Real-time 3D Human Pose Estimation from Video Based on Transformer

Weilin Chen,[1] **Jiahao Rao,**[2] **Wenhao Wang,**[3] **Yuchen Wan,**[4]

School of Informatics Xiamen University
[1]StuID :30920241154548 [2]StuID: 30920241154556
[3]StuID: 30920241154576 [4]StuID: 23020241154440

Abstract

3D human pose estimation is essential in applications like virtual reality, surveillance, and human-computer interaction. While CNNs and RNNs struggle with long-range dependencies in video, Transformer-based models such as PoseFormer have shown success in capturing spatiotemporal features for 3D human pose prediction. However, PoseFormer relies on future frames for accurate reconstruction, making it unsuitable for streaming or online tasks. Additionally, its inference demands significant computational resources, making it difficult for real-time use.To address these issues, Firstly, we propose the Pyramid Poseformer model based on the Poseformer. This model employs a Pyramid Transformer Encoder to replace its original Transformer Encoder, thereby reducing the consumption of computational resources. Secondly, we adjust the regression strategy of the model by regressing to the last frame of the input sequences, enabling it to adapt to real-time 3D human pose estimation task in videos. Thirdly, we leverage the Pyramid Poseformer model to train an SMPL(Skinned Multi-Person Linear model) parameter mapping model. By aligning the outputs of the Pyramid Poseformer with the inputs of the SMPL model, we achieve skinned representation of 3D human poses, enhancing the realism of the visualization.

Introduction

Human pose is the critical information in human-computer interaction, collaboration, and action analysis. The task of human pose estimation(HPE) aims to infer pose information from input images or videos, such as identifying and locating precise positions of key body joints (*e.g.* head, shoulders, elbows, wrists, knees, and ankles). By tracking and analyzing these key joints, we can further understand human behaviors and actions. Figure 1,demonstrates how we can recognize the action of eating by observing the sequence of movements in a subject's pose.

However, real-time human pose analysis from images or videos remains a significant challenge. Depth ambiguity and occlusion often lead to multiple possible 3D poses from a single 2D pose(Von Marcard et al. 2017). While many methods integrate temporal information from videos to improve accuracy, achieving real-time performance without sacrificing quality is difficult. Traditional models like temporal convolutional neural networks(Chen et al. 2021; Liu et al. 2020) and recurrent neural networks(Hossain and Little 2018) have been employed to capture temporal dependencies, but they struggle with long-range dependencies and computational efficiency, making them less ideal for real-time applications.

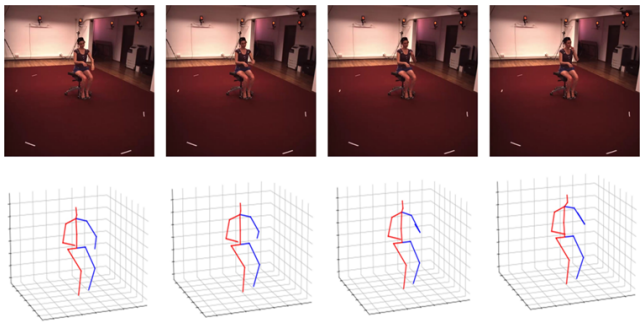

Figure 1: Example of 3D human pose estimation. By observing the sequence of a person's actions, it can be analyzed that the person is eating.

With the the advancement of deep learning, the Transformer model, known for its efficiency, scalability, and powerful modeling capabilities, has become dominant in natural language processing (NLP) due to its ability to capture global dependencies across long sequences. Consequently, Transformer models have naturally been extended to the domain of 3D human pose estimation (HPE). In recent studies, PoseFormer(Zheng et al. 2021),the first pure Transformer network for 2D-to-3D pose lifting, achieving state-of-the-art performance on several datasets and demonstrating strong expressiveness in HPE task.However, despite its accuracy, PoseFormer has yet to be fully optimized for real-time use, limiting its application in scenarios such as virtual reality (VR), augmented reality (AR), and other human-computer interaction systems that require both speed and precision.

