Spectral Graphormer: Spectral Graph-based Transformer for Egocentric Two-Hand Reconstruction using Multi-View Color Images Supplementary material

Tze Ho Elden Tse[1,2*] Franziska Mueller[1] Zhengyang Shen[1] Danhang Tang[1] Thabo Beeler[1]
Mingsong Dou[1] Yinda Zhang[1] Sasa Petrovic[1] Hyung Jin Chang[2] Jonathan Taylor[1] Bardia Doosti[1]
[1]Google [2]University of Birmingham

In this supplemental document, we provide:

- a summary of existing RGB hand pose estimation benchmarks (Sec 1);
- implementation details for networks and spectral filters (Sec 2);
- additional results and analysis (Sec 3);
- limitations of our method (Sec 4);
- additional qualitative examples (Sec 5).

1. Overview of existing datasets

Table 1 presents a summary of existing datasets for hand pose estimation. As existing egocentric datasets are either instrumented with visible markers [1] or having limited background variation [5], both of our synthetic and real datasets contain densely annotated two-hand with forearms.

2. Implementation details

In this section, we detail the network architectures in Section 2.1 and provide implementation details of spectral filters in Section 2.2.

2.1. Network architectures

Our proposed method. We provide in Table 2 the full details of our network architecture. We use ResNet-50 [4] as the backbone of our network. The input to our network model is N number of multi-view $224 \times 224 \times 3$ RGB images and the output are two hand meshes with 4023 vertices each hand.

METRO baseline. We extend METRO [7] from single-view setting to multi-view and detail the network architecture in Table 3. Each transformer encoder block has 4 layers and 3 attention heads. Other than the size of feature dimension, the hyperparameters for transformer encoders are consistent across our proposed method and this baseline.

*This work was done during an internship at Google.

Parametric baseline. We detail the network architecture of our parametric baseline in Table 4. The output of the network is passed to a pre-computed parametric hand model to produce hand mesh with 4023 vertices.

2.2. Implementation details of spectral filters

We perform spectral filtering by applying a customised filter to the spectral representation of the graph signal. The spectral representation depends on the adjacency matrix and the eigenvectors of the graph. Spectral filtering is performed by applying a filter function to the spectral coefficients. In the following, we detail Gaussian and Laplacian filters.

Given eigenvalue λ, the Gaussian filter function f_{gau} can be described as:

$$f_{gau}(\lambda) = e^{\frac{-\lambda^2}{2\sigma^2}}, \tag{1}$$

where σ is the standard deviation of the Gaussian distribution. We set $\sigma = 0.5$ in our experiments. Similarly, the Laplacian filter function f_{lap} can be described as:

$$f_{lap}(\lambda) = \lambda^{-\frac{1}{2}}. \tag{2}$$

As eigenvalues can have zero and negative values, the inverse square root computation can result in NaN (not a number) values. Therefore, we put a tolerance value that is close to zero to avoid this.

3. Additional analysis

3.1. Influence of different loss terms

In Table 5, we analyse the influence of different loss terms. In these experiments, we consider L1 losses on 3D mesh vertices and 2D re-projection loss, *i.e.* $\mathcal{L}_{mesh}$ and $\mathcal{L}_{2D}$, respectively. In addition, we experiment with mean squared euclidean distance loss $\mathcal{L}_{MSE}$ on hand mesh. For mesh regularisation, we apply edge length regularisation $\mathcal{L}_{edge}$ and minimise the Chamfer distances $\mathcal{L}_{cham}$. We find that the combination of $\mathcal{L}_{mesh}$, $\mathcal{L}_{2D}$ and $\mathcal{L}_{edge}$ delivers the optimal results.

Table 1: A comparison of existing RGB hand pose estimation benchmarks.

	# Frames	Two-hand	Egocentric	Markerless	Background variation	Dense annotation
FPHA [1]	105k	✗	✓	✗	✗	✗
FreiHAND [11]	37k	✗	✗	✓	✗	✗
InterHand2.6M [8]	2.6M	✓	✗	✓	✗	✗
H2O [5]	571k	✓	✓	✓	✗	✗
H_2O-3D [2]	76k	✓	✗	✓	✗	✗
Ego3DHands [6]	55k	✓	✓	✓	✓	✗
Ours (synthetic)	1M	✓	✓	✓	✓	✓
Ours (real)	61k	✓	✓	✓	✗	✓

3.2. Analysis on mesh refinement at inference

For quantitative evaluation, we followed [3, 9, 10] to include penetration depth (mm) and intersection volume (cm^3). Penetration depth refers to the maximum distances from hand mesh vertices to the other/self hand's surface when in a collision. Intersection volume is obtained by voxelising the meshes using a voxel size of $0.5cm$. We report the results in Table 6. These results demonstrate the robustness of our optimisation-based mesh refinement strategy.

