Project-4: Hand Pose Estimation

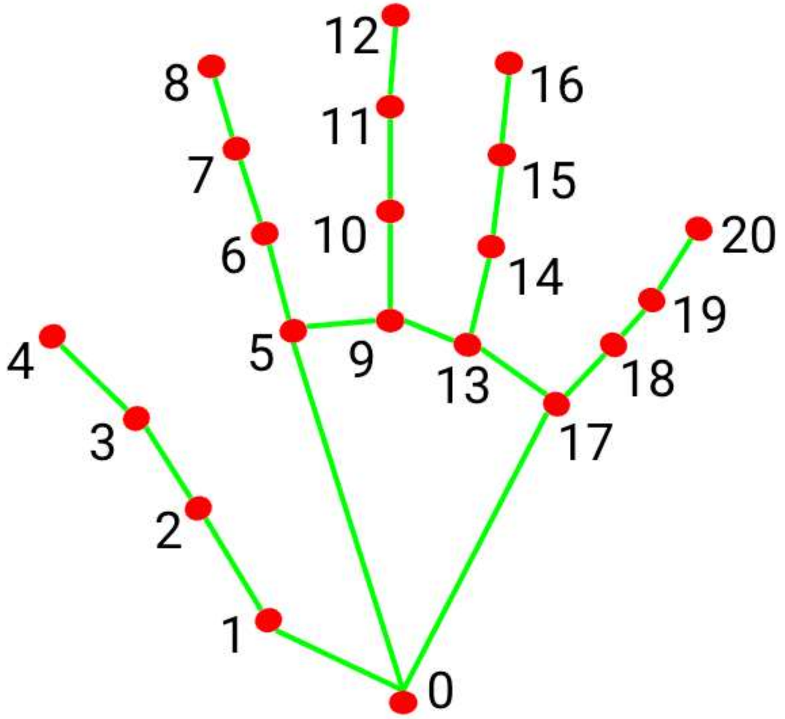
Yujiao Shi
SIST, ShanghaiTech
Autumn, 2024

Outline

- What is hand pose?
  - 2D keypoints
  - 3D keypoints
  - Mesh representation
- How to estimate hand pose?

2D Hand Keypoints

上海科技大学
ShanghaiTech University



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

2D Hand Keypoints

上海科技大学
ShanghaiTech University

How to estimate 2D Hand Keypoints?

上海科技大学
ShanghaiTech University

Input： RGB image

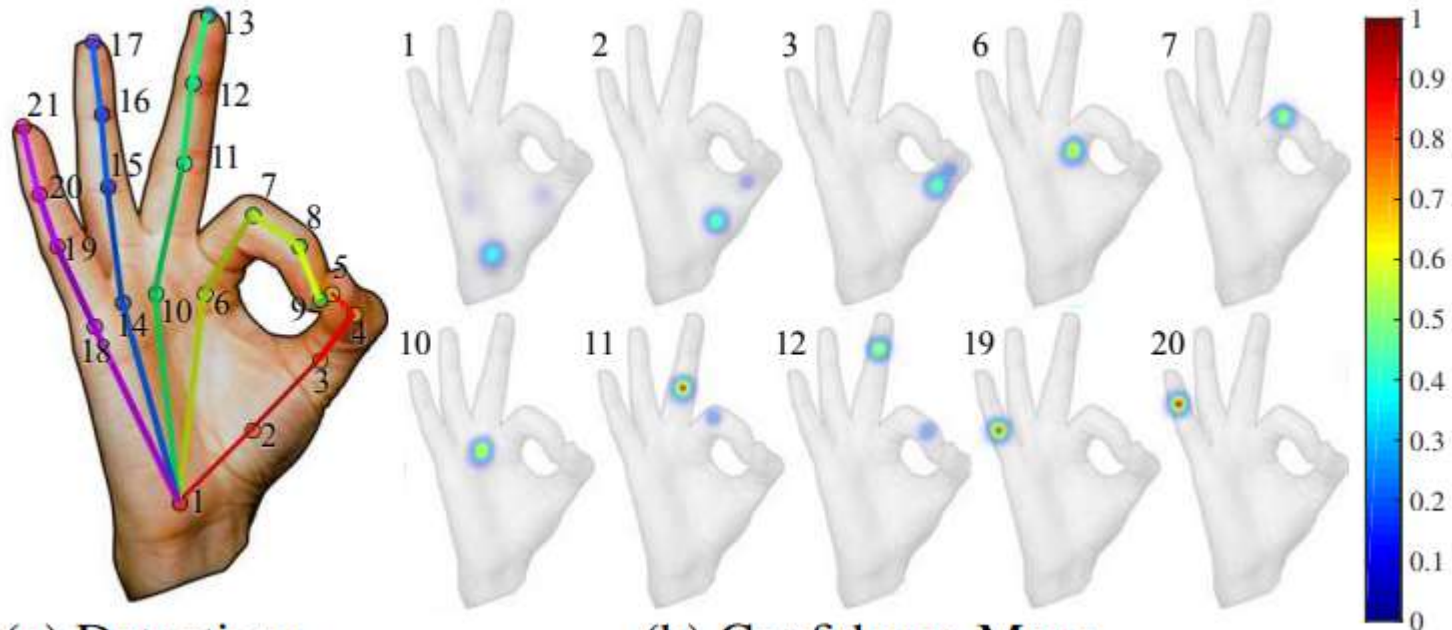
How to define output?

How to estimate 2D hand keypoints?

上海科技大学
ShanghaiTech University

Input

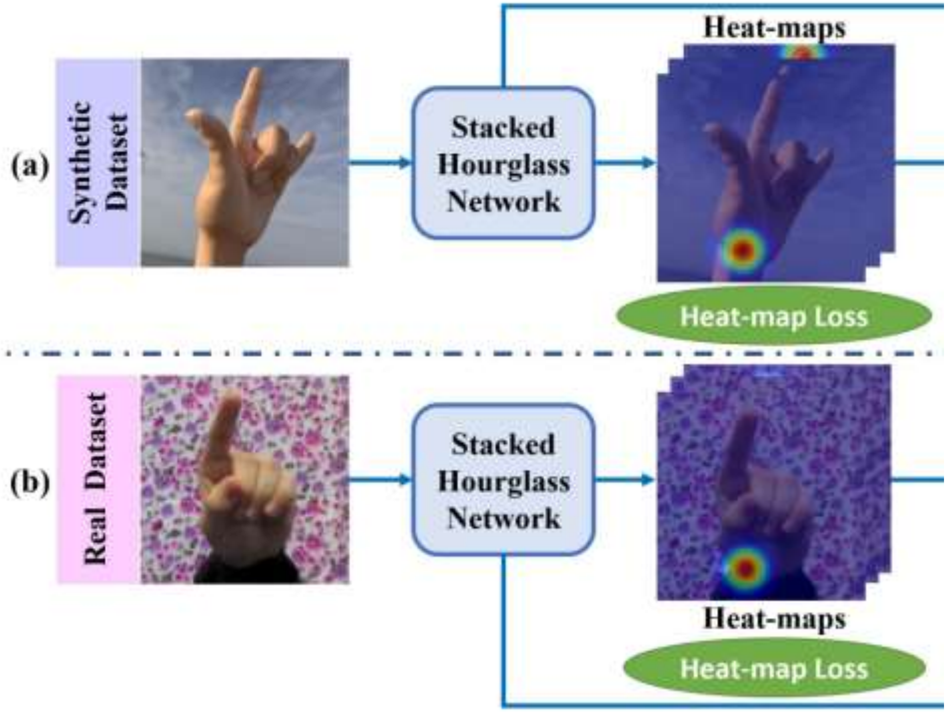
Output



(a) Detections

(b) Confidence Maps

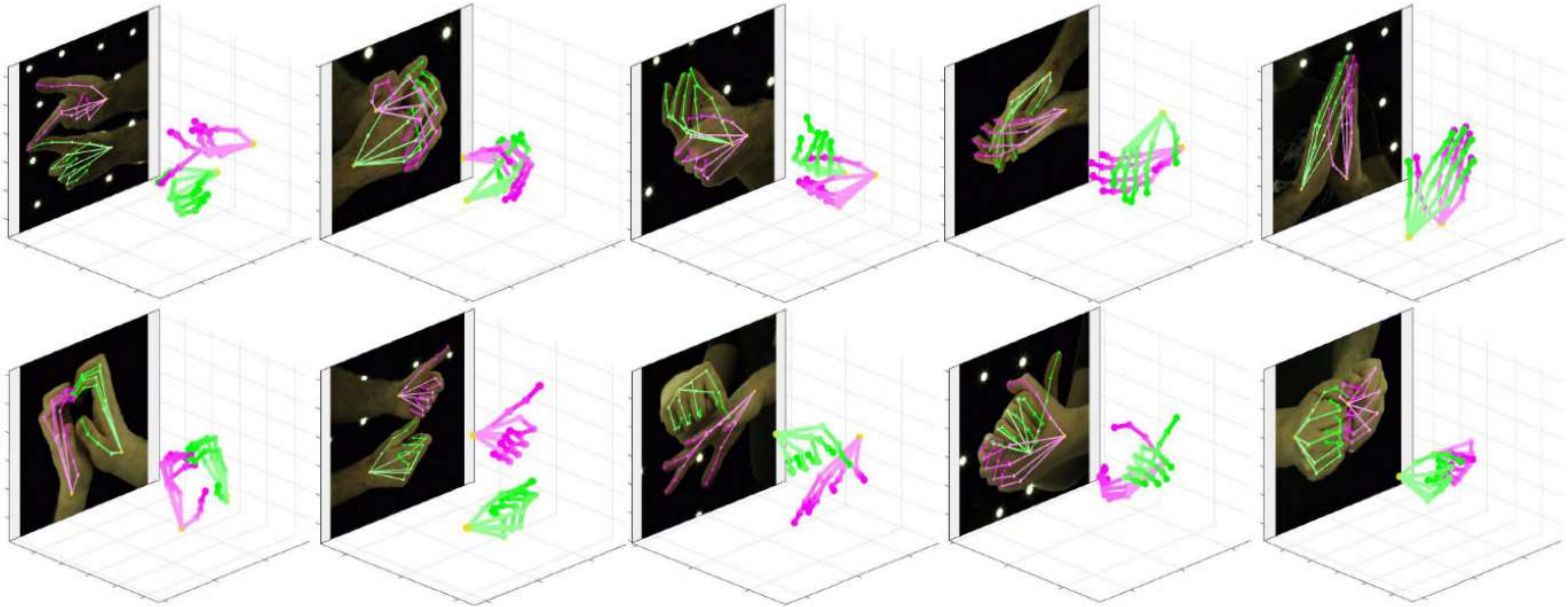
How will you design the network?



