Hand Segmentation and Pose Estimation Research Review

Evan Cornish, 03/31/2023

Hand Pose Summary

Hand Pose Summary and Progress

- Hand pose Summary
- Different CNN architectures
- Accuracies
- Segmentation
- My Progress
- Next Steps

Hand Pose Summary

- Early methods
  - Fit 3D articulated hand model to input visual data
    - Some non neural network based methods (HOPE network using canny, as well as Tompson paper with LBS (linear blend skinning)
- CNN models dominate

- Hand only problem has received most attention
  - HOI (Hand object interaction), and cluttered image are much less explored

Data Driven Approaches

- Search based methods
  - Perform badly in high dimensional space
- Random Forests
  - Heavily reliant on hand crafted features
- DNN based methods
  - Several methods (More coming)

Challenge Analysis

- Articulation
  - Each hand joint is a kinematic chain with 1 or 2 degrees of freedom (DOF)
- Orientation
  - Global orientation has an additional 6 DOF
  - Between articulation and orieintation we have 26 DOF
- Occlusion
  - Estimating 3D HP from 2D projection is ill-posed problem
  - Well posed problem:
    - Existence: at least 1 solution
    - Uniqueness: only 1 clear answer
    - Stability: small change to input leads to small change to output
  - Estimating 3D HP from 2D projection fails uniqueness, there is more than 1 correct answer
- Self-Similarity
- Depth and Scale Ambiguities
  - Causes over-fitting for environment
  - Methods tend to assume scale (distance from camera, or environmental structure)
- Noise
- Clutter
  - Most methods tend to assume no clutter
  - Also no object interaction
- Data Collection Expensive

State Of The Art DNN Models (As Per 2021)

- Multiple Paradigms
  - Generative Model Based (Not really DNN necessarily but can be)
  - Regression Based
  - Detection Based

Generative Model Based

- Hand model with prior anatomy built in as assumption
- Use non convex energy function defined to measure discrepancy between hand and model
- Energy function
  - Assigns low energy to correct values of remaining points
  - High energy to incorrect values
  - Loss function measures quality of energy produced by energy function
- Examples
  - PSO, ICP (iterative closest point), “Nonlinear optimization” (whatever that means)
  - Common paradigm used for offline modeling
- Weaknesses
  - Requires good initialization, which is not realistic
  - Often assumed that the previous frame has good info about the next frames initialization
  - Over trains on hand crafted from 3D model (doesn’t generalize well to all hand shapes and sizes)

Regression Based Models (Discriminative)

- E2E learning with direct prediction of joint locations
  - Baseline: global regression in 1 stage
    - Works terribly (This was Preston and I’s first attempt already)
  - Examples: DeepPrior(2015), DeepPrior++(2017)
    - Assumes hand is segmented and that depth invariance is not important
    - One of the best working DNN solutions
  - Transform into voxels, then do 3d CNN
    - Examples: 3DCNN, pointnet, pointnet++
    - pointnet preprocess with knn and point sampling
    - Some experiments made with CNN on multiple cameras (Not sure which models)

Detection Based Approaches (discriminative)

- Take a cluster of pixels and produce a 2D or 3D gaussian per joint
- Requires deconvolution process to produce heatmap
  - Time consuming and expensive
  - Hurts real time capabilities
- Generally more accurate than regression based methods
  - Only loosely true, not ALWAYS true
- Often requires some pose recovery algorithm like inverse kinematics
- Examples
  - Tompson, V2Vnet, PointNet(point clouds)