In this work, we aim to bridge this gap by modifying PoseFormer to make it suitable for real-time 3D pose estimation from videos. Achieving this will significantly enhance the practical use of Transformer-based models in real-time environments, opening new possibilities for immersive and interactive applications in VR, AR, and beyond.

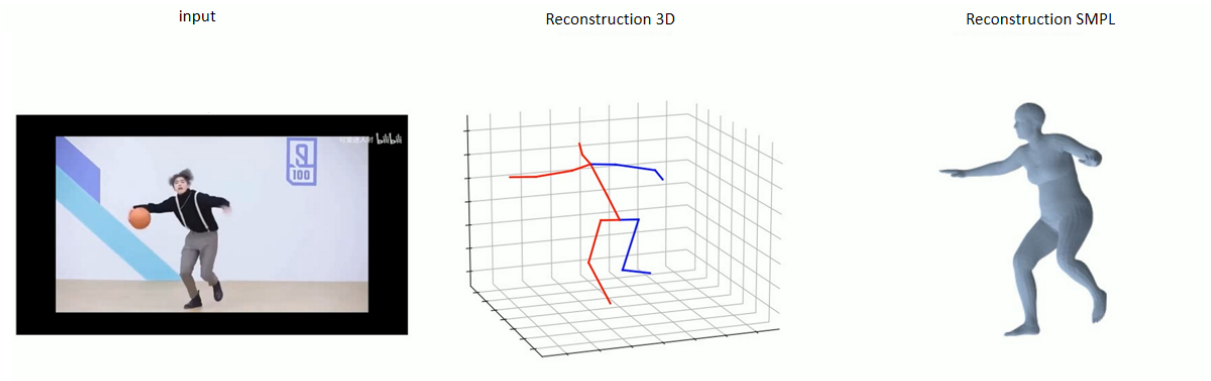

Figure 2: Visual representation of our work.The model takes a sequence of videos or images as input(lst-column), and outputs the reconstructed 3D human pose (2nd-column) as well as the skinned human body(3rd-column). It's noteworthy that all of this can be done in real-time (approximately 30 FPS), whereas the baseline cannot (10 FPS).

Related Work

3D Human Pose Estimation. Currently, the mainstream research approach for 3D pose estimation(HPE) is based on deep learning, which can be divided into direct estimation methods and 2D-to-3D Lifting methods(Li et al. 2022; Zheng et al. 2021; Zhang et al. 2022), as illustrated in Figure 3.

Direct methods use deep learning models such as CNNs or autoencoders to estimate 3D pose from images without the intermediate 2D pose representation, leveraging deep networks' fitting capabilities to avoid manual feature extraction. Certain methods(Pavlakos et al. 2017; Sun et al. 2018; Tekin et al. 2016)adopted this approach, which incurred significant computational expenses due to direct regression from the image space. In contrast, 2D-to-3D Lifting methods first obtain 2D poses and then predict 3D poses, benefiting from mature 2D pose estimation algorithms and offering simpler, faster training networks. Since this study focuses on video-based HPE using a Transformer-based architecture under the 2D-to-3D Lifting approaches, this section provides an overview solely of HPE based on Transformer architectures.

