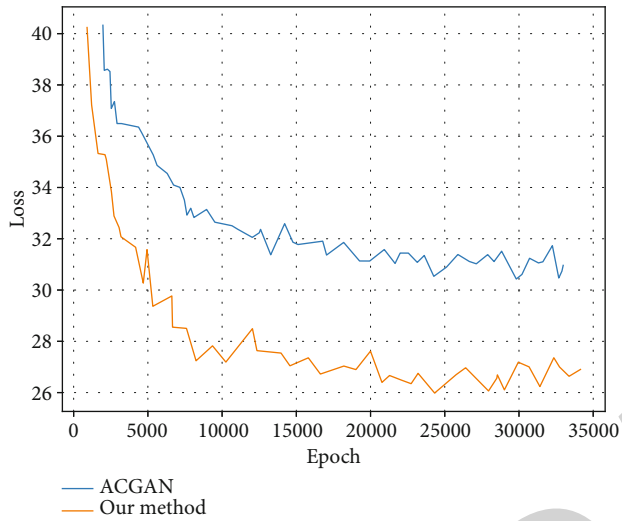

Figure 6: Comparison of classification loss.

network structure. In detail, we use ResNet34 as the front-end but compress the channels to half. The detailed structure is shown in the table. After this, there is an additional convolutional layer. Then, there are three pairs of deconvolution layers and the connection layers.

3.4. Design of the Loss Function. As a multitask neural network, we need to carefully design the loss function to balance each different branch. Otherwise, one branch may be weaker than the others.

In our case, for the classification branch, we directly use the L2 loss instead of the maximum cross-entropy loss. Then, for the other two branches, we focus only on the locations with joints, not the background. For the background region, we directly ignore the loss and allow the branch to output any result. We find that this makes the task easier, and the network can achieve lower errors.

4. Experiments and Analysis of Results

4.1. Data Set and Labels. In order to train the generative model in this paper, we collected about 25,000 headshots from the Web, from different people and with different resolutions. For the interception of people, we use the open source lbpcas-cade animeface script to locate and intercept the faces of anime characters from large images. In line with our purpose of image generation, the model in this paper is mainly applied to the avatars of personal information pages, etc. Therefore, these images are uniformly scaled to 64×64 and the higher resolution version 96×96 for training the network in this paper.

Since the generative model in this paper has an auxiliary classifier, it is necessary to classify the images in the real dataset. We found that when using the original WGAN [4], LSGAN [18], DCGAN [3], etc. for image generation, the hair of the person is blended with the background, and the boundary is not clear, resulting in poor visual realism, so the model in this paper uses hair color as the class of the image, in order that the model can learn how to "draw hair" to increase the realism of the image.

Based on the actual data set and the theory related to CIEDE2000 color difference calculation, we finally determined 8 categories of colors, which are black, white, red, green, blue, brown, purple, and yellow. Among them, red, green, blue, and yellow are the four colors with 90° color phase angle separation on the LAB color space, while the two colors brown and purple are 45° apart from yellow, green, red, and blue, respectively, on the color phase diagram. By intercepting the hair part in the sample to get its color, the CIEDE2000 color difference theory is used to calculate the class to which the color should most belong to determine the class of the image, so as to get the label.

4.2. Generate Model Evaluation

4.2.1. Generating Diversity and Resolving Pattern Collapse. Pattern collapse is the process in which the generative model converges to a point in the output space incorrectly in order to achieve the goal of deceiving the discriminator, i.e., generating an almost identical image, and even this image may be very unrealistic. This problem is better solved by the model in this paper on the collected data set. Figure 4 shows the generation of training images with different models at different epochs. The images generated by this model are more aesthetically pleasing to human vision than other models in general.

We find that these models can generate images normally in the early training period, but after the training epoch reaches 100 generations, both DCGAN and LS-GAN are prone to pattern collapse, as shown in Figure 5. In contrast, this model solves the pattern collapse problem to a certain extent and improves the "realism" of image generation.

