DiffAnnot: Improved Neural Annotator with Denoising Diffusion Model

Chaofan Lin
Zhiyuan College
Shanghai Jiao Tong University
Shanghai, China
siriusneo@sjtu.edu.cn

Tianyuan Qiu
Zhiyuan College
Shanghai Jiao Tong University
Shanghai, China
frank_qiu@sjtu.edu.cn

Hanchong Yan
Zhiyuan College
Shanghai Jiao Tong University
Shanghai, China
hanchong1128@sjtu.edu.cn

Muzi Tao
Zhiyuan College
Shanghai Jiao Tong University
Shanghai, China
mzi1228@sjtu.edu.cn

3D human reconstruction is an important task in computer vision to generate human 3D models from photos or videos. But the research of reconstruction requires human body data with annotation. Previous widely used neural-network-based annotators annotate 3D human models automatically but there is still room for quality improvement. In our research, we innovatively regard the deviation between the annotations and true human poses as a type of noise. We propose a new annotator DiffAnnot using a denoising diffusion model to further refine the annotations from a pre-trained annotator. We train and test DiffAnnot on various datasets including 3DPM, MPI-INF-3DHP and Halpe. DiffAnnot is evaluated on several widely used metrics including MPJPE and shows outstanding performance. The code is available publicly at *https://github.com/PaperL/Human-3D-Diffusion*.

Keywords—3D Human Reconstruction, 3D Annotation, Diffusion Model

I. INTRODUCTION

3D human reconstruction is aiming to localize human mesh vertices in the 3D space. Human body reconstruction based on images is an essential research task for activity understanding[1], image and video editing, VR/AR content creation, and costume design in films and games. Reconstruction for other 3D objects such as cars also has a wide range of application scenarios[2]. Human body reconstruction provides concise and simple human body models for further research in these areas. Precise and effective human reconstruction result makes it possible for other subsequent research to reduce error in the data end, therefore enhancing the training efficiency.

The existing reconstruction methods can be roughly divided into parameterized methods and non-parameterized methods[3]. Non-parameterized methods directly reconstruct the high-dimensional human surface grid instead of the low-dimensional parameter representation. This kind of method generally requires some special data acquisition equipment, such as a laser scanner or depth camera. There are also solutions using a direct method to extract features from sketches and reconstruct 3D objects[4]. The parametric reconstruction methods rely on a parameterized human body model based on statistics, and a set of low dimensional vectors, namely human parameters, describing the human 3D shape. Common parameterized human models include SCAPE[14], SMPL[12], SMPL-X[13], etc. And to fit these models, annotated data is required.

Data annotation is the action that annotators process data used in machine learning with the help of annotation tools, including image annotation, voice annotation, text annotation, etc. In the research of human body reconstruction, data annotation is a part of the data preparation. Although traditional manual data annotation has brought many job opportunities, 3D human annotation is costly and time-consuming. Additionally, most datasets related to the project are annotated by hardware, complicated and expensive to use. For these reasons, it is particularly necessary to research automatic annotation.

Previous work on 3D human automatic annotation has mainly included optimization-based 3D pseudo-GT annotators and neural network-based 3D pseudo-GT annotators. The optimization-based annotators fit 3D human model parameters for target 2D/3D joint coordinates or 3D point clouds, only fitting 3D human model parameters to each sample without considering others, thus they often produce wrong 3D pseudo-GT results. Recently introduced neural network-based 3D pseudo-GT annotators like SPIN[5] predict SMPL[12] parameters using a neural network and run an optimization-based annotator. NeuralAnnot[6], a neural annotator proposed by Gyeongsik et al, brought little insight, but it only aggregated different datasets such as MoCap datasets[28] and In-the-wild datasets. These annotators are all still to be improved for further application to automatically generate annotation.