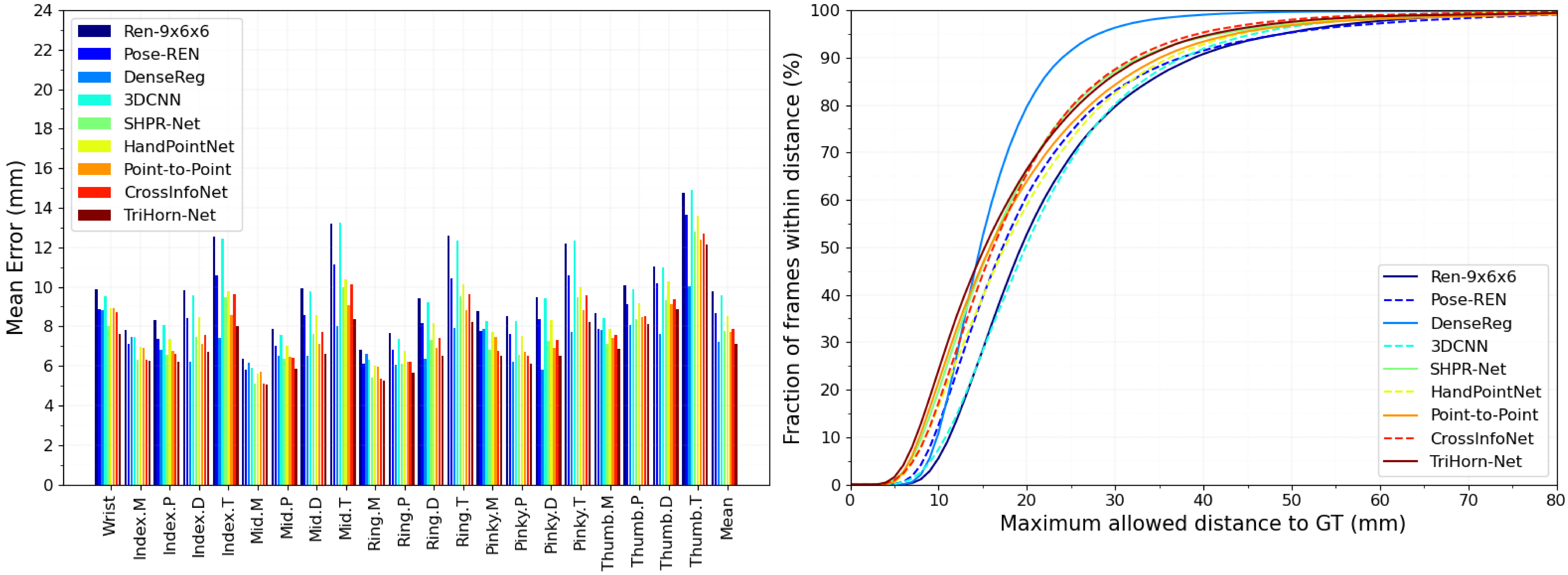
Structural Based Approaches

- Use PCA or structural model to restrict pose to kinematically plausible poses
- Don’t know as much about this, there are a bunch of sources, not sure which models are using it

Multi Stage and Ensemble Models

- DeepPrior uses a multi stage paradigm (don’t know what that means)
- Ensemble methods involve using 2D heatmap, 3D heatmap and 3D directional vector fields
  - Use each model to predict independently
- Examples
- Hand Branch Ensemble (HBE), A2J, JGR-2PO

