HandDAGT

- Conference: ECCV 2024
- Authors: Wencan Cheng, Eunji Kim, Jong Hwan Ko
- Hand or Body: Hand
- Input: 2D depth image and 3D hand point cloud
- Motivation: Existing graph methods in 3D hand pose estimation use static graphs which are unable to capture dynamic kinematic relations between joints in occluded scenarios. Also, self-occlusion and hand-object occlusion are major challenges for current methods.
- Method: A local 3D PointNet-based encoder extracts 3D local features from the 3D hand point cloud to represent a subsampled set of 3D super points. The 2D depth map is passed through a 2D autoencoder to extract 2D local features and then projected to 3D and concatenated with the 3D local features. The 3D features are used to generate keypoint embeddings and determine a 3d point patch for each joint. The embeddings and patches are passed as queries and keys to an adaptive graph transformer, which dynamically balances local attention for visible keypoints and kinematic attention for occluded ones.
- Limitations: The model can't process interacting hands and focuses on single-hand samples. Also, this method requires redundant computation due to the 2D feature extraction. They state that possible solutions include bidirectional learning and developing a lightweight 2D model.
- Paper: https://arxiv.org/pdf/2407.20542
- GitHub: https://github.com/cwc1260/HandDAGT

HandDiff

- Conference: CVPR 2024
- Authors: Wencan Cheng, Hao Tang, Luc Van Gool, Jong Hwan Ko
- Hand or Body: Hand
- Input: 2D depth image and 3D hand point cloud
- Motivation: Depth image-based methods either use 2D CNN-based approaches, which do not accurately capture the 3D structure, or 3D CNN-based approaches, which require large amounts of memory and computation. More recently, PointNet-based methods that process the 3D point cloud use a sparse point cloud to reduce computation at the cost of performance. Previous diffusion-based models rely on global features and overlook local details. Also, they are permutation-equivariant, which limits their ability to distinguish between joints.
- Method: The depth map and 3D point cloud are processed separately to generate 2D and 3D features. They are then concatenated and passed through a three-layer bias-induced layer (BIL) to generate joint-wise embeddings. The denoiser starts with a randomly sampled set of joint coordinates from a Gaussian distribution, and then for each time step out of a set amount, the denoiser samples local features around each joint and refines the joint coordinates using a kinematic-aware GCN.
- Limitations: The model can't process interacting hands and focuses on single-hand samples. Also, the denoiser's performance plateaus after around 5 timesteps.
- Paper: https://arxiv.org/pdf/2404.03159
- GitHub: https://github.com/cwc1260/HandDiff

HOISDF

- Conference: CVPR 2024
- Authors: Haozhe Qi, Chen Zhao, Mathieu Salzmann, Alexander Mathis
- Hand or Body: Hand
- Input: RGB image
- Motivation: Direct lifting methods do not utilize 3D intermediate representations and rely on the network to learn the mapping from 2D image to 3D pose. Coarse-to-fine methods make initial predictions and then refine them with explicit intermediate representations, such as hand joints or vertices, but require more computation and is not as precise as implicit intermediate representations, such as a signed distance field as a continuous function. Also, current signed distance field applications mainly use them to directly reconstruct 3D meshes instead of as an intermediate representation.
- Method: This method estimates signed distance field for both the hand and object to build the shape of each. The 3D space is split into many bins, each with a query point, and the ones on the surface, which have the lowest distance to the surface, are then used to extract hand and object features. These features are enhanced with multi head self-attention layers and then used to compute the 3D hand joints and mesh, as well as the object rotation and translation vectors. Each joint's 3D position is fine-tuned with the distance vectors from each nearby query point.
- Limitations: The hand and object meshes might intersect with each other in severely occluded scenarios. Also, this model was only trained on objects that are bigger than the hand, which are much less occluded than smaller objects.
- Paper: https://arxiv.org/pdf/2402.17062
- GitHub: https://github.com/amathislab/HOISDF