In this paper, we present a new neural annotator DiffAnnot, using a diffusion model to denoise and perform greatly in some 3D human datasets. We apply the latest progression in diffusion models to solve this problem from a denoising view, inspired by the improved denoising diffusion probabilistic models[25] proposed by Alex and Prafulla. These denoising diffusion models are a class of generative models, matching a data distribution by learning to reverse a multi-step noising process. Through the combination of the diffusion model and existing 3D human reconstruction models like Human Mesh Recovery (HMR)[7], we observe a remarkable loss decline in some classical conventional datasets such as 3DPW and MPI-INF-

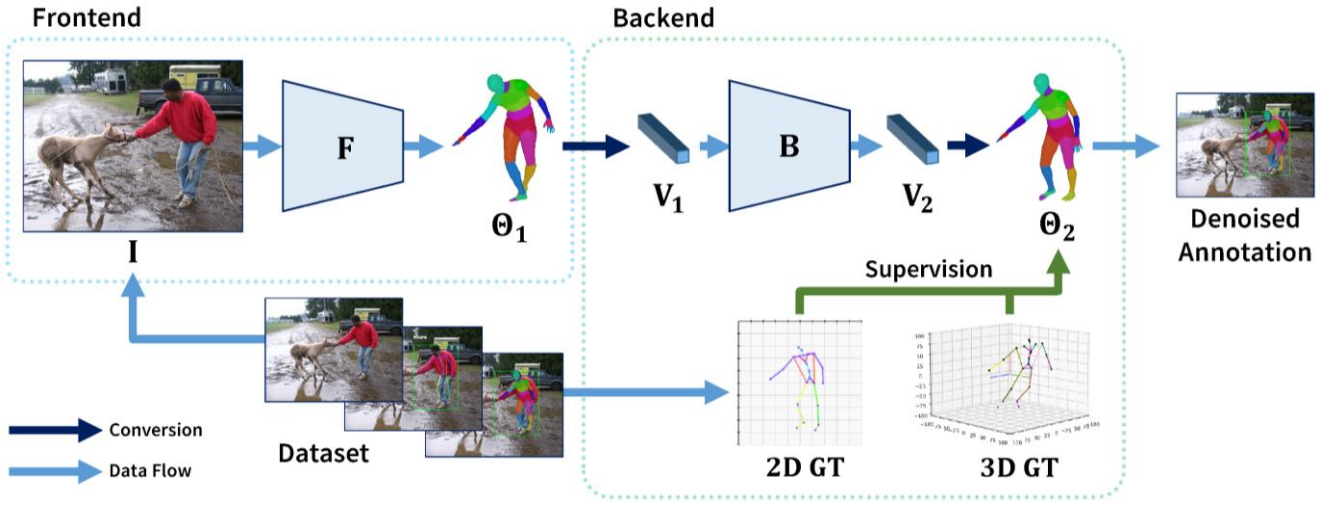

Fig. 1. DiffAnnot architecture overview. **Dataset** provides original image as model input **I** and **2D/3D ground truth**(joint coordinates) as supervision. **Frontend** takes input image **I** and uses pre-trained HMR model **F** to get rudimentary parameters $\boldsymbol{\Theta_1}$. **Backend** first coverts $\boldsymbol{\Theta_1}$ to a flatten vector $\mathbf{V_1}$. The diffusion model **B** denoises $\mathbf{V_1}$ and gets vector $\mathbf{V_2}$. Then $\mathbf{V_2}$ is restored to parameters $\boldsymbol{\Theta_2}$. **2D/3D ground truth** is used to calculate loss with $\boldsymbol{\Theta_2}$. The loss supervises **B** to learn a data distribution for denoising. In evaluation, $\boldsymbol{\Theta_2}$ is the **denoised annotation** as the result of DiffAnnot.

3DHP[15][16]. We also evaluate our method on Halpe[17], achieving impressive performance.

Our contribution and value of research can be summarized as follows:

- We present DiffAnnot, a new neural annotator applying denoising diffusion probabilistic models on existing 3D human reconstruction models, which is a momentous innovation.
- Our model performs extraordinarily well in some both conventional 3D human datasets like 3DPW and datasets for other purposes like Halpe, bringing obvious loss decline and obtaining better visualization results. Besides, our method has great robustness and extensibility.
- Automatic annotation with our method has greatly saved human and material sources, providing convenient support for future research.

II. Related Work

A. 3D Human Reconstruction and Annotation

Recent methods have shown progress rapidly in estimating the major body joints. There are many parameterized human models including SCAPE, SMPL, SMPL-X, etc. There are also many methods that estimate 3D bodies from images and several methods use deep learning to regress the parameters of SMPL. Zheng et al. proposed DeepHuman[8], an image-guided translation CNN for 3D human construction, fusing different scales of image features into the 3D space through volumetric feature transformation. Kanazawa et al. presented Human Mesh Recovery (HMR) [7], an end-to-end framework for reconstructing full 3D meshes from a single RGB image. As for neural annotation about 3D human construction, Moon et al. proposed the first one-stage neural network-based annotator[6], producing much more accurate pseudo-GTs than previous optimization-based annotators.