$$\mathcal{L}_{\mathcal{H}} = \sum_{j=1}^{J} \left\| \mathcal{H}_j - \hat{\mathcal{H}}_j \right\|_2^2$$

What are 3D Hand Keypoints?

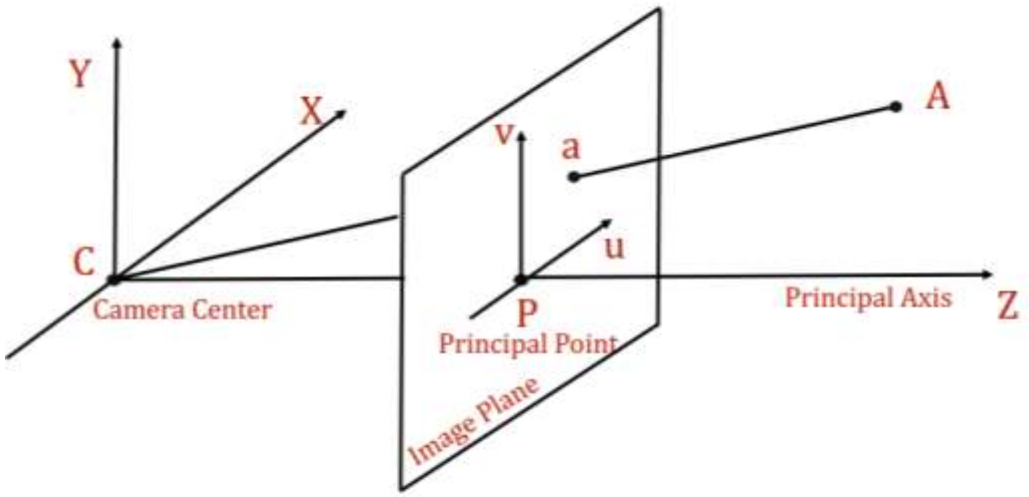
3D Hand Keypoints

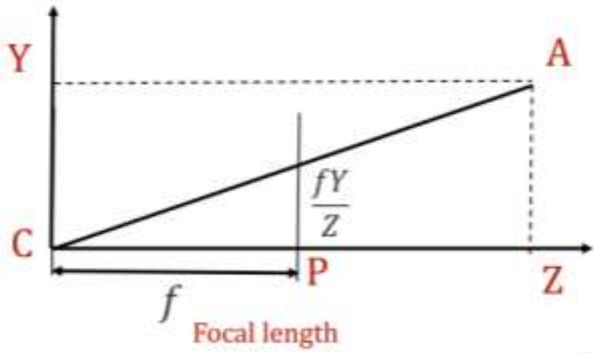


Perspective projection

Assumptions made:

- Principal point **at origin** of image plane
- Camera at center of world coordinates
- Square pixels





$$(X, Y, Z)^T \mapsto (u, v)^T = \left(\frac{fX}{Z}, \frac{fY}{Z}\right)^T$$

Euclidean Coordinates 3D World frame — Euclidean coordinates 2D Image plane

- How to write as linear mapping? (Homogeneous coordinates)

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

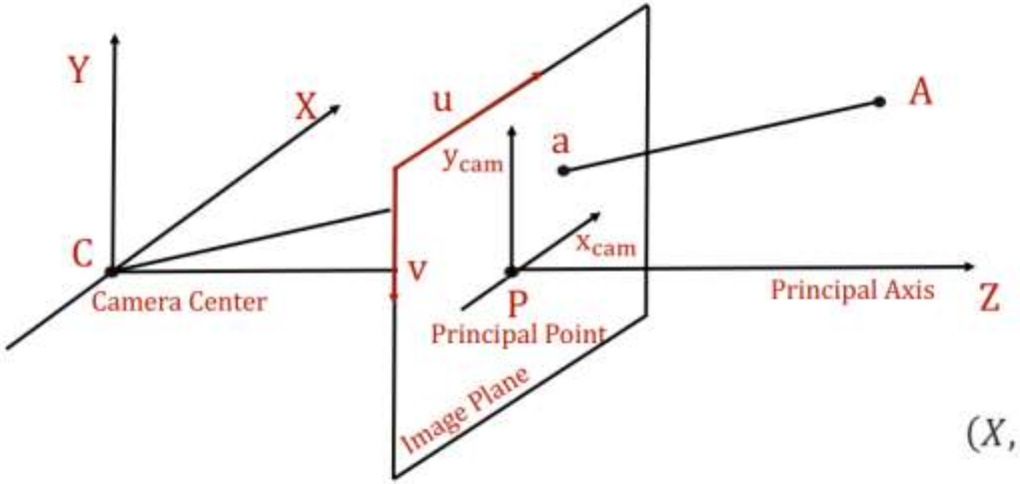
$$a = \mathbf{P}A$$

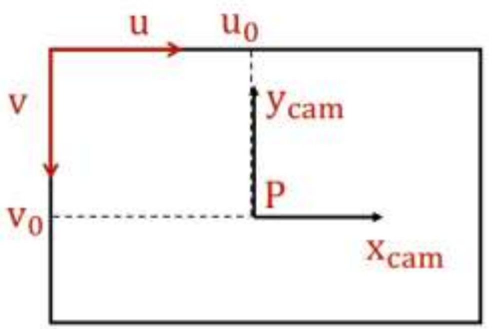
Camera projection matrix

Perspective projection

Assumptions made:

- Principal point at ~~origin~~ (u_0, v_0) of image plane
- Camera at center of world coordinates
- Square pixels





$$(X, Y, Z)^T \mapsto (u, v)^T = \left(\frac{fX}{Z} + u_0, \frac{fY}{Z} + v_0\right)^T$$

Euclidean Coordinates
3D World frame

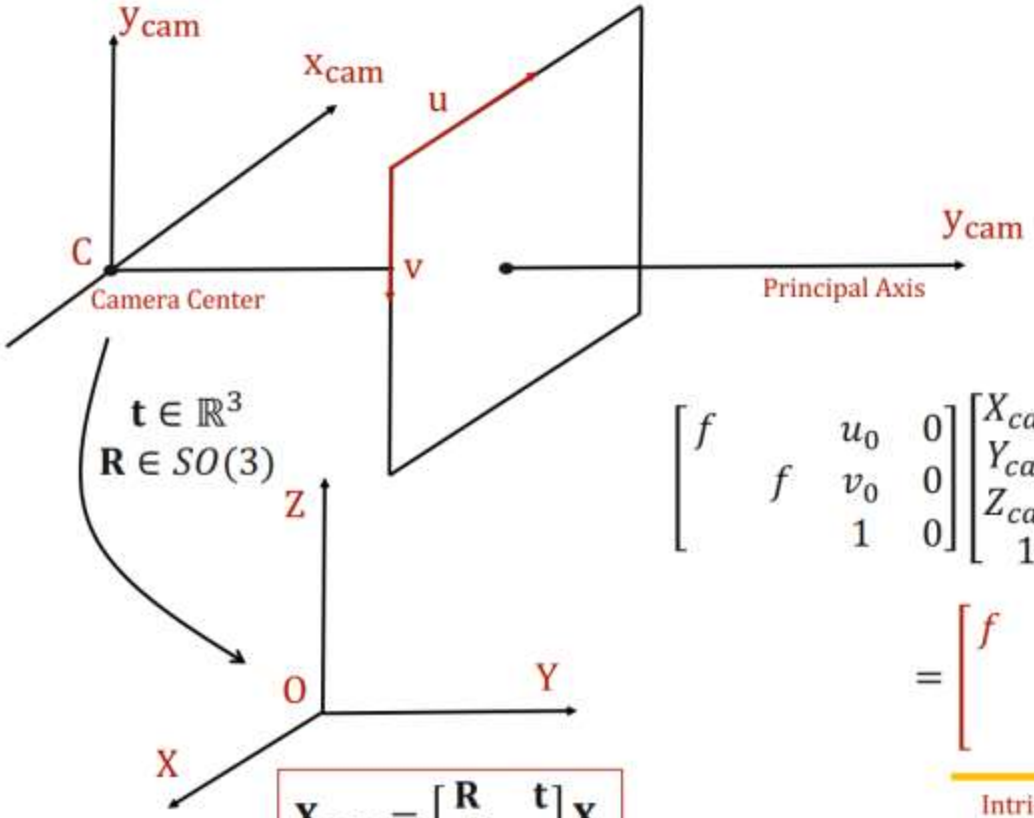
Euclidean coordinates
2D Image plane
(Note nonlinearity)