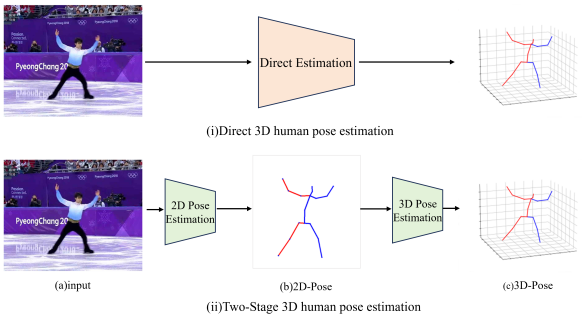

Figure 3: two approaches of 3D human pose estimation

Transformer architecture on 3D HPE. In 2017, Vaswani *et al.*(Vaswani 2017) introduced the Transformer architecture, excelling in computer vision tasks due to its self-attention mechanism. Yang *et al.*(Yang et al. 2020) combined it with convolutional blocks to create TransPose, explaining keypoint spatial dependencies. METRO(Lin, Wang, and Liu 2021) used Transformers for vertex-vertex and vertex-joint modeling, enabling 3D pose and mesh reconstruction from single images but neglecting temporal correlations in videos. Some researchers also explored the multi-view 3D human pose estimation scheme(He et al. 2020). Strided Transformer (Li et al. 2022) employed the Strided Transformer Encoder(STE) module to aggregate long-range information hierarchically, reducing costs but requiring a fixed Transformer order and only reconstructing the video's central frame, leading to information redundancy.

Poseformer and its variants. Zheng *et al.*(Zheng et al. 2021) introduced Poseformer, a Transformer-only architecture for 3D video pose estimation, built on ViT (Dosovitskiy 2020) without CNNs. It uses cascaded temporal and spatial Transformers to model joint relationships and frame sequences. Similar to Strided Transformer(Li et al. 2022), this architecture can only predict the 3D human pose of the center frame in the input sequence, belonging to the Seq2frame model, which inevitably suffers from issues such as redundancy in adjacent frames. On this basis, MixSTE(Zhang et al. 2022) uses a similar spatio-temporal Transformer structure to Poseformer, modeling the temporal movement of each joint and the spatial correlation between joints separately, constructing a Seq2Seq model. However, this structure relies on the data circulating between the two Transformer modules, resulting in significant computational overhead. PoseFormerV2(Zhao et al. 2023) were proposed to scale to long inputs by compactly representing skeletal sequences in the frequency domain, integrating temporal and frequency features while preserving PoseFormer's structure for better speed-accuracy balance.

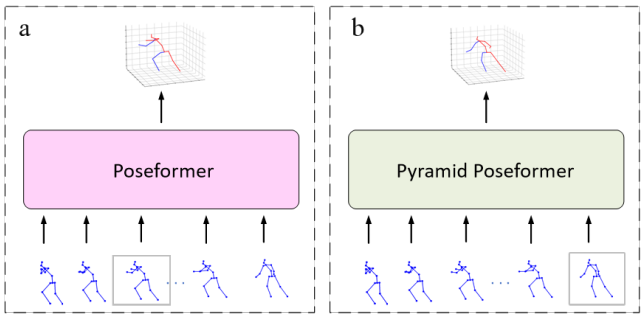

Figure 4: Comparison between the original PoseFormer model's regression strategy and the current approach during training.(a)the regression strategy of the original PoseFormer model;(b)the adjusted regression strategy of the PoseFormer model, called Pyramid Poseformer.

Proposed Solution

PoseFormer cleverly decouples the spatial and temporal information in video data, leveraging positional encoding and a multi-layer Transformer encoder to effectively capture the complex dependencies within input sequences, achieving accurate inference of human actions in key frames.

However, its reliance on spatiotemporal context requires waiting for future frames to regress the current frame as shown in Figure 4.(a). This can introduce inherent latency in real-time pose estimation. Additionally, PoseFormer's output is represented as 3D coordinates in the world coordinate system, which is incompatible with the axis-angle input format required by the SMPL model, limiting the ability to perform more detailed visualization of pose estimation results.

This part will explore methods to overcome these limitations, aiming to enhance the model's applicability in real-time pose estimation and subsequent visualization tasks.

Regression Strategy for video streams

To adapt PoseFormer for real-time video stream tasks, we have refined the regression strategy, with the detailed modifications illustrated in Figure 4.

The primary limitation of PoseFormer in this task is its dependence on future frames for estimating the current frame's pose, resulting in significant latency. To address this, we propose a modification in the regression target during training, shifting it from the middle frame of the sequence to the last frame as shown in Figure 4.(b). This adjustment allows the model to infer without relying on future frames, thereby eliminating the latency issue. At the same time, the model can still leverage past frames to capture sufficient motion context, ensuring that the accuracy of pose estimation is not severely impacted.