B. Denoising Deffusion Probabilistic Model

Sohl-Dickstein et al. introduced a class of generative models that match a data distribution via learning to reverse a gradual, multi-step noising process, namely diffusion probabilistic models[22]. Recently, Ho et al. showed equivalence between DDPM and score-based generative models Song and Ermon proposed to denoise score matching[23][24]. They also produced high-quality images using DDPM. Nichol et al. improved the generative model with some simple modifications, achieving competitive loglikelihoods while maintaining high sample quality.

III. Methods

In this section, we give a detailed introduction about the architecture of our DiffAnnot, which contains a classical human reconstruction model, a diffusion model and some intermediate components. Besides, its training strategy and some practice details are immediately followed.

A. Architecture Overview

The basic idea is that there is a deviation between reconstruction results of current method and the ground truth. We assume that this deviation can be regarded as a type of noise and then train a denoising model using a large amount of 3D human data to refine the previous reconstruction results. We use a diffusion model to do this denoising job given that it is one of the best solutions in this field.

Our DiffAnnot takes image **I** as the input and outputs revised 3D human model parameters $\boldsymbol{\Theta_2}$. **I** will firstly enter a normal human reconstruction model **F**, which is called the frontend of our architecture. The frontend predicts rudimentary parameters $\boldsymbol{\Theta_1}$. Indeed, the frontend is an implementation of the previous NeuralAnnot work which can do this task on its own with relatively not that good results. Then $\boldsymbol{\Theta_1}$ is processed to a flattened vector **V** and fed into the latter part, a diffusion model. The diffusion model **B**, called the backend, denoises the

previous result $\mathbf{\Theta_1}$ as mentioned above. After being trained, we can get the refined result $\mathbf{\Theta_2}$ from **B**, which is the final output of our annotator expected to achieve a better reconstruction than $\mathbf{\Theta_1}$.

B. 3D Human Model Parameters

We adopt SMPL24[12] as our 3D human model parameter model, which consists of 24 pose vectors $\{\boldsymbol{\theta_i}\}$ and 10 linear shape coefficients $\{\boldsymbol{\beta_i}\}$. Usually the first pose vector is separated out and called global orient or root, while the left 23 are known as body poses corresponding to 23 joints.

The linear shape coefficient $\boldsymbol{\beta_i}$ is just a scalar while there are many approaches to represent the pose vector $\boldsymbol{\theta_i}$, such as axis angle representation, rotation matrices and quadratic numbers. We choose rotation matrices as the universal representation in our architecture given that it brings more parameters to backend and enables more precise refinement.

About the flatten procedure, we should take a look at some shapes. According to the above, we can store $\{\boldsymbol{\theta_i}\}$ in a $24 \times 3 \times 3$ tensor in the rotation matrices representation. Therefore, the converted vector **V** has 226 dimensions.

C. Frontend

The frontend **F** itself is a neural annotator with classical HMR[7] as the end-to-end reconstruction method. Overall, HMR is a GAN architecture[9], which is a ResNet-50 encoder[10], a 3D regression component and a discriminator. Since it is an end-to-end reconstruction framework, **F** takes the image I as the input and directly predicts human shape and pose in the above form $\mathbf{\Theta_1}$.

D. Backend

The backend is nothing more than a diffusion model. While we use it to denoise, it is actually a probabilistic generative model. We adopt the improved version of DDPMs[25] which is proven to be equivalent to the score based generative models[26]. For a deep understanding of the reason that we use it to refine the human reconstruction, it is necessary to briefly review the formulation of DDPMs.