WildHands

- Conference: ECCV 2024
- Authors: Aditya Prakash, Ruisen Tu, Matthew Chang, Saurabh Gupta
- Hand or Body: Hand
- Input: RGB image
- Motivation: No existing method performs well in egocentric views as the perspective is distorted. Also, there is a lack of egocentric 3D annotated datasets outside of labs.
- Method: The image is cropped for hands to focus on fine-grained details. The hand is passed through a ResNet50 backbone to generate 7 feature maps, and then the maps are averaged to compute global image features. Intrinsics-aware positional encoding, which consists of sinusoidal functions, is incorporated to resolve the perspective distortion. The MANO parameters are initialized at 0 and then each of the parameters are passed through a 3-layer MLP with the feature vector and then added back to the parameters, three times. The result is the final MANO prediction.
- Limitations: The model does not perform well on images where the fingers are barely visible or contain complex poses. The KPE encoding requires camera intrinsic parameters. Also, they manually set hyperparameters for loss term weights.
- Paper: https://arxiv.org/pdf/2312.06583
- GitHub: https://github.com/ap229997/hands

NC-RetNet

- Conference: ECCV 2024
- Authors: Kaili Zheng, Feixiang Lu, Yihao Lv, Liangjun Zhang, Chenyi Guo, Ji Wu
- Hand or Body: Body
- Input: 2D Pose frames in 17-kpt format
- Motivation: Most previous methods lift 2D keypoints to 3D but it is not very accurate as one 2D detection may correspond to multiple 3D skeletons. Previous methods that process videos and extract temporal information, process the frames with a sliding-window that treats past and future frames equally, but a larger chunk size means relying on more future frames before inference, which increases inference latency.
- Method: This model is built off the RetNet architecture but with non-causal masking to incorporate future frames. They implement a chunkwise recurrent representation of RetNet, where the video sequence is split into chunks and the chunks are processed in parallel but the frames in the chunk are processed recurrently. This includes a cross-chunk state that tracks long-term features. Also, Rotary Position Encoding (RPE), a type of relative positional encoding, handles temporal relationships between frames in the attention mechanism of RetNet. They train on larger chunks to capture more temporal relationships but then transfer knowledge from large chunks to smaller chunks at inference to reduce latency.
- Limitations: While, the paper examines the effect of the cross-chunk state in transferring knowledge, The theory behind how the method is able to transfer knowledge is unclear. Also, the method has only been tested for 2D-to-3D lifting and the method's application to other 3d pose estimation methods or sequential data tasks have not been explored.
- Paper: https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/04820.pdf
- GitHub: https://github.com/Kelly510/PoseRetNet

PAFUSE

- Conference: ECCV 2024
- Authors: Nermin Samet, Cédric Rommel, David Picard, Eduardo Valle
- Hand or Body: Hand and Body
- Input: 2D Pose frames in 133-kpt format
- Motivation: Current methods process all keypoints in a single network and are not designed to adapt to different scale and motion variance across body parts like hand, face, and main body. This is especially prevalent in videos or sequential frames where temporal constraints vary across different body parts.
- Method: This method lifts 2D keypoints to 3D and processes the hands, face, and body with separate networks so each network can better adapt to that body part. Each network is a denoising diffusion probabilistic model (DDPM) to predict 3D pose from the 2D keypoints. During inference, the model starts with pure Gaussian noise and processes it with the 2D keypoints and the timestep to slowly denoise it at each step into the predicted 3D keypoints. The DDPMs process a temporal sliding window to ensure that temporal information is incorporated. During training, the model starts with noisy ground-truth 3D keypoints.
- Limitations: The model was only trained and evaluated on the H3WB dataset as it was the only dataset to have 3D pose labels and sequential frames for temporal data. They did qualitatively evaluate the model in in-the-wild scenarios but the only quantitative data is from the H3WB dataset. Also, the frames in the H3WB dataset were irregularly sampled with large gaps so longer windows did not perform well but could on a consistently-sampled dataset.
- Paper: https://arxiv.org/pdf/2407.10220
- GitHub: https://github.com/valeoai/PAFUSE