During inference, a queue is used to store a sequence of past frames, with its size matching PoseFormer's expected input sequence length. When a new frame is received, it is added to the queue while the oldest frame is removed. The model then performs inference on the updated sequence. According to the new training strategy, PoseFormer will regress the 3D pose of the last frame in the sequence, which corresponds to the current input frame. By introducing minimal changes to PoseFormer, this approach effectively balances speed and accuracy, making the model suitable for real-time pose estimation tasks.

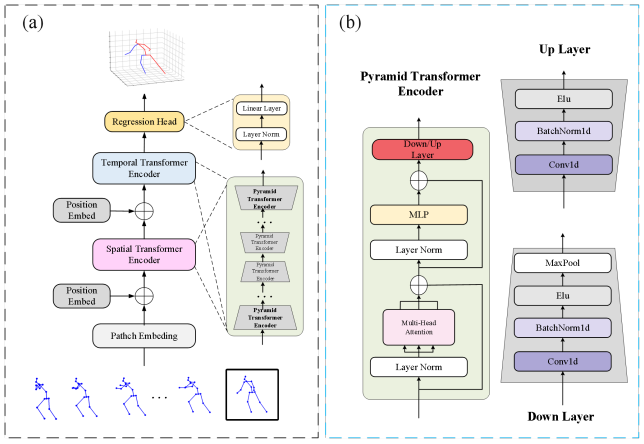

Figure 5: the detail of Pyramid Poseformer model network architecture. (a) shows the detailed architecture of the overall network; (b) illustrates the specific encoder design, including the upsampling and downsampling mechanisms.

Pyramid Poseformer

The core component of PoseFormer, the Transformer encoder, leverages the self-attention mechanism to address long-range dependencies and facilitate effective information flow, allowing the model to capture global information from the input sequence more effectively. However, this attention mechanism is computationally expensive, particularly for long sequences, which significantly increases the computational cost.

To enhance inference speed, we proposes the Pyramid PoseFormer, which aims to mitigate the computational burden associated with stacking multiple Transformer encoders. The architecture of the model is depicted in Figure 5.

Unlike the original PoseFormer, Pyramid Poseformer replaces the standard Transformer encoder with a Pyramid one. This new encoder builds upon the original design by adding a Down Layer or Up Layer to handle the dimensionality reduction and expansion of the input sequence.

In PoseFormer, the input matrix $Z_0 \in R^{J\times c}$ is processed through L layers of Transformer encoders, producing an output matrix $Z_L \in R^{J\times c}$ with the same dimensionality. In contrast, Pyramid PoseFormer adopts a hierarchical approach. The input matrix $Z_0 \in R^{J\times c}$ passes through $\frac{L}{2}$ stacked Down Pyramid Transformer encoders, progressively reducing its dimensionality to matrices of decreasing size:

$$Z_1 \in R^{J\times \frac{c}{2}}, Z_2 \in R^{J\times \frac{c}{4}}, \ldots, Z_{L/2} \in R^{J\times \frac{c}{2^{L/2}}} \quad (1)$$

Then, the output matrix passes through a series of stacked Up Pyramid Transformer encoders, gradually restoring the original input dimensions, with output sizes given by:

$$Z_{\frac{L}{2}+1} \in R^{J\times \frac{c}{2^{L/2-1}}}, Z_{\frac{L}{2}+2} \in R^{J\times \frac{c}{2^{L/2-2}}}, \ldots, Z_L \in R^{J\times \frac{c}{2}} \quad (2)$$