Generally, the diffusion model contains two processes: the forward process or the diffusion process and the backward process or the denoising process. Starting from a data distribution $x_0 \sim q(x_0)$, what the forward process does is adding Gaussian noise to the original data distribution step by step according to some variance β_t.

$$q(x_t|x_{t-1}) = \mathcal{N}(\mathrm{x_t}; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

Knowing how to do such variance schedule or noise schedule is the most important question in a diffusion model. The variance schedule here can be learned by reparameterization [29]. By reparameterizing, with $\alpha_t = 1 - \beta_t$ and $\overline{\alpha_t} = \prod_{s=0}^{t} \alpha_s$, we have

$$x_t = \sqrt{\overline{\alpha_t}}x_0 + \sqrt{1-\overline{\alpha_t}}\epsilon \tag{2}$$

where ϵ is the standard Gaussian noise with $\epsilon \sim \mathcal{N}(0,1)$. And we get the diffusion kernel as follows.

$$q(x_t|x_{t-1}, x_0) = \mathcal{N}(\mathrm{x_t}; \sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})\mathbf{I}) \tag{3}$$

For the backward process, it is easy to apply Bayes theorem to derive Eq (5).

$$\widetilde{\mu_t}(\mathrm{x_t}, \mathrm{x_0}) = \frac{\sqrt{\overline{\alpha_{t-1}}}\,\beta_t x_0 + \sqrt{\alpha_t}(1-\overline{\alpha_{t-1}})x_t}{1-\overline{\alpha_t}}$$

$$q(x_{t-1}|x_t,\ x_0) = \mathcal{N}(\mathrm{x_t}; \widetilde{\mu_t}(\mathrm{x_t}, \mathrm{x_0}), \tilde{\beta}_t\mathbf{I}) \tag{5}$$

where $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\overline{\alpha_{t-1}}}{1-\alpha_t}\beta_t$. Eq (3) and Eq (5) can generally represent these two processes and give us a math overview of DDPMs.

Practically, to make the diffusion model adaptive to **V**, we modify some components in the current diffusion model, replacing UNet[11] with a simple multi-layer perceptron.

And after being trained in the dataset, we sample the refined result $\mathbf{\Theta_2}$ from the diffusion model, starting from the original output $\mathbf{\Theta_1}$. The sampling or inference rule will be introduced later.

E. Training Strategy

Since the frontend is pretrained, we only introduce the training algorithm for our modified diffusion model here. Recall that the target of the diffusion model is to learn a distribution $\widetilde{\mu_t}$. To be supervised by 2D or 3D GT, we add a term in the loss function using the same metrics we will introduce in the experiment part, MPJPE for 2D GT supervision and PA-MPJPE for 3D GT supervision. To balance the original loss and our additionally introduced loss, coefficients γ_{3D}, γ_{2D} are needed to scale them. Then the loss function can be formulated as

$$L = L_{ori} + \gamma_{3D}L_{3D} + \gamma_{2D}L_{2D}$$

In the case 3D GT or 2D GT is not provided, the corresponding term is set to 0.

F. Inference

The inference here means the process of getting the refined result $\mathbf{\Theta_2}$ from the trained diffusion model by sampling starting from $\mathbf{\Theta_1}$. It is worth noting that in each denoising step the result does not always get better, for which we will give an intuitive visualization in the experiment part. Therefore, a head-on problem is how we get the optimized result when sampling, that is, select appropriate sampling steps s such that when we take input $\mathbf{\Theta_1}$ as x_s and denoising it back to x_0, the result $\mathbf{\Theta_2} = x_0$ is the best with respect to some specified metrics.

Note that when we train our diffusion model, there is a parameter s_{max} we must set to specify the maximum number of steps of diffusion. Let $L'(s)$ be the loss when taking s sampling steps, then our task can be formalized as find

$$s = \arg\min L'(s)$$

A simple idea may be directly searching $s \in \{0, 1, \dots, s_{max}\}$ to get the best result, which is certainly time-consuming. Observing that although L' may not be monotonous nor unimodal, it is unlikely to have multiple minimum points. Then we can assume that if the result is continuously getting worse for some period, it is time to stop searching. The algorithm can be described as follows.

Algorithm 1 SelectBestResult(model, x, gt, s_{max}, K)

$L_{last}, L_{best} \leftarrow +\infty$
best $\leftarrow$ None
counter $\leftarrow 0$
for i in $\{0, 1, \ldots, s_{max}\}$ **do**
  sampled $\leftarrow$ sample(model, x, i)
  $L_{now} \leftarrow$ calculate_loss(sampled, gt)
  if $L_{now} > L_{last}$ **then**
    counter $\leftarrow$ counter +1
  else
    counter $\leftarrow 0$
  end if
  if counter $> K$ **then**
    return best
  end if
  if $L_{now} < L_{best}$ **then**
    best $\leftarrow$ sampled
    $L_{best} \leftarrow L_{now}$
  end if
  $L_{last} \leftarrow L_{now}$
end for
return best

Fig. 2. DiffAnnot architecture overview.

where x is the input $\boldsymbol{\Theta_1}$, gt is the ground truth used to calculate loss and K is a threshold of how long this getting-worse situation is allowed to sustain.