TokenHMR: Advancing Human Mesh Recovery with a Tokenized Pose Representation

- Conference: CVPR 2024
- Authors: Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, Yao Feng, Michael J. Black
- Hand or Body: Body
- Input: RGB image
- Motivation: Current models are only able to either align the pose with the image or determine accurate 3D pose coordinates. The more accurate the method is at fitting 2D keypoints, the less accurate it is at predicting 3D pose, and vice versa. They determined the problem to be inaccurate camera parameters and biases in pseudo-ground truth datasets, which results in the detected 2D pose and the projected 3D joints not aligning.
- Method: They first introduce a loss function that penalizes large 2D and p-GT errors while minimally penalizing smaller ones, to reduce overfitting. A Vision Transformer backbone processes the image into feature tokens. They pretrain a Vector Quantized Variational Autoencoder (VQ-VAE) with large motion capture datasets. They discretize the human poses using the autoencoder to convert pose estimation from continuous regression to token classification, which restricts predictions to valid poses. The codebook serves as a vocabulary of valid poses, to ensure the model outputs realistic human poses.
- Limitations: The discretization of pose lowered the loss of 3D accuracy by about 2.5 mm.
- Paper: https://arxiv.org/pdf/2404.16752
- GitHub: https://github.com/saidwivedi/TokenHMR

Person-in-WiFi 3D: End-to-End Multi-Person 3D Pose Estimation with Wi-Fi

- Conference: CVPR 2024
- Authors: Kangwei Yan, Fei Wang, Bo Qian, Han Ding, Jinsong Han, Xing Wei
- Hand or Body: Body
- Input: WiFi signals
- Motivation: Camera-based pose estimation rely on proper lighting conditions and field of view, and they struggle in severe occlusion scenarios. Wi-Fi methods perform well under occlusion and do not capture sensitive details, but there has not been a method for multi-person 3D pose estimation.
- Method: Through the WiFi signals, Channel State Information (CSI) samples are collected, containing amplitude and phase distortions caused by human motion. A linear transformation is applied to denoise the CSI phases. The CSI samples are then split into tokens, which represent the subcarriers that are being sent between the devices. The tokens are embedded with spatial-temporal embeddings and processed through six Transformer encoder layers. The Pose decoder starts with 100 randomly initialized queries that each represents a hypothesis for a person's pose and contains three DETR layers. The pose is initialized from the most confident outputs of the encoder, and then passed through each DETR layer to be refined. The Refine Decoder performs the same as the Pose Decoder but starts on the high confident pose predictions from the Pose decoder rather than 100 randomly initialized queries and is meant for fine-tuning.
- Limitations: The model is trained with annotations generated by the Azure Kinect SDK, which limits the performance upper bound. The method does not implement any cross-location generalization methods and is sensitive to spatial configurations and the environment. The dataset does not have diverse arm positions, which resulted with the arms having much lower accuracy, especially the hands.
- Paper: https://openaccess.thecvf.com/content/CVPR2024/papers/Yan_Person-in-WiFi_3D_End-to-End_Multi-Person_3D_Pose_Estimation_with_Wi-Fi_CVPR_2024_paper.pdf
- GitHub: https://github.com/aiotgroup/Person-in-WiFi-3D-repo

RePOSE: 3D Human Pose Estimation via Spatio-Temporal Depth Relational Consistency

- Conference: ECCV 2024
- Authors: Ziming Sun, Yuan Liang, Zejun Ma, Tianle Zhang, Linchao Bao, Guiqing Li, and Shengfeng He
- Hand or Body: Body
- Input: 2D Pose frames in 17-kpt format
- Motivation: Current pose estimation methods struggle with occlusions because they rely on absolute depth, while this paper introduces a method that uses relative depth. Also, 2D-to-3D pose lifting suffers a one-to-many issue where one 2D pose can be projected to multiple valid 3D poses.
- Method: This transformer-based method is designed to incorporate spatial and temporal information for lifting from 2D to 3D pose. The 2D pose skeletons on predicted with an off-the-shelf network and then passed through a fully connected layer. Spatial positional and temporal positional encodings are applied to the resulting vector. It is passed through a Spatial Transformer and a Temporal Transformer independently and the outputs are combined using Adaptive Fusion. The result is passed through another fully connected layer and a regression network to predict the 3D pose. They include loss terms for both spatial and temporal depth consistency, where spatial loss is determined from the differences in depth across different joints in the same frame, while temporal loss is determined from the difference in depth in the same joint across different frames.
- Limitations: This model just lifts 2D pose to 3D so its performance depends on the accuracy of the 2D pose keypoints. Also, the model's accuracy decreases in complex body positions, interactions with objects, or low-light environments.
- Paper: https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/02925.pdf
- GitHub: https://github.com/StupidAdam/RePOSE