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX + Zu_0 \\ fY + Zy_0 \\ Z \end{bmatrix} = \begin{bmatrix} f & & u_0 & 0 \\ & f & v_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Perspective projection

Assumptions made:

- Principal point at ~~origin~~ (u_0, v_0) of image plane
- Camera at ~~center~~ $\mathbf{R}, \mathbf{t}$ of world coordinates
- Square pixels



$$\boxed{\mathbf{X}_{\mathbf{cam}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}}$$

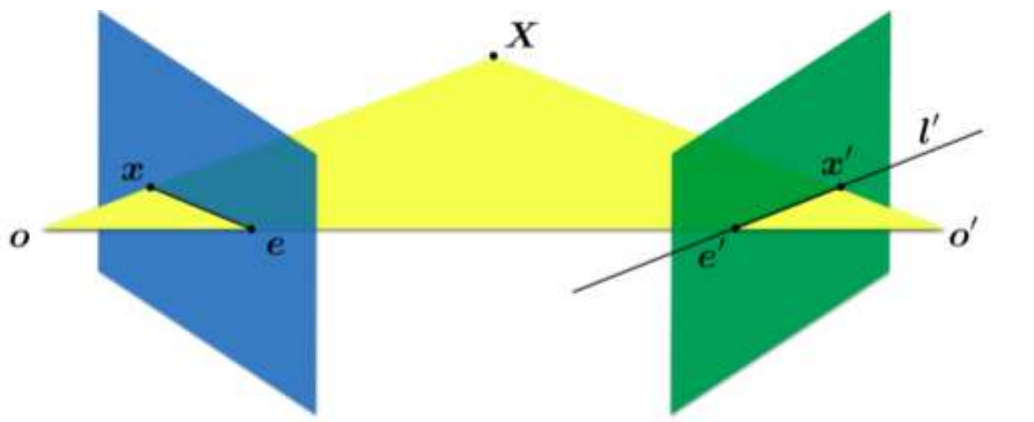
$$\begin{bmatrix} f & & u_0 & 0 \\ & f & v_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} f & & u_0 & 0 \\ & f & v_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}$$

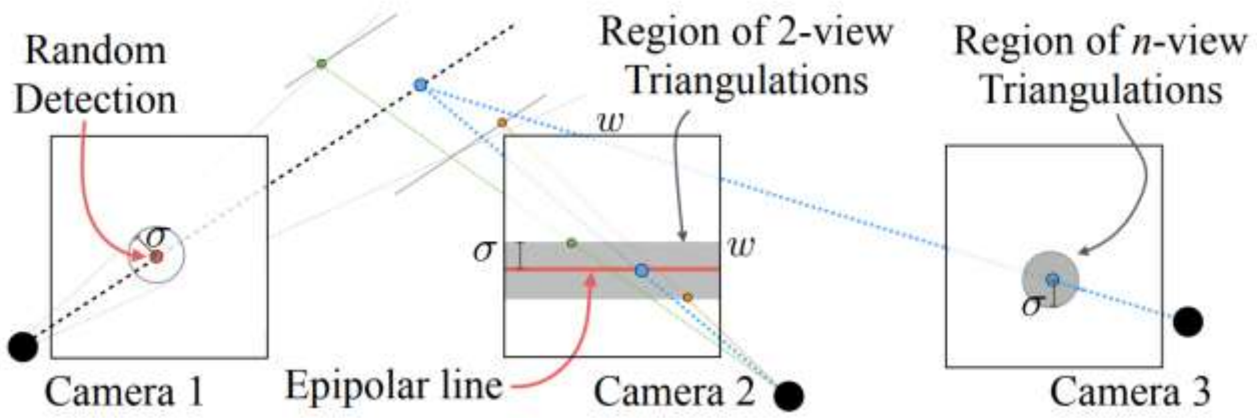
$$= \underbrace{\begin{bmatrix} f & & u_0 \\ & f & v_0 \\ & & 1 \end{bmatrix}}_{\text{Intrinsic matrix}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\text{Extrinsic matrix}} \mathbf{X}$$

How to estimate 3D hand keypoints?

- When depth map is available as input

- When depth map is not available as input during inference
  - Depth map is accessible in the training data
  - Multi-view images with relative poses are available

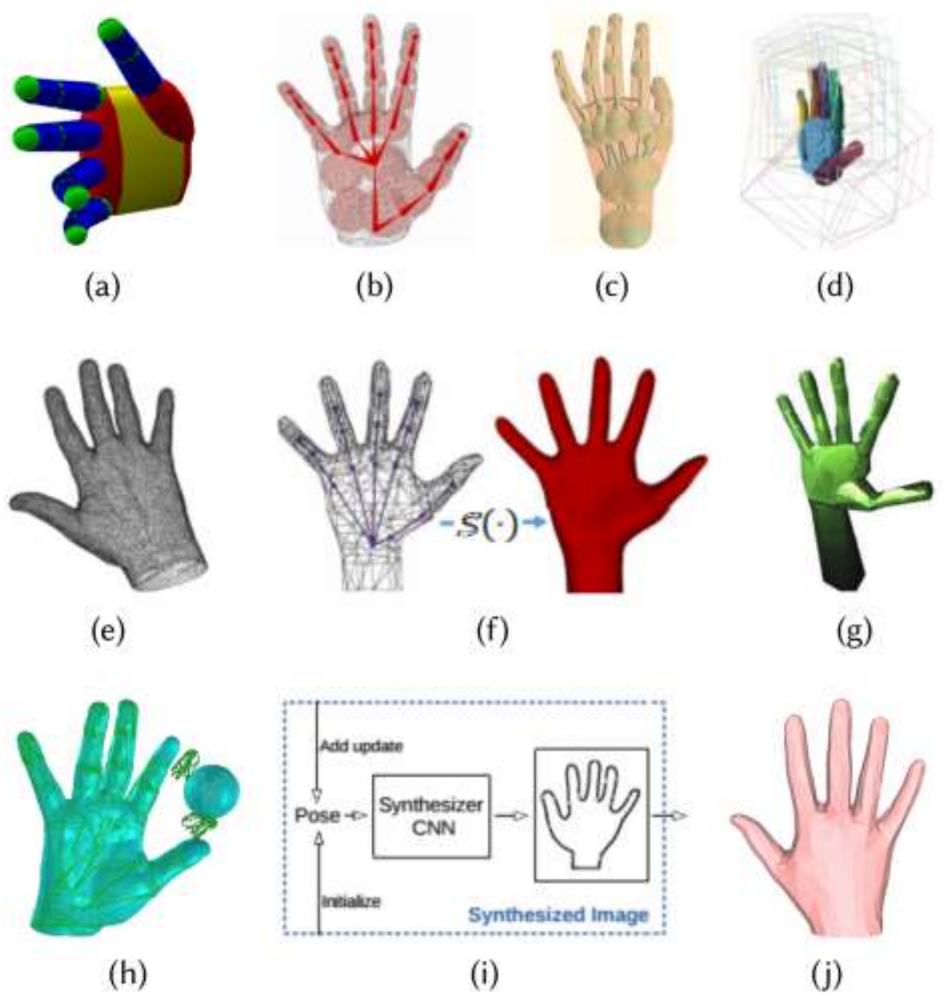




What is Hand Mesh?

Hand Models

上海科技大学
ShanghaiTech University



(a) Primitives approximation [Oikonomidis et al. 2011a]
(b) Sum-of-Gaussians model [Sridhar et al. 2013],
(c) Sphere-Meshes [Tkach et al. 2016] can be thought of as a generalization of the previous models,
(d) Articulated TSDF for a voxelized shape-primitive hand model [Schmidt et al. 2014]
(e) Triangular Mesh [Ballan et al. 2012; Tzionas et al. 2016]
(f) Loop Subdivision Surface of a triangular control mesh [Khamis et al. 2015]
(g) Convex Bodies for tracking [Melax et al. 2013]
(h) Convex Parts of a triangular mesh for contact point detection [Tzionas et al. 2016]
(i) Learned Model using a CNN to synthesize images of a given hand pose [Oberweger et al. 2015]
(j) MANO model.