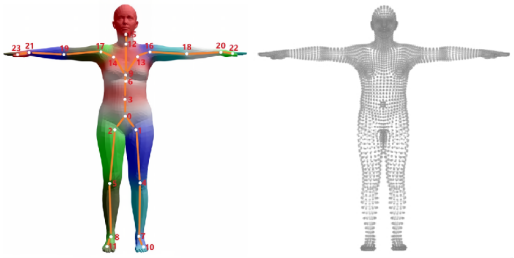

Figure 6: Joints, vertices, and mesh of the SMPL model

Skinned Multi-Person Linear Model

In the SMPL model, the hierarchical structure of 24 joints is defined using a kinematic tree. Joint 0 serves as the root, and the poses of the remaining 23 joints are determined by their rotation angles relative to their parent joints, as defined by the kinematic tree. These rotations are described using an axis-angle representation. Typically, the axis-angle format is a four-tuple (x, y, z, θ), which represents a rotation of θ degrees about an axis $\mathbf{e} = (x, y, z)^T$. In SMPL, this is simplified using a three-dimensional vector $\theta = (x, y, z)$, where the rotation axis is the unit vector $\mathbf{e} = \frac{\theta}{\|\theta\|}$ and the magnitude of the rotation is $\|\theta\|$.

To address the issue of end-to-end data incompatibility, we leverage the strong fitting capabilities of neural networks to learn the complex mapping from 3D coordinates to SMPL pose parameters, thus avoiding the need for manually designed conversion methods.

Experiment

Pyramid Poseformer

Dataset Human3.6M(Ionescu et al. 2013) is the most widely used indoor dataset in the field of 3D single-person pose estimation. It features 11 professional actors (6 male and 5 female) performing 15 actions such as sitting, walking, and making phone calls. The dataset contains a total of 3.6 million video frames captured in indoor environments. Each performer's pose is recorded from four different viewpoints and annotated with precise keypoint coordinates using a marker-based motion capture system.

Human3.6M uses MPJPE to evaluate model performance. MPJPE stands for mean per joint position error, which is the average of the Euclidean distances between the true and estimated positions of all joints, calculated as:

$$MPJE = \frac{1}{j}\sum_{k=1}^{j}\|p_k - \widehat{p_k}\|^2 \tag{3}$$

Where p_k and $\widehat{p_k}$ represent the true position and the estimated position of the kth joint respectively, j denotes the total number of joints, measured in millimeters.

Parameters This work follows the methodology of Poseformer(Zheng et al. 2021) and adopts the same experimental settings. During the data loading phase, horizontal flipping is used for data augmentation. The optimizer is Adam, with a learning rate set to 2e-4 and a weight decay factor of 1e-6. An exponential learning rate decay strategy is employed, with a decay factor of 0.98. The batch size is set to 512, and the model is trained for 150 epochs. All 15 actions are used for both training and testing, where the model is trained on datasets S1, S5, and S6, and tested on datasets S9 and S11 (Si represents the performer).

Table 1: Comparison between our model and the baseline.Our model performs comparably to the baseline, but with a 45% reduction in parameters and a 1.5x improvement in inference speed.

Model	Sequence Length(f)	Parameters(M) ↓	FLOPs(M) ↓	MPJPE ↓	FPS ↑
Poseformer	9	9.58	150	42.9	361
	27	9.59	452	38.3	339
	81	9.60	1358	35.1	317
Ours	9	4.45	102	43.0	556
	27	4.46	305	38.0	518
	81	4.47	920	35.2	**490(×1.55)**