3.3. Complete results for model compression

We provide complete experimental results for model compression in Table 7. We find that the feature channel size C cannot drop under 64 as performance drops massively. Therefore, we fix $C = 64$ and reduce the number of vertices for template hand mesh V'.

3.4. Complete results for multi-view fusion

We provide complete experimental results for different multi-view fusion strategies in Table 8. We find that the combination of soft-attention fusion and mesh segmentation consistently improves the performance.

3.5. Illustration of different fusion strategies

We illustrate two fusion strategies in Fig. 2 which were considered for Table 3 in the main text. In Fig. 2a), as max pooling is applied on each of the input image separately, the resulting features are in size of $[N \times 2048]$. On the other hand, in Fig. 2b), only one global feature vector is computed per data sample by applying max pool across multi-view input images.

3.6. Additional quantitative comparison on real dataset

We provide additional quantitative comparison with METRO [7] using our real dataset in Table 9. In these experiments, we vary the number of input camera views for training and withheld 3 unseen camera views for evaluation. Our method consistently outperforms our METRO baseline across different number of input views. This shows the generalisation ability of our proposed method.

4. Method limitations

Though our method results in accurate and physically-plausible high fidelity two-hand reconstructions, the results are sometimes not plausible when self-penetration is highly complex (see Fig. 1). We believe this problem can be tackled in the future by incorporating temporal information and more advance physical modeling into our proposed framework. In particular, the current optimisation-based mesh refinement works on self-penetrations only. Solving interpenetrations during hand-hand interactions is in-line with our future goal.

5. Additional examples

We provide additional qualitative examples on our synthetic dataset and real dataset in Fig. 3 and Fig. 4, respectively.

Table 2: Architecture of our network. B refers to batch size and N refers to the number of multi-view RGB input images. Note that the duplicated layer numbers are performed in parallel and the output of layer 13 is concatenated with template hand mesh 804×3 before feeding to transformer encoder in layer 14.

Layer	Operation	Dimensionality
	Input	$B \times N \times 224 \times 224 \times 3$
1	ResNet-50	$B \times N \times 7 \times 7 \times 2048$
2	Upsampling 2D	$B \times N \times 14 \times 14 \times 2048$
3	Convolution 2D	$B \times N \times 12 \times 12 \times 256$
4	Batch normalisation	$B \times N \times 12 \times 12 \times 256$
5	ReLU	$B \times N \times 12 \times 12 \times 256$
6	Upsampling 2D	$B \times N \times 24 \times 24 \times 256$
7	Convolution 2D	$B \times N \times 22 \times 22 \times 256$
8	Batch normalisation	$B \times N \times 22 \times 22 \times 256$
9	ReLU	$B \times N \times 22 \times 22 \times 256$
2	Upsampling 2D	$B \times N \times 14 \times 14 \times 2048$
3	Convolution 2D	$B \times N \times 12 \times 12 \times 256$
4	Batch normalisation	$B \times N \times 12 \times 12 \times 256$
5	ReLU	$B \times N \times 12 \times 12 \times 256$
6	Upsampling 2D	$B \times N \times 24 \times 24 \times 256$
7	Convolution 2D	$B \times N \times 22 \times 22 \times 256$
8	Batch normalisation	$B \times N \times 22 \times 22 \times 256$
9	ReLU	$B \times N \times 22 \times 22 \times 256$
10	Convolution 1D	$B \times N \times 22 \times 22 \times 7$
11	Spatial softmax	$B \times N \times 22 \times 22 \times 7$
12	Matrix multiplication	$B \times N \times 7 \times 256$
13	Max pooling	$B \times 7 \times 256$
14	Transformer encoder	$B \times 804 \times 259$
15	Fully-connected layer	$B \times 804 \times 130$
16	Transformer encoder	$B \times 804 \times 130$
17	Fully-connected layer	$B \times 804 \times 65$
18	Transformer encoder	$B \times 804 \times 65$
19	Fully-connected layer	$B \times 804 \times 3$
20	Fully-connected layer	$B \times 617 \times 3$
21	Spectral filtering	$B \times 617 \times 3$
22	Fully-connected layer	$B \times 1234 \times 3$
23	Spectral filtering	$B \times 1234 \times 3$
24	Fully-connected layer	$B \times 2468 \times 3$
25	Spectral filtering	$B \times 2468 \times 3$
26	Fully-connected layer	$B \times 4023 \times 3$
20	Fully-connected layer	$B \times 617 \times 3$
21	Spectral filtering	$B \times 617 \times 3$
22	Fully-connected layer	$B \times 1234 \times 3$
23	Spectral filtering	$B \times 1234 \times 3$
24	Fully-connected layer	$B \times 2468 \times 3$
25	Spectral filtering	$B \times 2468 \times 3$
26	Fully-connected layer	$B \times 4023 \times 3$
	Output	$B \times 8046 \times 3$