Hand mano model

上海科技大学 ShanghaiTech University

Mano Model

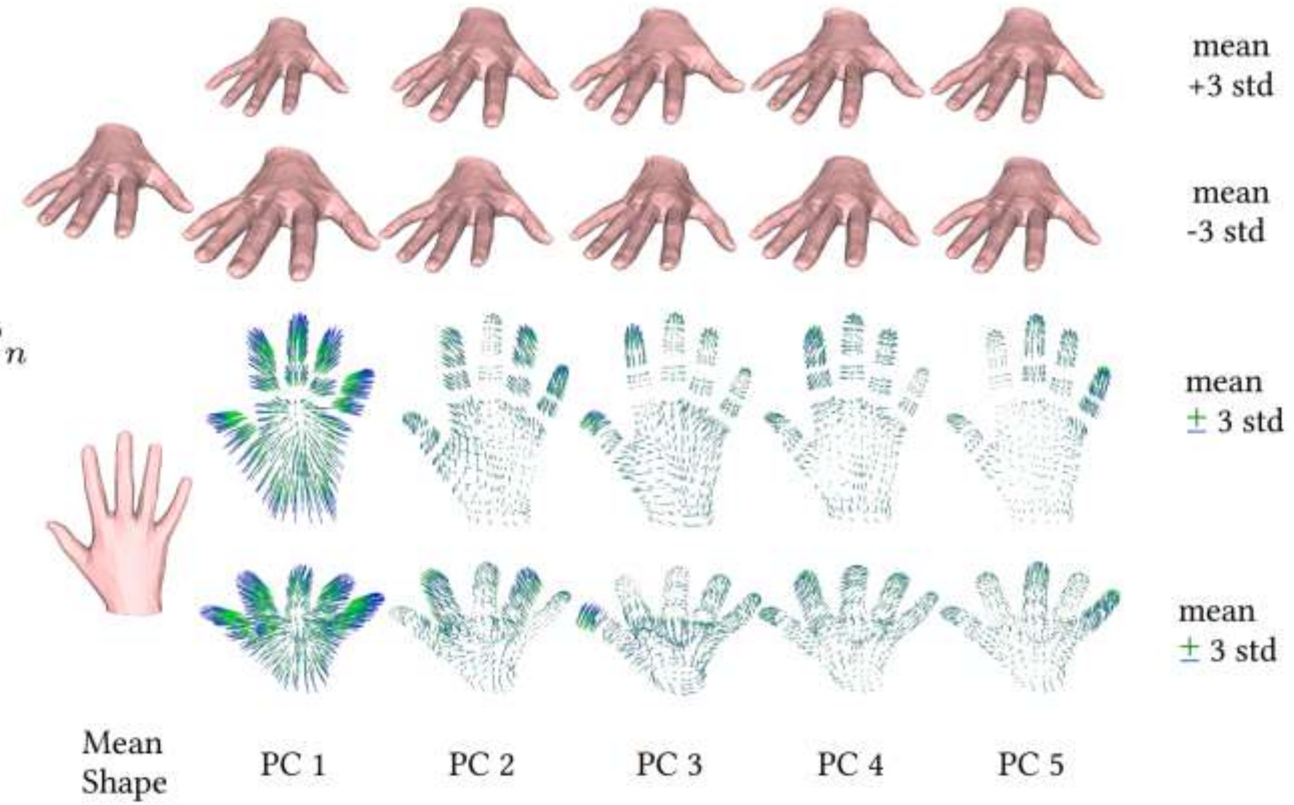
$$M(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(T(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathcal{W})$$

$$T(\boldsymbol{\beta}, \boldsymbol{\theta}) = \bar{\mathbf{T}} + \sum_{n=1}^{|\boldsymbol{\beta}|} \beta_n \mathbf{S}_n + \sum_{n=1}^{9K} (R_n(\boldsymbol{\theta}) - R_n(\boldsymbol{\theta}^*))\mathbf{P}_n$$

$$\boldsymbol{V}_h, \boldsymbol{P}_h = \mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}) + \boldsymbol{P}_{h,0}$$

$$\boldsymbol{P}_h \in \mathbb{R}^{21\times 3} \quad \boldsymbol{V}_h \in \mathbb{R}^{778\times 3}$$

$$\boldsymbol{P}_{h,0} \in \mathbb{R}^3$$

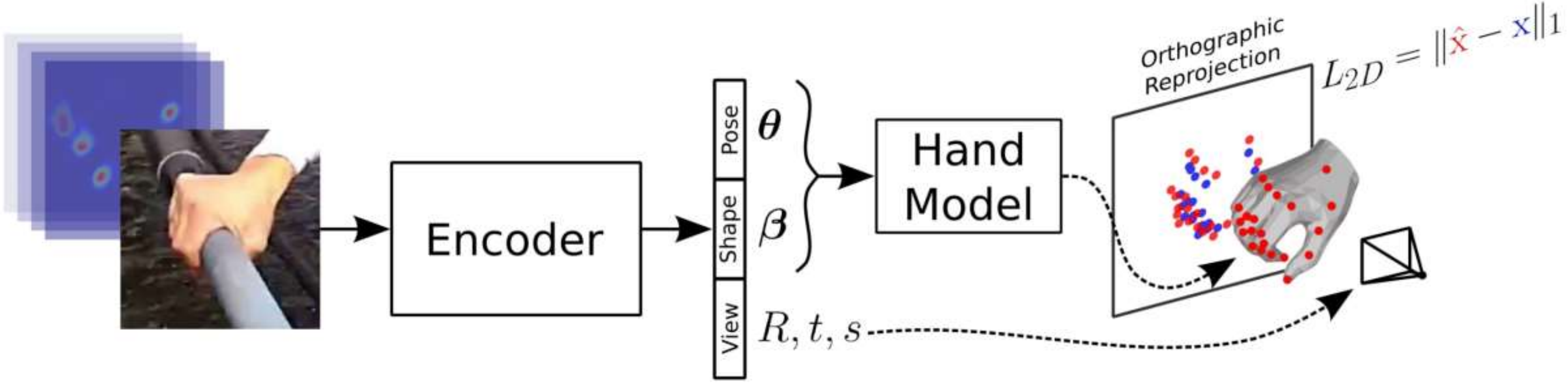


Mano Parameters

$$\boldsymbol{\beta} \in \mathbb{R}^{10} \quad \boldsymbol{\theta} \in \mathbb{R}^{16\times 3}$$

How to Estimate Hand Mesh?

3D Hand Shape and Pose from Images in the Wild , CVPR 2019



$$\hat{\mathrm{x}} = s\Pi(RJ(\boldsymbol{\beta}, \boldsymbol{\theta})) + t,$$
$$\hat{\mathrm{y}} = s\Pi(RM(\boldsymbol{\beta}, \boldsymbol{\theta})) + t$$

Supervision

$$L = L_{2D} + \alpha_{3D} L_{3D} + \alpha_{mask} L_{mask} + \alpha_{reg} L_{reg}$$

No GT
Mano Pose

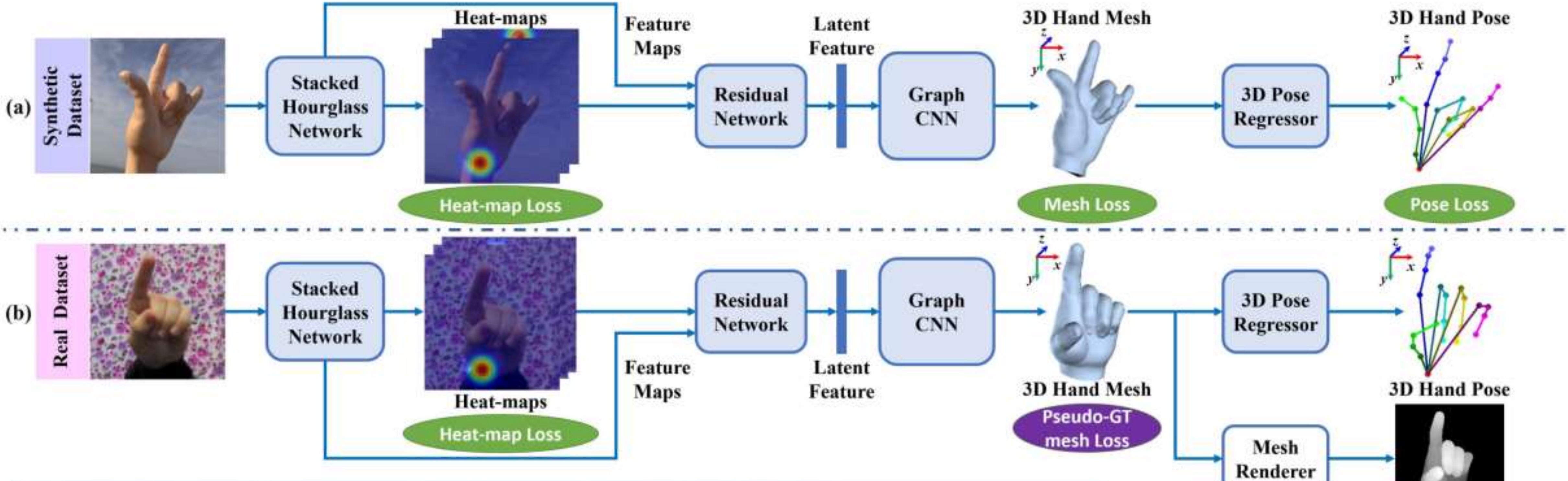
$$L_{2D} = \|\hat{\mathrm{x}} - \mathrm{x}\|_1,$$

$$L_{3D} = \|RJ(\boldsymbol{\beta}, \boldsymbol{\theta}) - \mathrm{x}_{3D}\|_2^2$$

$$L_{mask} = 1 - \frac{1}{N} \sum_i H(\hat{\mathrm{y}}_i)$$

$$L_{reg} = \|\boldsymbol{\theta}\|_2^2 + \alpha_{\beta} \|\boldsymbol{\beta}\|_2^2$$

3D Hand Shape and Pose Estimation from a Single RGB Image, CVPR 2019



(a) On synthetic dataset: (fully-supervised)	Loss = heat-map loss + mesh loss + pose loss
(b) On real dataset: (weakly-supervised)	Loss = heat-map loss + depth map loss + pseudo-ground truth mesh loss

Supervision-Synthetic Data

上海科技大学 ShanghaiTech University

$$\mathcal{L}_{fully} = \lambda_{\mathcal{H}}\mathcal{L}_{\mathcal{H}} + \lambda_{\mathcal{M}}\mathcal{L}_{\mathcal{M}} + \lambda_{\mathcal{J}}\mathcal{L}_{\mathcal{J}}$$