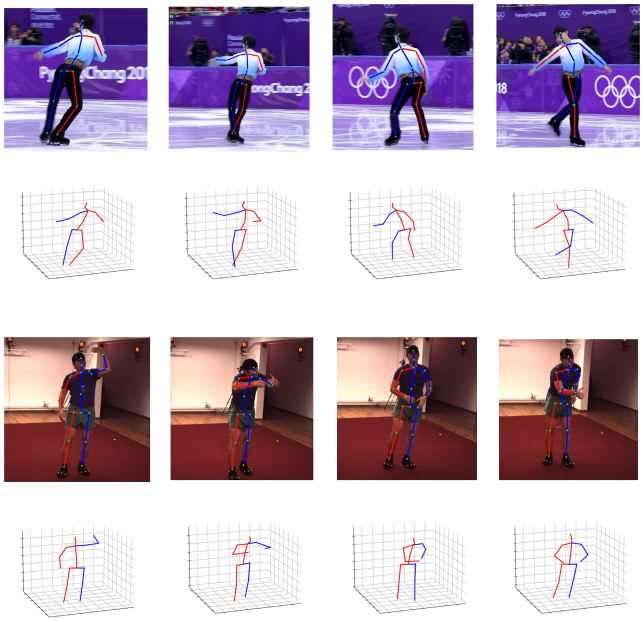

Figure 7: Pose Estimation Results of Pyramid Poseformer on skating action(top) and Human3.6M(down).

Results Table 2 shows the estimation error for different actions using the Poseformer and Pyramid Poseformer models under different regression strategies and input conditions.

Analysis of Table 2 For the same regression strategies, the estimation error of Pyramid Poseformer is comparable to that of Poseformer.Comparing regression strategies, the approach of regressing the final frame results in a 5%-8% performance drop for both models compared to reconstructing intermediate frames. However, this trade-off eliminates the reliance on future frames, making the models suitable for real-time estimation tasks.

When replacing ground-truth 2D coordinates with YOLO-Pose outputs, the introduction of additional uncer-

Table 2: Comparison of MPJPE↓ results across different models and regression strategies ($f = 81$).GT: use of ground-truth 2D coordinates; YOLO-Pose: use the 2D coordinates detected by the YOLO-Pose model;Pf: Poseformer; P-Pf:Pyramid Poseformer(ours);f:input sequence length; The symbol (*) denotes regression to the last frame of the input sequence, while unmarked entries represent regression to the middle frame.

Action	GT				YOLO-Pose			
	Pf	Pf(*)	P-Pf	P-Pf(*)	Pf	Pf(*)	P-Pf	P-Pf(*)
Direct.	34.7	39.4	34.5	39.0	43.8	48.8	44.2	49.5
Disc.	37.2	40.8	37.6	40.9	45.7	49.9	46.5	50.0
Eat	33.4	36.9	33.3	37.0	42.9	47.0	43.2	47.4
Greet	35.2	39.4	35.6	40.1	46.7	51.4	47.1	51.8
Phone	35.4	40.6	35.7	40.5	49.2	55.4	49.5	54.4
Photo	37.7	42.1	37.3	42.6	54.4	58.4	54.9	59.5
Pose	39.0	43.2	39.5	44.0	47.6	52.5	48.2	53.3
Pur.	33.5	35.6	33.2	36.3	43.9	45.4	44.2	46.5
Sit	41.3	44.8	41.8	44.9	57.4	60.8	58.1	62.2
SitD.	42.3	46.0	42.9	46.8	68.7	73.5	69.6	73.6
Smo.	35.3	38.9	35.6	39.3	47.7	51.8	48.2	51.4
Wait	35.3	39.1	35.6	39.6	42.7	47.2	43.0	48.0
WalkD	33.2	37.4	33.1	37.6	47.6	52.0	47.9	52.3
Walk	26.3	29.3	26.2	29.8	33.9	37.1	34.1	38.3
WalkT	25.9	28.7	25.8	28.3	34.1	37.6	34.3	36.2
Avg.	35.1	38.8	35.2	39.1	47.1	51.3	47.5	51.6

tainty and noise mimics real-world application scenarios. This inevitably leads to a decline in estimation accuracy. Nonetheless, the performance degradation is similar for both Pyramid Poseformer and Poseformer, highlighting the robustness and generalization capabilities of Pyramid Poseformer under noisy conditions.

Figure 7 illustrates the pose estimation results of Pyramid Poseformer on the real-world skating scenario and the Greeting action from subject S11, respectively. From the visual results, the estimated poses demonstrate clear humman contours and accurately captured key joint positions, indicating that the model effectively captures human posture information.