IV. Experiments

In this section, we perform experiments to get a glimpse of the results of our DiffAnnot.

A. Implementation Details

Our implementation is strongly dependent on the 3D human parametric model toolbox MMHuman3D[27], whose HMR implementation serves as both our baseline and the frontend of our architecture. And the backend refers to the codebase of Improved Denoising Diffusion Probabilistic Models[25]. In detail, the backbone of the HMR is ResNet50 and pretrained on a mixed dataset: MoSh[18], MPI-INF-3DHP[16], LSP[19], LSPET[19], MPII[20], COCO 2014[21].

In respect of the parameters, the initial learning rate of our diffusion model is 8×10^{-5} and the EMA rate is set to 0.9999. Additionally, we use the diffusion with a uniform scheduled sampler and 1000 diffusion steps. K in the inference algorithm is set to 10 in our implementation.

B. Datasets

Our experiments are conducted in three different datasets: 3DPW[15], MPI-INF-3DHP[16], and Halpe[17]. As stated in table I, here the first two have 3D GT joint coordinates, which are commonly known as the MoCap datasets[28] while the last one only provides 2D joint coordinates, which are known as the In-the-Wild datasets. Our annotator can either be supervised by 3D joints or 2D.

TABLE I. Size and Content of Datasets

Dataset	Number of Pictures (Total)	Number of Pictures (Cleansed)	Has 3D GT?
3DPM	35515	35515	Yes
MPI-INF-3DHP	2875	2875	Yes
Halpe	38118	11809	No

Note that before feeding to our model, we need to do a data cleansing to get rid of some pictures which are unsuitable for our tasks, for example, pictures with too many people and pictures with a very incomplete human body. This is especially important in the Halpe dataset since it is not created specifically for this task. And for 3DPM and MPI-INF-3DHP, we only choose their test set since the HMR is pretrained on these two datasets using their train set.

C. Metrics

To measure the effectiveness of our method, we introduce two classical metrics in the region of human reconstruction. Because the annotator is supervised by joint positions, the loss is measured by mean per joint position error (MPJPE). It can be calculated as follows:

$$MPJPE = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \| J_i - J_i' \|^2 \quad (1)$$

Where N is the number of joints. Depending on different conventions, it can be 14, 17, 24, etc. And for those datasets which give 3D GTs, we can additionally add the PA-MPJPE as one of our metrics. Compared to MPJPE, it will do an alignment first by rotating or translating and then calculate the mean error.

D. Results

For comparison, we will use the original HMR as the backbone of the NeuralAnnot[6]. The main difference between these two annotators lies in whether there is a diffusion model to fix their annotations, which shows the effect of our method. We use the same pre-trained weights in frontend for a fair comparison. The results are shown in table II.

TABLE II. Loss Drement

Dataset	HMR Annotator		DiffAnnot (Ours)	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
3DPM	112.34	67.53	**111.14**	**65.53**
MPI-INF-3DHP	125.17	90.36	**124.63**	**86.28**
Halpe	103.54	/	**99.94**	/

Fig. 3. Comparison of several 3D human annotation visualization between HMR and DiffAnnot.

Those data illustrate that our method has a slight but significant improvement compared to the baseline neural annotator.

From a different perspective, the proportion of those with a decrement in respect of these two metrics in the overall data also makes some sense. Note that the diffusion part does not always make the results better, given that the original reconstruction result is good enough so the diffusion method only makes it deviate from the ground truth. But by simple observation, the result will not get worse as well since when sampling takes 0 step, it is equivalent to the original method. So we can divide the results into two parts: those being optimized and those remaining the same. The proportion is shown as the following table III:

TABLE III. PROPORTION OF OPTIMIZED SAMPLES IN DIFFERENT DATASETS

Dataset	Proportion of Optimized Samples
3DPM	0.68
MPI-INF-3DHP	0.78
Halpe	0.62

As can be seen from the table, averagely and approximately, 70% of the total data reach better results.