Heat map loss: $\mathcal{L}_{\mathcal{H}} = \sum_{j=1}^{J} \left\| \mathcal{H}_j - \hat{\mathcal{H}}_j \right\|_2^2$

Mesh loss: $\mathcal{L}_{\mathcal{M}} = \lambda_v \mathcal{L}_v + \lambda_n \mathcal{L}_n + \lambda_e \mathcal{L}_e + \lambda_l \mathcal{L}_l$

$$\mathcal{L}_v = \sum_{i=1}^{N} \left\| \boldsymbol{v}_i^{3D} - \hat{\boldsymbol{v}}_i^{3D} \right\|_2^2 + \left\| \boldsymbol{v}_i^{2D} - \hat{\boldsymbol{v}}_i^{2D} \right\|_2^2$$

Vertices loss

$$\mathcal{L}_n = \sum_t \sum_{(i,j)\in t} \left\| \langle \hat{\boldsymbol{v}}_i^{3D} - \hat{\boldsymbol{v}}_j^{3D}, \boldsymbol{n}_t \rangle \right\|_2^2$$

Normal loss

$$\mathcal{L}_e = \sum_{i=1}^{E} \left(\|\boldsymbol{e}_i\|_2^2 - \|\hat{\boldsymbol{e}}_i\|_2^2 \right)^2$$

Edge loss

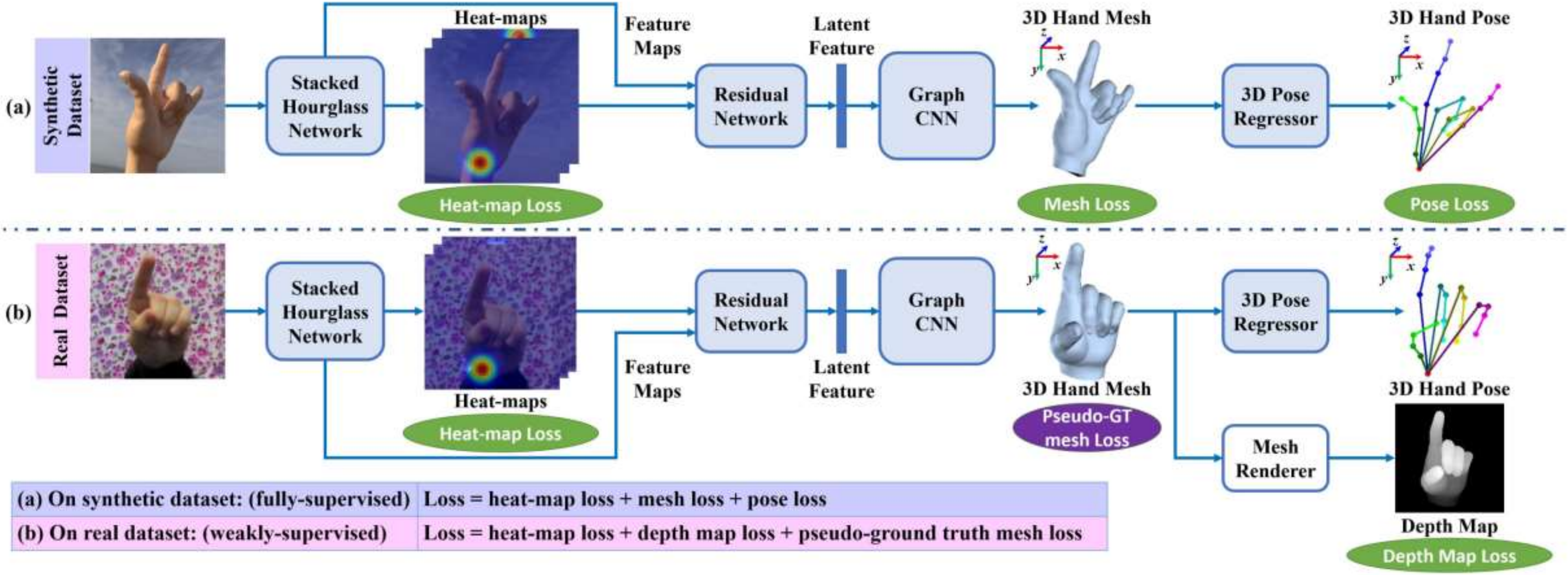
$$\mathcal{L}_l = \sum_{i=1}^{N} \left\| \boldsymbol{\delta}_i - \sum_{\boldsymbol{v}_k \in \mathcal{N}(\boldsymbol{v}_i)} \boldsymbol{\delta}_k \Big/ B_i \right\|_2^2$$

prevents the neighboring vertices from having opposite offsets

3D Pose loss: $\mathcal{L}_{\mathcal{J}} = \sum_{j=1}^{J} \left\| \phi_j^{3D} - \hat{\phi}_j^{3D} \right\|_2^2$

Joint Locations

3D Hand Shape and Pose Estimation from a Single RGB Image, CVPR 2019



(a) On synthetic dataset: (fully-supervised)	Loss = heat-map loss + mesh loss + pose loss
(b) On real dataset: (weakly-supervised)	Loss = heat-map loss + depth map loss + pseudo-ground truth mesh loss

Supervision - Real Data

$$\mathcal{L}_{fully} = \lambda_{\mathcal{H}}\mathcal{L}_{\mathcal{H}} + \lambda_{\mathcal{M}}\mathcal{L}_{\mathcal{M}} + \lambda_{\mathcal{J}}\mathcal{L}_{\mathcal{J}}$$

Heat map loss: $\mathcal{L}_{\mathcal{H}} = \sum_{j=1}^{J} \left\| \mathcal{H}_j - \hat{\mathcal{H}}_j \right\|_2^2$

Pseudo-GT Mesh loss: $\mathcal{L}_{\mathcal{M}} = \cancel{\lambda_v\mathcal{L}_v + \lambda_n\mathcal{L}_n} + \lambda_e\mathcal{L}_e + \lambda_l\mathcal{L}_l$

$$\cancel{\mathcal{L}_v = \sum_{i=1}^{N} \left\| \boldsymbol{v}_i^{3D} - \hat{\boldsymbol{v}}_i^{3D} \right\|_2^2 + \left\| \boldsymbol{v}_i^{2D} - \hat{\boldsymbol{v}}_i^{2D} \right\|_2^2}$$

Vertices loss

$$\cancel{\mathcal{L}_n = \sum_t \sum_{(i,j)\in t} \left\| \langle \hat{\boldsymbol{v}}_i^{3D} - \hat{\boldsymbol{v}}_j^{3D}, \boldsymbol{n}_t \rangle \right\|_2^2}$$

Normal loss

$$\mathcal{L}_e = \sum_{i=1}^{E} \left(\|\mathbf{e}_i\|_2^2 - \|\hat{\mathbf{e}}_i\|_2^2 \right)^2$$