SMPL Parameter Mapping Model

Figure 8: SMPL reconstruction effects in different scenarios with different actions.

The model's performance is evaluated using Mean Per Vertex Error (MPVE), a metric commonly used in 3D shape reconstruction and registration tasks. MPVE represents the average Euclidean distance between all corresponding vertices of two models.

Table 3: MPVE ↓ of parameter mapping models on test sets (f=27)

Sequence Name	MPVE ↓
courtyard_basketball_01	0.174
courtyard_box_00	0.171
courtyard_laceShoe_00	0.162
courtyard_relaxOnBench_01	0.169
downtown_walkUphill_00	0.188
flat_packBags_00	0.152
outdoors_climbing_01	0.187
outdoors_freestyle_00	0.189
outdoors_parcours_00	0.221
outdoors_slalom_00	0.197
Average	0.181

As observed in Table 3, the parameter mapping model exhibits the highest estimation error in outdoor scenarios. This can be attributed to the increased complexity of outdoor environments, greater background noise, and more dynamic motions compared to other experimental settings. These factors lead to larger errors in capturing 2D poses using YOLO-Pose, which are further compounded during subsequent processing, ultimately amplifying the final estimation errors. Overall, with an input sequence length of 27 frames, the SMPL parameter mapping model achieves an average error of 0.181 on the test set, which is within an acceptable range for practical applications.

Figure 8 presents reconstructed 3D human models for various scenes and actions, generated by mapping 3D pose coordinates to SMPL parameters using the SMPL parameter mapping model.The results demonstrate that the mapping model effectively captures the relationship between 3D coordinates and SMPL parameters across different scenarios and actions, which highlights the model's robustness and stability in handling complex poses and movements.

Conclusion

This paper enhances the Poseformer model by addressing its limitations in real-time pose estimation. Firstly, a modified regression strategy is introduced to effectively handle video streams. Secondly, the Pyramid Poseformer model, featuring a Pyramid Transformer structure, significantly boosts inference speed while reducing computational costs. Lastly, an SMPL parameter mapping model is trained using the Pyramid Poseformer, enabling detailed reconstruction and visualization of human poses, thus balancing efficiency and accuracy.

References

Chen, T.; Fang, C.; Shen, X.; Zhu, Y.; Chen, Z.; and Luo, J. 2021. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1): 198–209.

Dosovitskiy, A. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

He, Y.; Yan, R.; Fragkiadaki, K.; and Yu, S.-I. 2020. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 7779–7788.

Hossain, M. R. I.; and Little, J. J. 2018. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, 68–84.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7): 1325–1339.

Li, W.; Liu, H.; Ding, R.; Liu, M.; Wang, P.; and Yang, W. 2022. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 25: 1282–1293.

Lin, K.; Wang, L.; and Liu, Z. 2021. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1954–1963.

Liu, R.; Shen, J.; Wang, H.; Chen, C.; Cheung, S.-c.; and Asari, V. 2020. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5064–5073.

Pavlakos, G.; Zhou, X.; Derpanis, K. G.; and Daniilidis, K. 2017. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7025–7034.

Sun, X.; Xiao, B.; Wei, F.; Liang, S.; and Wei, Y. 2018. Integral human pose regression. In *Proceedings of the European conference on computer vision (ECCV)*, 529–545.

Tekin, B.; Rozantsev, A.; Lepetit, V.; and Fua, P. 2016. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 991–1000.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Von Marcard, T.; Rosenhahn, B.; Black, M. J.; and Pons-Moll, G. 2017. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer graphics forum*, volume 36, 349–360. Wiley Online Library.

Yang, S.; Quan, Z.; Nie, M.; and Yang, W. 2020. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2(6).

Zhang, J.; Tu, Z.; Yang, J.; Chen, Y.; and Yuan, J. 2022. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13232–13242.

Zhao, Q.; Zheng, C.; Liu, M.; Wang, P.; and Chen, C. 2023. Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8877–8886.

Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; and Ding, Z. 2021. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 11656–11665.