Here we can visualize some of the optimized data to deeply where and how the original reconstruction results are optimized. We select three frames in the 3DPW dataset, and visualize and compare the human mesh. Fig 3. (a) illustrates that our method puts the left man's arm down. This is reasonable because according to the context (meaning the before and after frames of current frame), this man's covered right arm should be close to his body. The diffusion model magically improves this part. The fix shown in Fig 3. (b) does a similar thing, which puts the man's right arm down to make it closer to his body.

Fig 3. (c) is the most exaggerated one. There is a big mistake in the original reconstruction result. The body of the fencer is totally opposite to the ground truth. As a result, the PA-MPJPE of this frame is 136.68, which is more than twice the average. Interestingly, it seems that our DiffAnnot tries to rotate the total body around to the correct position. But it is hard for the denoising process to completely correct this big mistake. It does its best effort towards the right direction and successfully drops the loss to 104.74, with over 20% decrement.

While we have seen two "Put Arm Down" fixes, Fig 3. (d) is a "Put Arm Up". As shown in the picture, our method successfully corrects the elbow to the correct position. However, in respect of the hand part, there is still room for improvement.

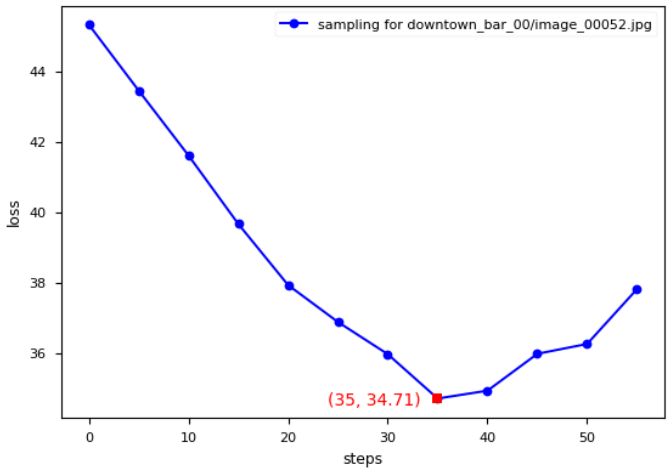

Fig. 4. Example of loss changes in the progress of sampling

Fig. 5. Visualization of steps in the sampling progress

E. Discussion in Sampling

As stated above, there is a simple algorithm in finding the best result during sampling. Here we can visualize the trend of loss and human mesh in the sampling progress, hoping to bring a deeper understanding of our method.

We choose the same frame as Fig 3.(a), which is downtown_bar_00/image_00052.jpg in 3DPW dataset, as an example.

Fig. 4. describes the relationship between the number of denoising steps and the loss. By selecting appropriate steps, which is 35 in this example, we can get the best result. To show that this phenomenon is not just overfitting but a significant improvement, we visualize the mesh changes in this process as Fig. 5.

We can see that the arm is gradually put down as the sampling steps increase. In step 35, the arm is lowered to the most suitable location which achieves minimal loss.

V. Conclusion

Our goal is to get a better neural annotator, which is able to automatically generate data annotation for a 3D mesh model of a human body from a single RGB image. To that end, our method DiffAnnot uses a score-based denoising diffusion model on the existing method, which has been able to generate annotations for the datasets. In DiffAnnot, the noise can be removed from the annotation generated by the existing method, leading to better results with higher accuracy. We applied our method to three datasets with thousands of images of real humans with diverse poses. Our results were compared with the annotations generated by HMR[7] model from MMHuman3D toolbox[27], which revealed that our method makes a remarkable improvement. Additionally, we visualized some examples of our refined 3D human annotations, compared them with original annotations and figured out the concrete body parts that have been improved. Moreover, the visualization may enlighten the areas to be improved, shedding light on the deeper research.

Acknowledgment

We would like to express our deepest appreciation to Prof. Cewu Lu and his Ph.D. student Xinpeng Liu for their inspiration and advice of this research. Thanks should also go to OpenMMLab project for the great open-source codebase MMHuman3D.

References

[1] Li, Yong-Lu, Xinpeng Liu, Han Lu, Shiyi Wang, Junqi Liu, Jiefeng Li, and Cewu Lu. "Detailed 2d-3d joint representation for human-object interaction." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10166-10175. 2020.

