HandDAGT: A Denoising Adaptive Graph Transformer for 3D Hand Pose Estimation

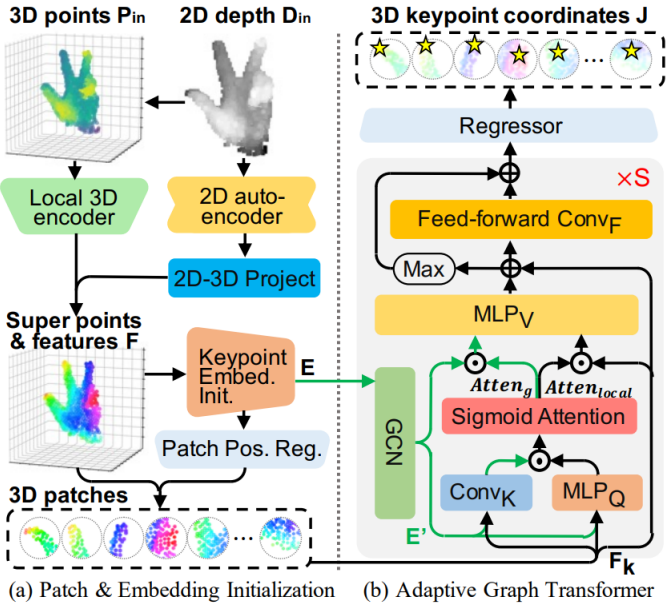
Wencan Cheng, Eunji Kim, Jong Hwan Ko

Accepted for ECCV 2024 (Paper) (GitHub)

This paper proposes a transformer-based architecture that uses the image depth map and hand point cloud to predict the 3D hand joint locations. The depth map and point cloud are used to generate keypoint embeddings and 3D local patches, which are then passed to the graph transformer as queries and keys to determine the hand keypoints. Also, random noise is incorporated in the training process to improve the model's noise and occlusion handling.

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

1. **Super Points Generation**

   A convolutional layer is used to extract a subsampled 3D super point set from the points in the hand point cloud. The depth is fed into a ConvNeXt-based autoencoder to generate a 2D local visual feature map. The 2D point features are then projected to 3D and concatenated to the super points.

2. **Initialization of Keypoint Embedding and Patch Position**

   The super points are fed through two convolutional layers to generate 3D global vectors. The 3D global vectors are then replicated J times and then passed through a three-layer bias-induced layer to generate the J keypoint embeddings. The embeddings are linear transformed to project them into the 3D space as the initial keypoint locations. The K-nearest super points around each point form its 3D point patch.

3. **Adaptive Graph Transformer Decoder**

   The keypoint embeddings are refined using a GCN to incorporate kinematic relationships between keypoints. The 3D patches and enhanced keypoint embeddings are passed through the sigmoid attention mechanism to dynamically balance local attention for visible keypoints and kinematic attention for occluded ones. The updated embeddings are aggregated using residual connections (to ensure original geometric details are preserved) and a point set convolution, and then a linear transformation projects them to the 3D keypoint coordinates.

Method	Mean keypoint error (mm)		Input
	ICVL	NYU	
Ren-9x6x6 [17]	7.31	12.69	D
DeepPrior++ [32]	8.1	12.24	D
Pose-Ren [4]	6.79	11.81	D
DenseReg [44]	7.3	10.2	D
CrossInfoNet [10]	6.73	10.08	D
JGR-P2O [11]	6.02	8.29	D
SSRN [37]	6.01	7.37	D
PHG [36]	5.97	7.39	D
HandPointNet [13]	6.94	10.54	P
Hand-Transformer [21]	6.47	9.80	P
Point-to-Point [16]	6.3	9.10	P
V2V [31]	6.28	8.42	V
HandFolding [8]	5.95	8.58	P
HandR2N2 [6]	5.70	7.27	P
IPNet [35]	5.76	7.17	D+P
HandDAGT (Ours)	**5.66**	**7.12**	D+P

Method	Mean keypoint error (mm)					Input
	S0	S1	S2	S3	AVG	
A2J [47]	23.93	25.57	27.65	24.92	25.52	D
Spurr et al. [39]	17.34	22.26	25.49	18.44	18.44	RGB
METRO [25]	15.24	-	-	-	-	RGB
Tse et al. [42]	16.05	21.22	27.01	17.93	20.55	RGB
HandOcc [33]	14.04	-	-	-	-	RGB
IPNet [35]	8.03	9.01	8.60	7.80	8.36	D+P
HandDAGT (Ours)	**7.72**	**8.68**	**8.22**	**7.52**	**8.03**	D+P

Limitations

1. The model can't process interacting hands (multi-hand interaction)
2. 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.