Table 3: Architecture of METRO baseline.

Layer	Operation	Dimensionality
	Input	$B \times N \times 224 \times 224 \times 3$
1	ResNet-50	$B \times N \times 7 \times 7 \times 2048$
2	Max pooling 2D	$B \times 2048$
3	Transformer encoder	$B \times 804 \times 2051$
4	Fully-connected layer	$B \times 804 \times 1026$
5	Transformer encoder	$B \times 804 \times 1026$
6	Fully-connected layer	$B \times 804 \times 513$
7	Transformer encoder	$B \times 804 \times 513$
8	Fully-connected layer	$B \times 804 \times 256$
9	Fully-connected layer	$B \times 804 \times 3$
10	Fully-connected layer	$B \times 1624 \times 3$
11	Fully-connected layer	$B \times 4023 \times 3$
12	Fully-connected layer	$B \times 8046 \times 3$
	Output	$B \times 8046 \times 3$

Table 4: Architecture of parametric baseline.

Layer	Operation	Dimensionality
	Input	$B \times N \times 224 \times 224 \times 3$
1	ResNet-50	$B \times N \times 7 \times 7 \times 2048$
2	Max pooling 2D	$B \times 2048$
3	Fully-connected layer	$B \times 1024$
4	Fully-connected layer	$B \times 512$
5	Fully-connected layer	$B \times 200$
	Output	$B \times 200$

Table 5: Impact of different loss terms on our synthetic dataset. Hand errors are given in millimeters (mm).

$\mathcal{L}_{mesh}$	$\mathcal{L}_{2D}$	$\mathcal{L}_{MSE}$	$\mathcal{L}_{edge}$	$\mathcal{L}_{cham}$	Error
✓	✓				1.55
✓	✓	✓			1.48
✓	✓		✓		1.38
✓	✓	✓	✓		1.38
✓	✓	✓	✓	✓	1.49

Table 6: Quantitative evaluation on the impact of mesh refinement at inference.

	Before refinement	After refinement
Max. penetration (mm)	5.3	0.16
Intersection vol. (cm^3)	2.0	0.09

Table 7: Ablations of different backbones and hyperparameters. We denote P_{cnn} and P_{total} to be the number of parameters for CNN backbone and total model, respectively.

Backbone	P_{cnn}	V'	C	Error	P_{total}
ResNet-50	23.5M	804	256	1.38	58.3M
EfficientNet-B3	12.9M	804	256	2.53	47.7M
EfficientNet-B2	9.2M	804	256	2.55	44M
EfficientNet-B1	7.8M	804	256	2.74	42.6M
EfficientNet-B0	5.3M	804	256	2.85	40.1M
EfficientNet-B3	12.9M	804	128	3.00	40.6M
EfficientNet-B2	9.2M	804	128	3.20	36.9M
EfficientNet-B1	7.8M	804	128	3.20	35.5M
EfficientNet-B0	5.3M	804	128	2.91	33M
EfficientNet-B3	12.9M	804	64	4.00	38.4M
EfficientNet-B2	9.2M	804	64	3.87	34.7M
EfficientNet-B1	7.8M	804	64	4.04	33.3M
EfficientNet-B0	5.3M	804	64	4.31	30.8M
EfficientNet-B3	12.9M	804	32	89.7	37.3M
EfficientNet-B0	9.2M	804	32	90	29.7M
EfficientNet-B0	5.3M	160	256	4.12	42.8M
EfficientNet-B0	5.3M	160	128	4.96	37.8M
EfficientNet-B0	5.3M	80	64	6.89	34.2M

Table 8: Performance of different multi-view fusion strategies. We report hand error for both settings. K refers to the number of clusters for template hand mesh. Note that we do not include spectral filtering in the graph decoder here.