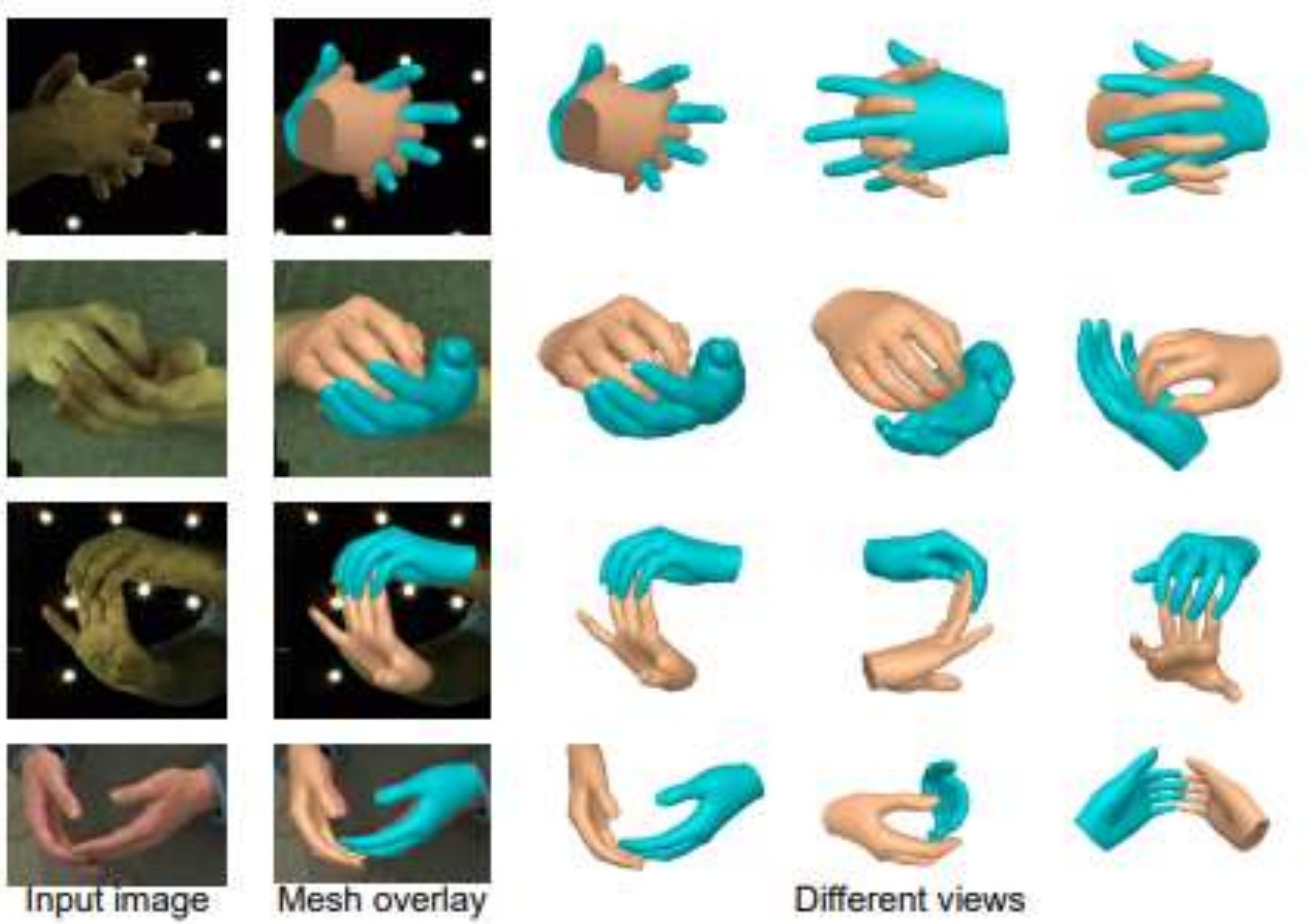
Edge loss

$$\mathcal{L}_l = \sum_{i=1}^{N} \left\| \boldsymbol{\delta}_i - \sum_{\boldsymbol{v}_k \in \mathcal{N}(\boldsymbol{v}_i)} \boldsymbol{\delta}_k \Big/ B_i \right\|_2^2$$

prevents the neighboring vertices from having opposite offsets

Depth Map loss: $\mathcal{L}_{\mathcal{D}} = smooth_{L1}\left(\mathcal{D}, \hat{\mathcal{D}}\right), \ \hat{\mathcal{D}} = \mathcal{R}\left(\hat{\mathcal{M}}\right)$

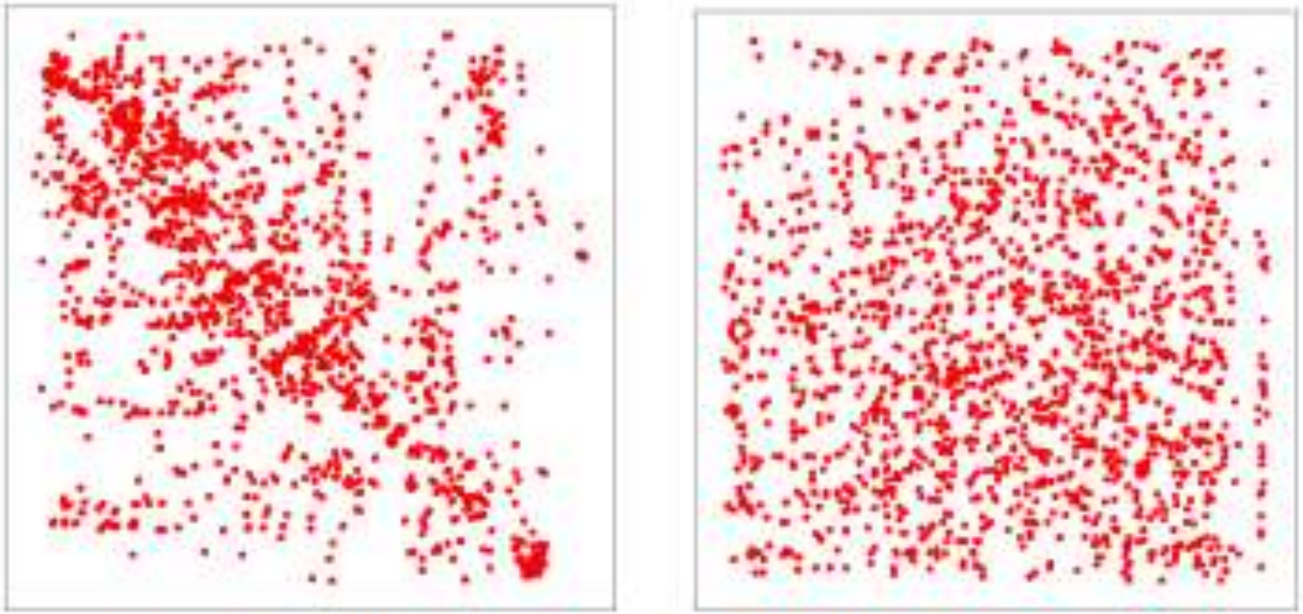
Interacting Two-Hand 3D Pose and Shape Reconstruction from Single Color Image



ICCV 2021

Two hand relationship

上海科技大学
ShanghaiTech University

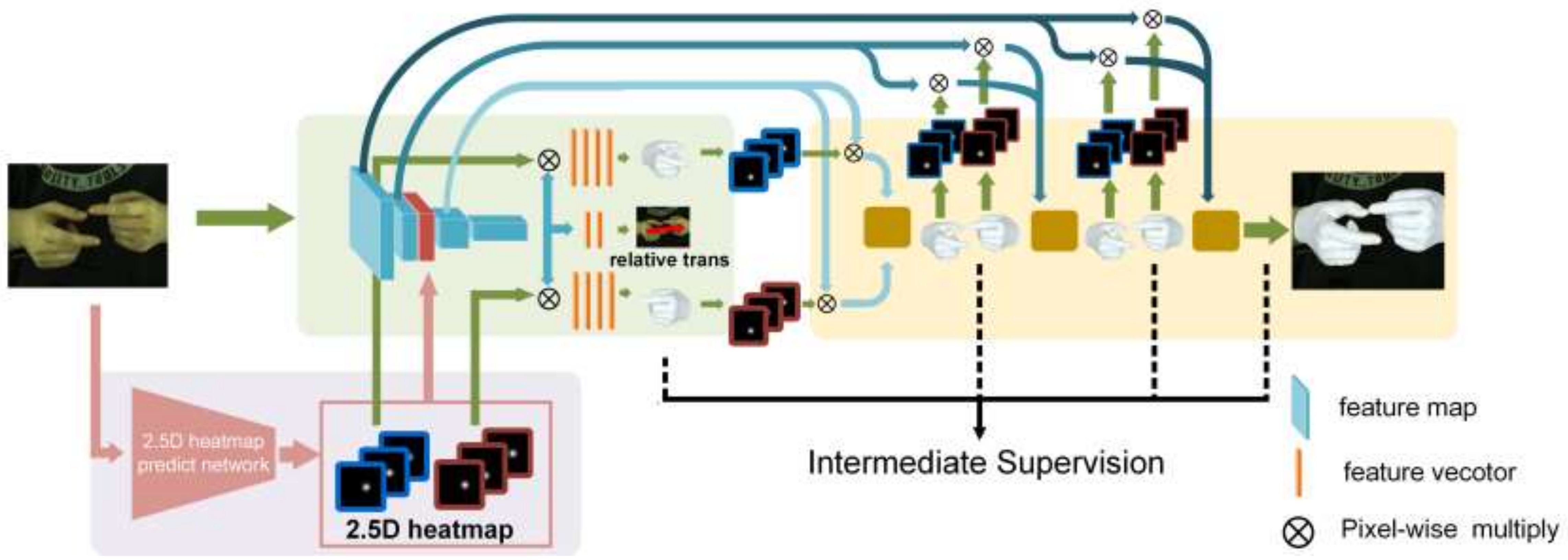


(a) Paired hand poses (b) Unpaired hand poses

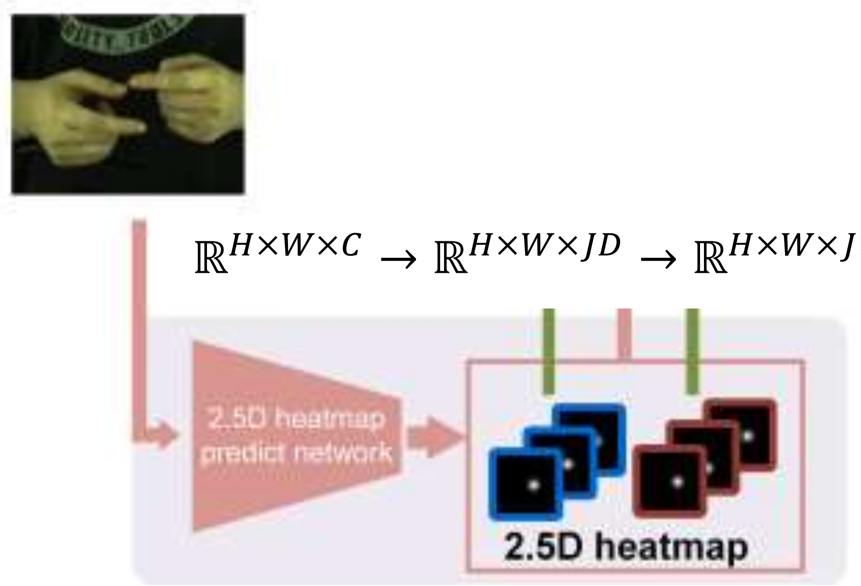
Inspired by [26], we use 2D manifold representation, where the hand pose (without root rotation) of each hand is projected to 1D manifold by t-SNE [34] and used as x, y coordinates, respectively. We find that the paired hand poses show clear correlation in 2D space, but the distribution of unpaired hand poses is almost random.

