Hand Pose Estimation from Local Surface Normals

Chengde Wan[1], Angela Yao[2], Luc Van Gool[1, 3]
[1]Computer Vision Laboratory, D-ITET, ETH Zurich; [2]Institue of Computer Science, University of Bonn; [3]VISICS, ESAT, K.U. Leuven

1 Highlights

• Hand pose estimation from single depth image with surface normal

• local reference frames with local surface normals are invariant to rigid body transformation, so no data augmentation is needed

• local surface normal difference is invariant to rigid body transformation and better captures the local geometrical property of hand surface, e.g., curvature

2 Random normal difference

calculate random position $r(\cdot,\cdot)$ → loop up feature map $\phi_{\mathcal{I}}(\cdot)$ → calculate feature difference $\Delta(\cdot,\cdot)$

$$f_{\mathcal{I}}(\mathbf{p}_i, \delta_1, \delta_2) = \Delta(\phi_{\mathcal{I}}(r(\mathbf{p}_i, \delta_1)), \phi_{\mathcal{I}}(r(\mathbf{p}_i, \delta_2)))$$

3 Random normal difference

invariant to rigid body transformation iff

• random offset invariant to rigid body transformation, i.e., the relative position between $\mathbf{p}_i$ and $r(\mathbf{p}_i, \delta_j)$ remains unchanged after transformation:

$$T(\mathbf{p}_i - r(\mathbf{p}_i, \delta_j)) = T(\mathbf{p}_i) - r(T(\mathbf{p}_i), \delta_j)$$

• feature channel difference invariant to rigid body transformation:

$$\Delta(\phi_{\mathcal{I}}(\mathbf{q}_1), \phi_{\mathcal{I}}(\mathbf{q}_2)) = \Delta(\phi_{\mathcal{I}'}(\mathbf{q'}_1), \phi_{\mathcal{I}}(\mathbf{q'}_2))$$

Random normal difference feature

• $r(\cdot,\cdot)$ is calculated **with the help of local surface normal**, invariant to in plane rotation:

- for edge point, determined by normal
- for inner point, determined by normal + reference point

• $\Delta(\cdot,\cdot)$ is **the dot product of two surface normal directions**, invariant to rigid body transformation, while depth difference is not invariant to rigid body transformation.

Approximated surface normal calculation

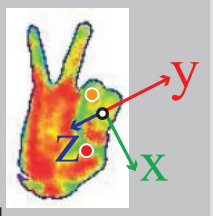
• Different treatment of edge vs. inner points from point cloud

• Edge points: drops down to 2D curve normal estimation

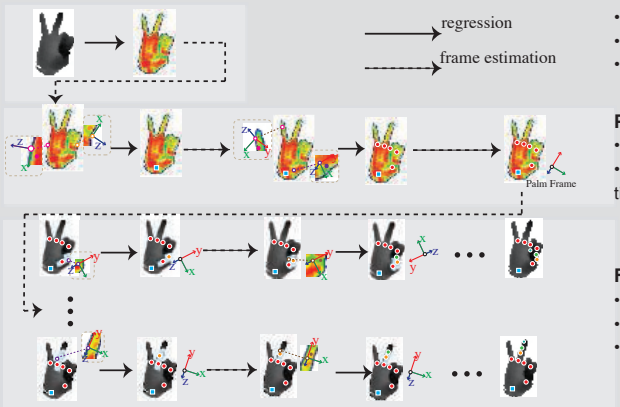
• Inner points: find direction with smallest eigenvalue

• Eigen-decomposition on each point is computationally expensive, replace with a random forset regressor (14 vs. 4 ms)

• Reparameterize normal vector as polar and azimuth angle, modeled as Von Mises Distribution to maximize entropy during training. More details can be found in the paper



4 Hierarchical estimation



Frame Conditioned Regression Forest

• Regress the offset between joint j and point i conditioned on referend frame

• Training: offset is normalized w.r.t. the referenced frame;

• Testing: firstly regress w.r.t. referenced frame, then transformed into the global frame

Palm Pose Estimation

• wrist location estimated with only edge points

• surface normal assumed orthogonal to the image plane based on orthographic projection

$$\mathbf{x}^{(i)}_{wrist} = \mathbf{n},$$
$$\mathbf{y}^{(i)}_{wrist} = \mathbf{z}^{(i)}_{wrist} \times \mathbf{x}^{(i)}_{wrist},$$
$$\mathbf{z}^{(i)}_{wrist} = \mathbf{n}_i,$$

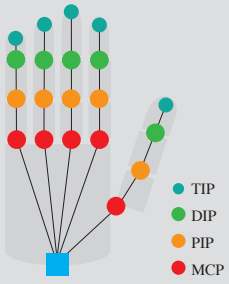
Finger Pose Estimation

• use the palm frame to estimate the PIP points

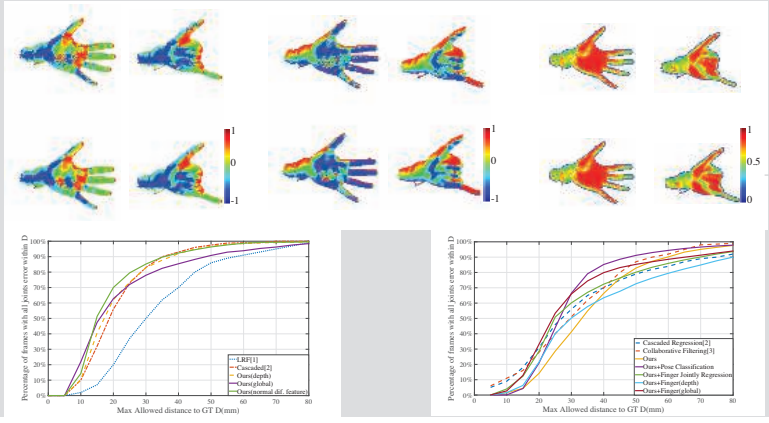
• use the finger frame to estimate the DIP/TIP points

• More details can be found in the paper

$$\mathbf{x}^{(i)}_{MCP} = \mathbf{y}^{(i)}_{MCP} \times \mathbf{z}^{(i)}_{MCP},$$
$$\mathbf{y}^{(i)}_{MCP} = \frac{\mathbf{n}_i \times (\mathbf{p}_{wrist} - \mathbf{p}_i)}{\|\mathbf{n}_i \times (\mathbf{p}_{wrist} - \mathbf{p}_i)\|_2},$$
$$\mathbf{z}^{(i)}_{MCP} = \mathbf{n}_i,$$



5 Results



References:
[1] Tang, D., Chang, H.J., Tejani, A., Kim, T.K.: Latent regression forest: Structured estimation of 3D articulated hand posture. In: CVPR. (2014)
[2] Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: CVPR. (2015)
[3] Choi, C., Sinha, A., Choi, J.H., Jang, S., Ramani, K.: A collaborative filtering approach to real-time hand pose estimation. In: ICCV. (2015)