[2] Martin Dörfler, Tomáš Pivoňka, Karel Košnar, and Libor Přeučil, "Application of Surface Reconstruction for Car Undercarriage Inspection," Journal of Advances in Information Technology, Vol. 12, No. 4, pp. 327-333, November 2021. doi: 10.12720/jait.12.4.327-333.

[3] Ye, Mao, and Ruigang Yang. "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2345-2352. 2014.

[4] Masaji Tanaka, Toshiaki Kaneeda. "A Method of Reconstructing 3D Models from Sketches by Extracting Features." Vol. 5, No. 3, pp. 74-78, August, 2014. doi:10.4304/jait.5.3.74-78.

[5] Kolotouros, Nikos, Georgios Pavlakos, and Kostas Daniilidis. "Convolutional mesh regression for single-image human shape reconstruction." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4501-4510. 2019.

[6] Moon, Gyeongsik, and Kyoung Mu Lee. "Neuralannot: Neural annotator for in-the-wild expressive 3d human pose and mesh training sets." arXiv preprint arXiv:2011.11232 (2020).

[7] Kanazawa, Angjoo, Michael J. Black, David W. Jacobs, and Jitendra Malik. "End-to-end recovery of human shape and pose." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7122-7131. 2018.

[8] Zheng, Zerong, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. "Deephuman: 3d human reconstruction from a single image." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7739-7749. 2019.

[9] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." *Communications of the ACM* 63, no. 11 (2020): 139-144.

[10] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[11] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241. Springer, Cham, 2015.

[12] Loper, Matthew, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. "SMPL: A skinned multi-person linear model." *ACM transactions on graphics (TOG)* 34, no. 6 (2015): 1-16.

[13] Pavlakos, Georgios, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J. Black. "Expressive body capture: 3d hands, face, and body from a single image." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10975-10985. 2019.

[14] Anguelov, Dragomir, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. "Scape: shape completion and animation of people." In *ACM SIGGRAPH 2005 Papers*, pp. 408-416. 2005.

[15] Von Marcard, Timo, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. "Recovering accurate 3d human pose

in the wild using imus and a moving camera." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 601-617. 2018.

[16] Mehta, Dushyant, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. "Monocular 3d human pose estimation in the wild using improved cnn supervision." In *2017 international conference on 3D vision (3DV)*, pp. 506-516. IEEE, 2017.

[17] Fang, Hao-Shu, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[18] Loper, Matthew, Naureen Mahmood, and Michael J. Black. "MoSh: Motion and shape capture from sparse markers." *ACM Transactions on Graphics (ToG)* 33, no. 6 (2014): 1-13.

[19] Yu, Xiang, Feng Zhou, and Manmohan Chandraker. "Deep deformation network for object landmark localization." In *European conference on computer vision*, pp. 52-70. Springer, Cham, 2016.

[20] Andriluka, Mykhaylo, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. "2d human pose estimation: New benchmark and state of the art analysis." In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686-3693. 2014.

[21] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In *European conference on computer vision*, pp. 740-755. Springer, Cham, 2014.

[22] Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. "Deep unsupervised learning using nonequilibrium thermodynamics." In *International Conference on Machine Learning*, pp. 2256-2265. PMLR, 2015.

[23] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851.

[24] Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." *Advances in Neural Information Processing Systems* 32 (2019).

[25] Nichol, Alexander Quinn, and Prafulla Dhariwal. "Improved denoising diffusion probabilistic models." In *International Conference on Machine Learning*, pp. 8162-8171. PMLR, 2021.

[26] Song, Yang, and Stefano Ermon. "Improved techniques for training score-based generative models." Advances in neural information processing systems 33 (2020): 12438-12448.

[27] OpenMMLab: MMHuman3D codebase, https://github.com/open-mmlab/mmhuman3d

[28] MoCap dataset, http://mocap.cs.cmu.edu/

[29] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

AUTHORS' BACKGROUND

Your Name	Title	Research Field	Personal website
Chaofan Lin	Undergraduate	Machine Learning System	http://me.tric.space/
Tianyuan Qiu	Undergraduate	Computer Vision	http://www.paperlane.pub/
Hanchong Yan	Undergraduate	Computer Vision	
Muzi Tao	Undergraduate	Computer Vision	