Overall Framework

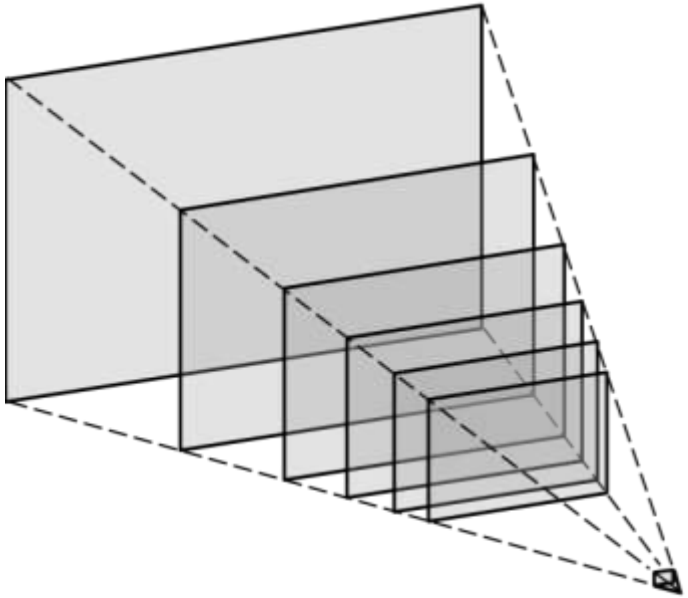
上海科技大学 ShanghaiTech University



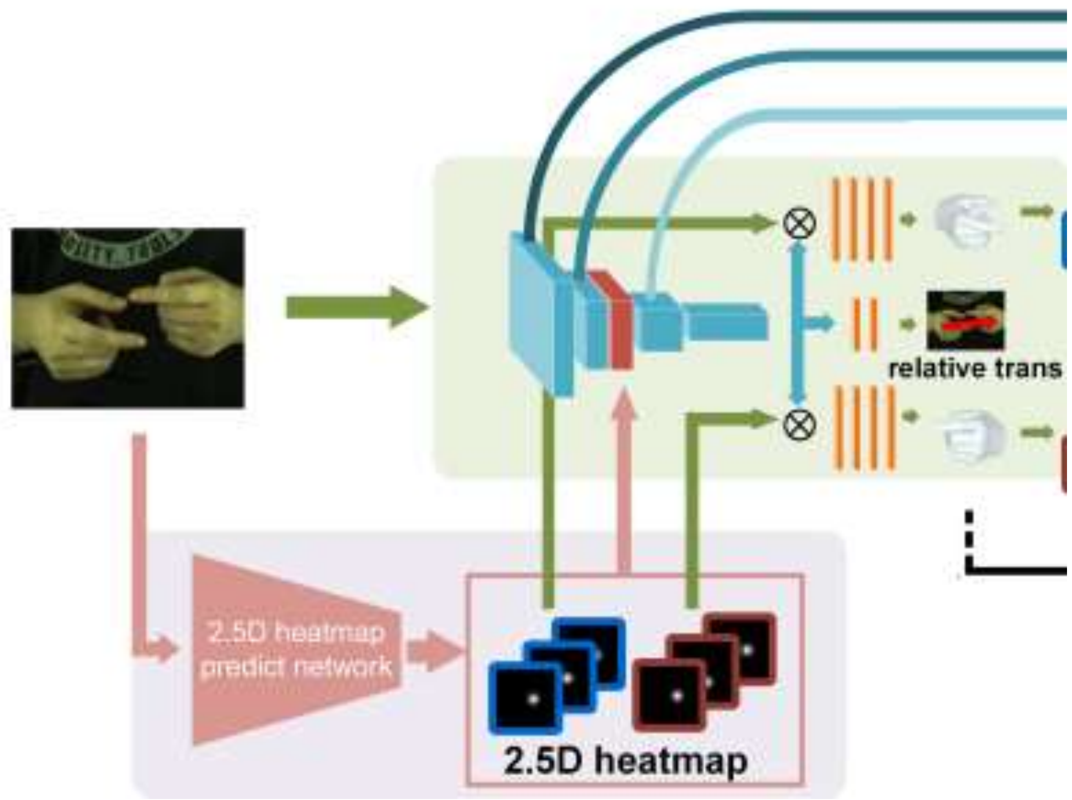
2.5D Joint keypoints heatmap estimation

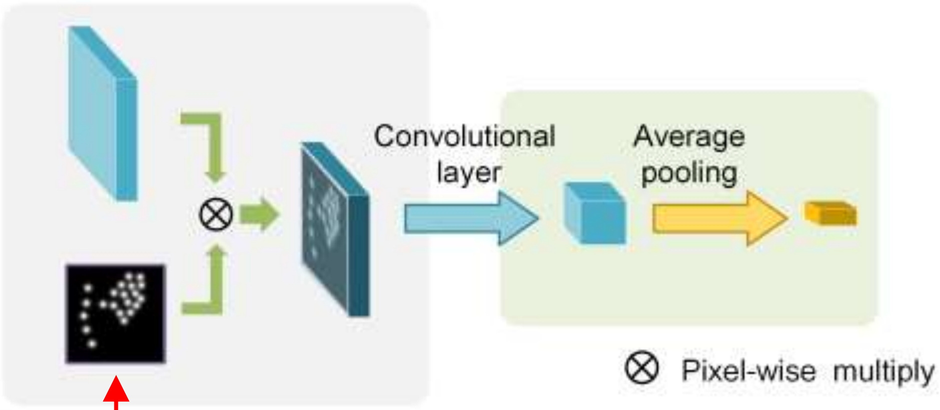


$\mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times JD} \rightarrow \mathbb{R}^{H\times W\times J\times D}$



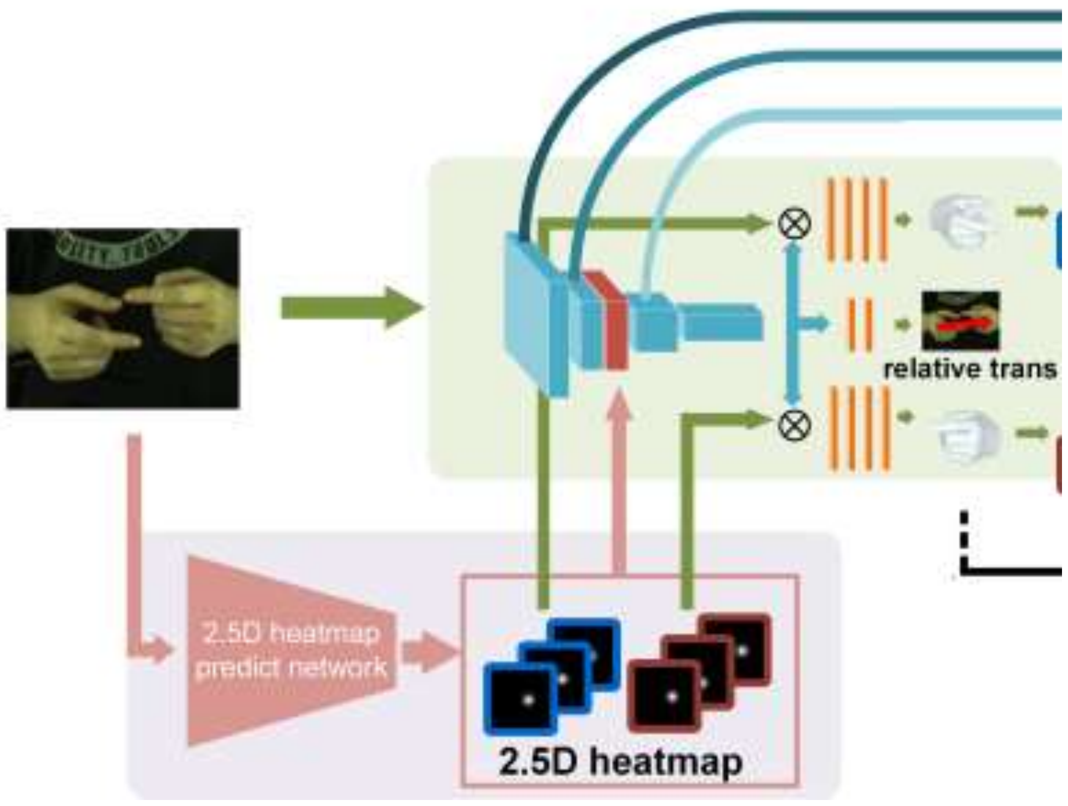
Feature Extraction with Keypoints Attention





$$\mathbf{A}_h = 1 - \prod_{i=1}^{K}(1-\mathcal{H}_{h_i}), \quad h \in \{right, left\},$$