To verify the reliability of our model on this dataset (less prone to pattern collapse), we counted several different GAN models and conducted 10 experiments at the maximum number of iterations we set (at which the model should have converged), and the number of pattern collapses occurred as listed in Table 1.

Among them, the WGAN experiment uses the RMS optimizer and sets the learning rate to 0.002; DCGAN, LSGAN, and ACGAN use the Adam optimizer and sets the learning rate to 0.0002, while our LMV-ACGAN uses the RMS optimizer and sets the learning rate to 0.002 for the optimization of G and D___CNN parameters and uses the Adam optimizer and sets the learning rate to 0.0002 for the three*M*LP networks with Adam optimizer and learning rate set to 0.0002, beta1 to 0.8, and beta2 to 0.98.

For generative diversity, we use the FID (Fréchet Inception Distance) metric to judge the goodness of the model. x denotes the distribution of the real sample set, g denotes the

distribution of the generator output, μ denotes the mean of the distribution, Σ denotes the variance of the distribution, and Tr denotes the trace of the matrix, and the FID is performed according to equation (8) calculation.

$$\mathrm{FID}(x, g) = \left\|\mu_x - \mu_g\right\|_2^2 + \mathrm{Tr}\left(\Sigma_x + \Sigma_g - 2\sqrt{\Sigma_x \Sigma_g}\right), \quad (8)$$

where μ_x and μ_g denote the mean values of the features of the real image and the generated image after extracting the intermediate layer by the same inception network, and Σ_x and Σ_g denote the variance of the extracted intermediate layer feature values. Lower FID implies higher quality and diversity of images.

Since FID is more sensitive to model collapse and more robust to noise, this distance score of FID will be quite high if there is only one image. Therefore, FID can be used to describe the diversity of GAN networks.

In this paper, we experimented with several GAN models in Table 2 and obtained the generator model by training 100,000 batches without pattern collapse and generated 25,000 samples as the distribution sampling of g by this model and calculated the FID distance from the real data set as the FID score of this generator model.

From the experimental data in Table 2, it can be analyzed that providing labeled data on this generation task can be very effective in improving the generation quality of the generative model and at the same time can effectively reduce the probability of pattern collapse occurrence. At the same time, the model in this paper can better ensure the diversity of image generation.

Figure 6 compares the changes of auxiliary classifier losses during training between the original ACGAN and our method. From the figure, it can be seen that although the increase of hidden parameters and multiscale discriminations lead to the increase of model parameters, the increase of parameters does not affect the convergence speed of this model excessively. At the same time, due to the addition of the discriminant information, the mean value of classification loss decreases from 2.94 to 2.81 after 35,000 iterations, which improves the classification ability of the auxiliary classifier to a certain extent, and this also improves the accuracy of the generator G in generating images by category. At the same time, this model can achieve better discriminative results with fewer iterations than ACGAN.

5. Conclusions

In this paper, two new neural network structures are designed for 2D and 3D pose extraction tasks, and the corresponding GPU-oriented acceleration schemes are given. Experimental results show that the pose recognition and animation generation system proposed in this paper achieves the set speed and accuracy goals with an average error of only 110 mm, and real-time animation generation can reach a speed of more than 30 frames per second. It demonstrates the successful application of this paper in the field of computer graphics based on deep learning techniques.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

References

[1] Y. Lu, J. Chai, and X. Cao, "Live speech portraits: real-time photorealistic talking-head animation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–17, 2021.

[2] M. Z. Lifkooee, C. Liu, Y. Liang, Y. Zhu, and X. Li, "Real-time avatar pose transfer and motion generation using locally encoded Laplacian offsets," *Journal of Computer Science and Technology*, vol. 34, no. 2, pp. 256–271, 2019.

[3] D. P. P. Nagalakshmi Vallabhaneni, "The analysis of the impact of yoga on healthcare and conventional strategies for human pose recognition," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 1772–1783, 2021.

[4] L. Yu, J. Yu, M. Li, and Q. Ling, "Multimodal Inputs Driven Talking Face Generation With Spatial–Temporal Dependency," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 203–216, 2020.