	Single-view	Multi-view
METRO [7]	10.87	-
METRO [7] + avg. pool	-	8.71
METRO [7] + max pool	-	7.09
Ours ($K = 1$)	-	6.59
Ours ($K = 2$)	-	5.71
Ours ($K = 3$)	-	5.19
Ours ($K = 4$)	-	4.79
Ours ($K = 5$)	-	4.59
Ours ($K = 6$)	-	5.28
Ours ($K = 7$)	-	**3.72**
Ours ($K = 8$)	-	3.79

Table 9: Error rates on our real dataset. Our method consistently outperforms METRO on different number of input views.

	5-views	3-views	2-views
METRO [7]	9.67	12.0	16.1
Ours	**4.34**	**6.94**	**7.73**

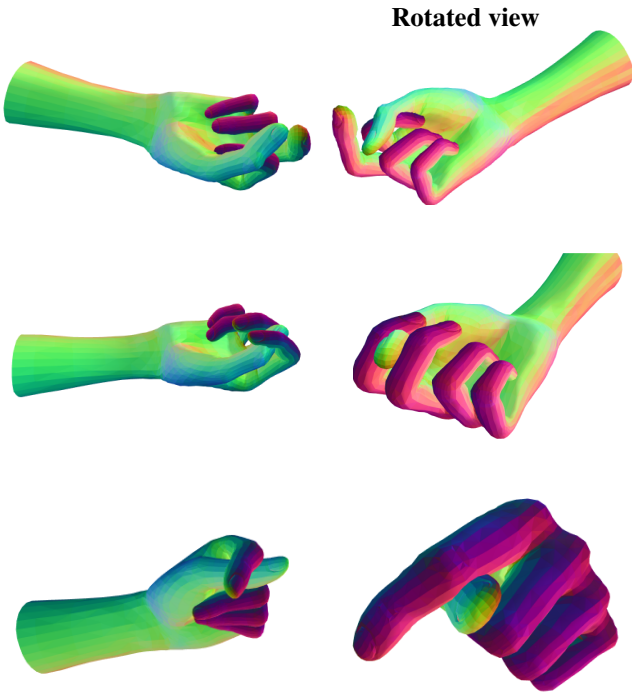

Figure 1: Failure examples for mesh refinement.

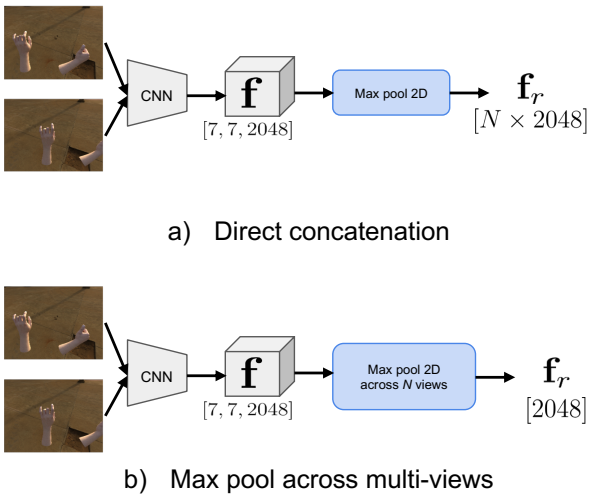

Figure 2: Illustration of different feature fusion strategies.

Figure 3: Additional qualitative examples on our synthetic dataset.

Figure 4: Additional qualitative examples on our real dataset.

References

[1] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 1, 2

[2] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint Transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *CVPR*, 2022. 2

[3] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[5] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *ICCV*, 2021. 1, 2

[6] Fanqing Lin, Connor Wilhelm, and Tony Martinez. Two-hand global 3D pose estimation using monocular RGB. In *WACV*, 2021. 2

[7] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 1, 2, 4

[8] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *ECCV*, 2020. 2

[9] Tze Ho Elden Tse, Kwang In Kim, Aleš Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *CVPR*, 2022. 2

[10] Tze Ho Elden Tse, Zhongqun Zhang, Kwang In Kim, Aleš Leonardis, Feng Zheng, and Hyung Jin Chang. S^2Contact: Graph-based network for 3D hand-object contact estimation with semi-supervised learning. In *ECCV*, 2022. 2

[11] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In *ICCV*, 2019. 2