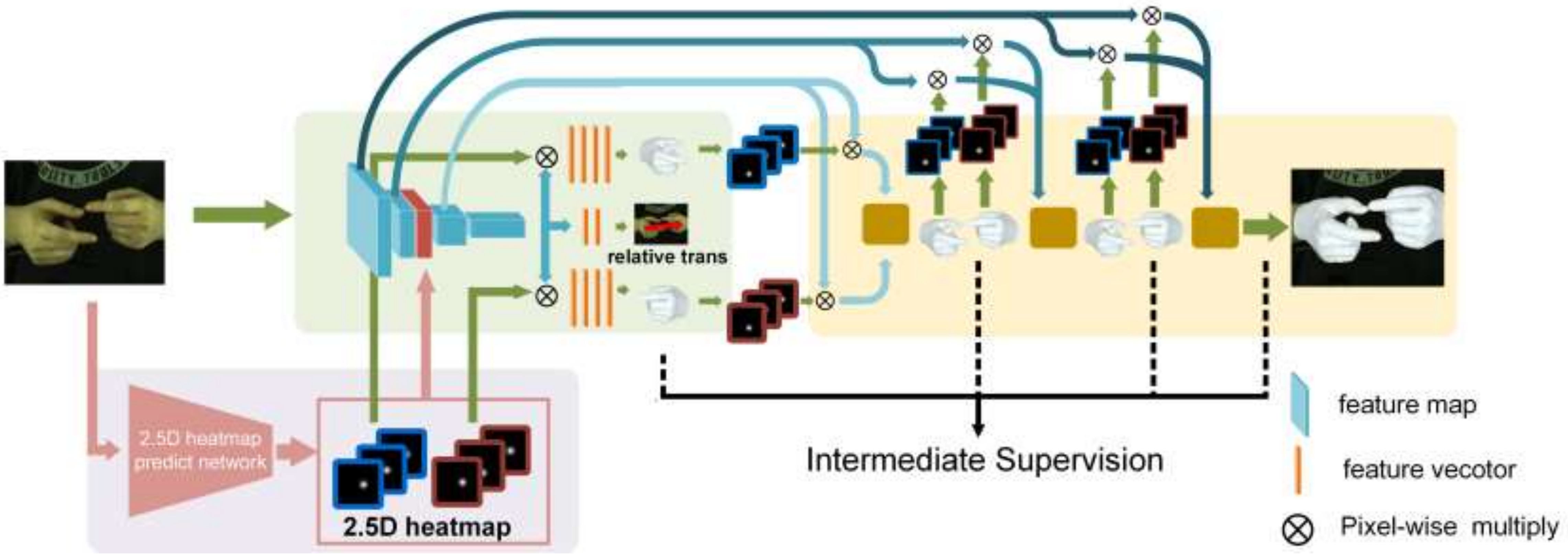
Relative Transformation



$$\mathbf{J}_{left,i}^{right} = s(\mathbf{J}_{left,i}^{left} + \Delta)$$

where $\mathbf{J}_{left,i}^{right}$ and $\mathbf{J}_{left,i}^{left}$ are the left hand joints in the right hand coordinate system and the left hand joints in the left hand coordinate system, respectively.

Overall Framework



Supervision

- Two hand loss
  - Joint offset loss $L_o = \sum_{i=1}^{K} ||(\mathbf{J}_{right,i} - \mathbf{J}_{left,i}) - (\mathbf{J}^*_{right,i} - \mathbf{J}^*_{left,i})||_2^2$
  - Shape consistence loss $L_c = ||\beta_{right} - \beta_{left}||_2^2$
- Single hand loss
  - Joint loss $L_J = \sum_{h\in\{left,right\}} \sum_{i=1}^{K} ||\mathbf{J}_{h,i} - \mathbf{J}^*_{h,i}||_1$
  - Bone length loss $L_l = \sum_{h\in\{left,right\}} \sum_{b} ||\frac{l^*_{h,b}}{l^*_{h,ref}} - l_{h,b}||_2^2$
  - Shape loss $L_M = \sum_{h\in\{left,right\}} \mathbf{1}||\beta_{\mathbf{h}} - \beta^*_{\mathbf{h}}||_{\mathbf{2}}^{\mathbf{2}}$
  - Regularizer loss $L_{reg} = \sum_{h\in\{left,right\}} \lambda_\beta ||\beta_h||_2^2 + ||\theta_h||_2^2$

Supervision

$$\begin{aligned} L_{total} = &\lambda_o L_o + \lambda_c L_c + \lambda_J L_J \\ &+ \lambda_l L_l + \lambda_M L_M + \lambda_{reg} L_{reg} \end{aligned} \tag{10}$$

where $\lambda_o, \lambda_c, \lambda_J, \lambda_l, \lambda_M, \lambda_{reg}$ are the loss weights, and they are set to 1, 0.01, 10, 100, 0.1, and 0.05, respectively.

References

- Boukhayma, Adnane, Rodrigo de Bem, and Philip HS Torr. "3d hand shape and pose from images in the wild." *CVPR* 2019.
- Zhang, Baowen, et al. "Interacting two-hand 3d pose and shape reconstruction from single color image." ICCV 2021.
- https://github.com/iscas3dv/Two-Hand-Shape-Pose_v2
- Li, Mengcheng, et al. "Interacting attention graph for single image two-hand reconstruction." CVPR 2022
- https://mediapipe.readthedocs.io/en/latest/solutions/hands.html
- https://medium.com/@turgay2317/hand-detection-and-finger-counting-in-python-40f21719f1b6

Project Requirement (Basics)

上海科技大学 ShanghaiTech University

- Take pictures (20 images) of your own or your friends’ hands, with diverse perspective/viewpoint/background
  - Evaluate the trained model of [Zhang et al. ICCV 2021] on your newly collected hand images;
  - Visualize the estimated 2D & 3D keypoints, and mesh models (overlayed with the original image);
  - Analyze the performance, especially on failure scenarios.

```python
def register_heatmap(self,xyc:torch.Tensor,J:torch.Tensor,origin_size:int,output_size:int):
    '''
    xyc: shape of [B, N, 3];  N -- Joint Number; 3 -- [u,v,confidence]
    J: shape of [B, N, 3]; 3D joints coordinates
    '''
    device_run=xyc.device
    batch_size=xyc.shape[0]
    M=torch.cat([J[:,:,:2,None],torch.eye(2,device=device_run)[None,None,:,:].repeat(batch_size,J.shape[1],1,1)],dim=-1)
    wM=xyc[:,:,2,None,None]*M
    wB=xyc[:,:,2,None,None]*xyc[:,:,:2,None]
    wM=wM.reshape(batch_size,-1,3)
    wB=wB.reshape(batch_size,-1,1)
    MTM=torch.bmm(wM.transpose(2,1),wM)
    MTB=torch.bmm(wM.transpose(2,1),wB)
    sT=torch.bmm(torch.inverse(MTM),MTB)[:,:,0].detach()
    ratio=output_size/origin_size
    projected_xy=(J[:,:,:2]*sT[:,None,0,None]+sT[:,None,1:])*ratio
    #sigma = cfg.sigma * 2 * max(ratio, 0.25)
    heatmap=self.generate_batch_heatmap(projected_xy,output_size, sigma: 3)
    output_heatmap=1-torch.prod(1-heatmap,dim=1)
    return output_heatmap
```

Least Square Method

$$Ax = b$$

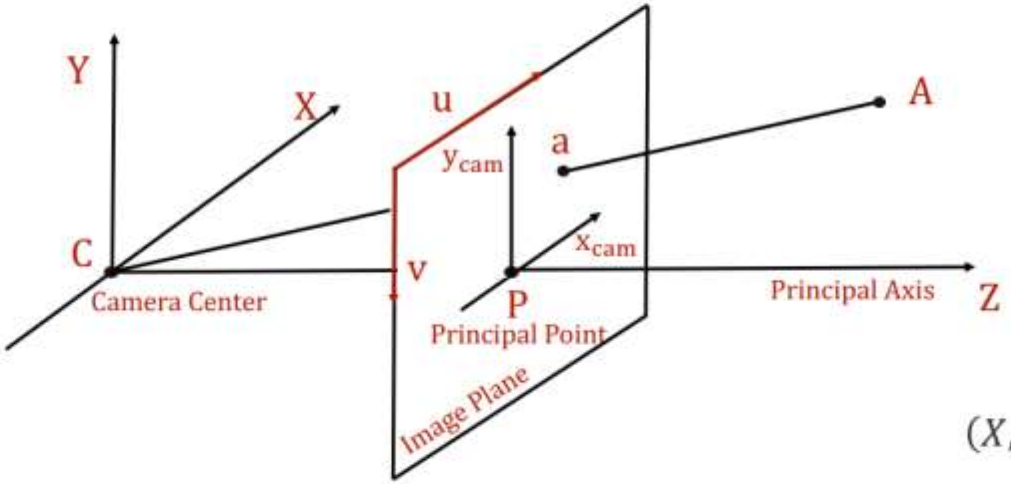
$$x = (A^T A)^{-1} A^T b$$

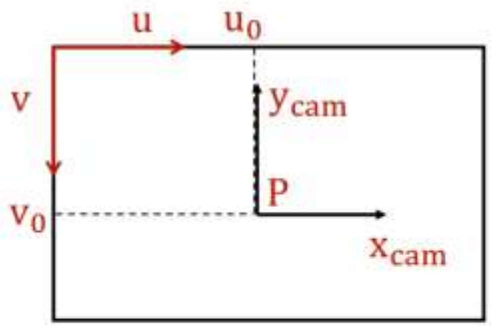
Perspective Projection

Perspective projection

Assumptions made:

- Principal point at ~~origin~~ (u_0, v_0) of image plane
- Camera at center of world coordinates
- Square pixels





$$(X, Y, Z)^T \mapsto (u, v)^T = \left(\frac{fX}{Z} + u_0, \frac{fY}{Z} + v_0\right)^T$$

Euclidean Coordinates
3D World frame

Euclidean coordinates
2D Image plane
(Note nonlinearity)

Call for Presentations!

上海科技大学
ShanghaiTech University

- Project-5
  - Multi-view Stereo for view synthesis
  - NeRF
  - 3DGS
  - Other view synthesis variants.

- Project-6
  - Latent Diffusion
  - LoRA
  - ControlNet
  - Other diffusion variants.