[5] M. M. Tiwari, M. T. Tiwari, G. Rajendran, and R. Suson, "Deep learning approach for generating 2D pose estimation from video for motion capture animation," *International Journal of Future Generation Communication and Networking*, vol. 13, no. 2, pp. 1556–1561, 2020.

[6] N. Kang, J. Bai, J. Pan, and H. Qin, "Interactive animation generation of virtual characters using single RGB-D camera," *The Visual Computer*, vol. 35, no. 6-8, pp. 849–860, 2019.

[7] H. Li, D. Zeng, L. Chen, Q. Chen, M. Wang, and C. Zhang, "Immune multipath reliable transmission with fault tolerance in wireless sensor networks," in *International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 513–517, Springer, Singapore, January 2016.

[8] C. H. Cao, Y. N. Tang, D. Y. Huang, G. Wei Min, and Z. Chunjiong, "IIBE: an improved identity-based encryption algorithm for WSN security," *Security and Communication Networks*, vol. 2021, Article ID 8527068, 8 pages, 2021.

[9] X. I. E. Tao, C. ZHANG, and X. U. Yongjian, "Collaborative parameter update based on average variance reduction of historical gradients," *Journal of Electronics and Information Technology*, vol. 43, no. 4, pp. 956–964, 2021.

[10] X. Meng, J. Pan, H. Qin, and P. Ge, "Real-time fish animation generation by monocular camera," *Computers & Graphics*, vol. 71, pp. 55–65, 2018.

[11] K. Sato, T. Nose, and A. Ito, "HMM-based photo-realistic talking face synthesis using facial expression parameter mapping with deep neural networks," *Journal of Computer and Communications*, vol. 5, no. 10, p. 50, 2017.

[12] C. Zhi-chao and L. Zhang, "Key pose recognition toward sports scene using deeply-learned model," *Journal of Visual Communication and Image Representation*, vol. 63, p. 102571, 2019.

[13] N. Liu, T. Zhou, Y. Ji, Z. Zhao, and L. Wan, "Synthesizing talking faces from text and audio: an autoencoder and sequence-to-sequence convolutional neural network," *Pattern Recognition*, vol. 102, p. 107231, 2020.

[14] S. Raman, R. Maskeliūnas, and R. Damaševičius, "Markerless dog pose recognition in the wild using ResNet deep learning model," *Computers*, vol. 11, no. 1, p. 2, 2022.

[15] D. Xu, X. Qi, C. Li, Z. Sheng, and H. Huang, "Wise information technology of med: human pose recognition in elderly care," *Sensors*, vol. 21, no. 21, p. 7130, 2021.

[16] N. Chen, Y. Chang, H. Liu, L. Huang, and H. Zhang, "Human pose recognition based on skeleton fusion from multiple kinects," in *In 2018 37th Chinese Control Conference*, Wuhan, China, 2018, July.

[17] H. Wang, P. He, N. Li, and J. Cao, "Pose recognition of 3D human shapes via multi-view CNN with ordered view feature fusion," *Electronics*, vol. 9, no. 9, p. 1368, 2020.

[18] L. Van Tran and H. Y. Lin, "BiLuNetICP: a deep neural network for object semantic segmentation and 6D pose recognition," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11748–11757, 2020.

[19] M. Burns, F. Cruciani, P. Morrow, C. Nugent, and S. McClean, "Using convolutional neural networks with multiple thermal sensors for unobtrusive pose recognition," *Sensors*, vol. 20, no. 23, p. 6932, 2020.

[20] S. Fujiuchi, R. Hachiuma, K. Hasegawa, and H. Saito, "Synthesize talking anime-heads images by tunneling through human-heads domain," *IEICE Technical Report; IEICE Tech. Rep.*, vol. 120, no. 300, pp. 7–11, 2020.

[21] R. Liao, S. Yu, W. An, and Y. Huang, "A model-based gait recognition method with body pose and human prior knowledge," *Pattern Recognition*, vol. 98, p. 107069, 2020.

[22] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3D human pose estimation: motion to the rescue," *Advances in Neural Information Processing Systems*, vol. 32, pp. 12949–12961, 2019.