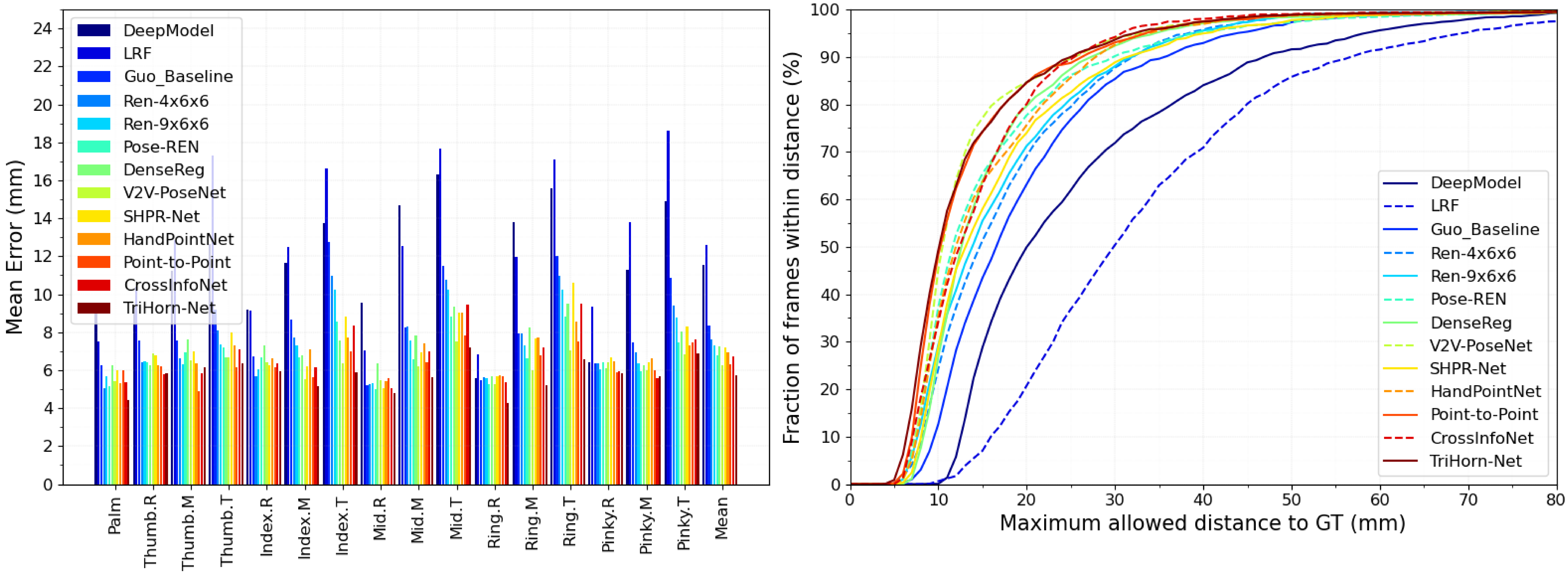
MSRA Error



Notable Models Error in avg mm

- TriHorn-Net: 7.13
- JGR-P20: 7.55 (Ensemble Method)
- DenseReg: 7.23
- P2P: 7.7 (CNN method, detection based (heatmap))
- DeepPrior++: 9.5 (CNN regression)

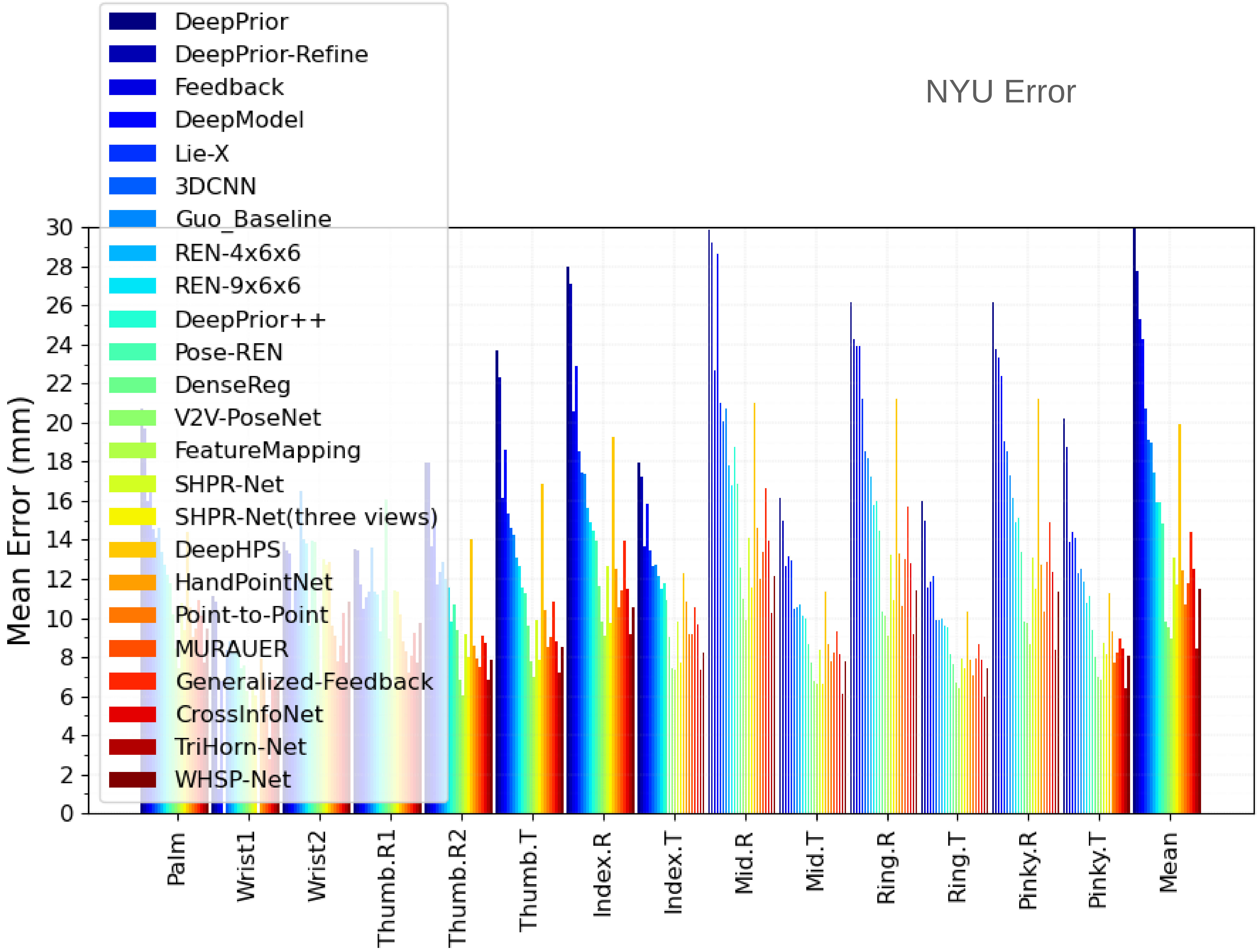
ICVL Error

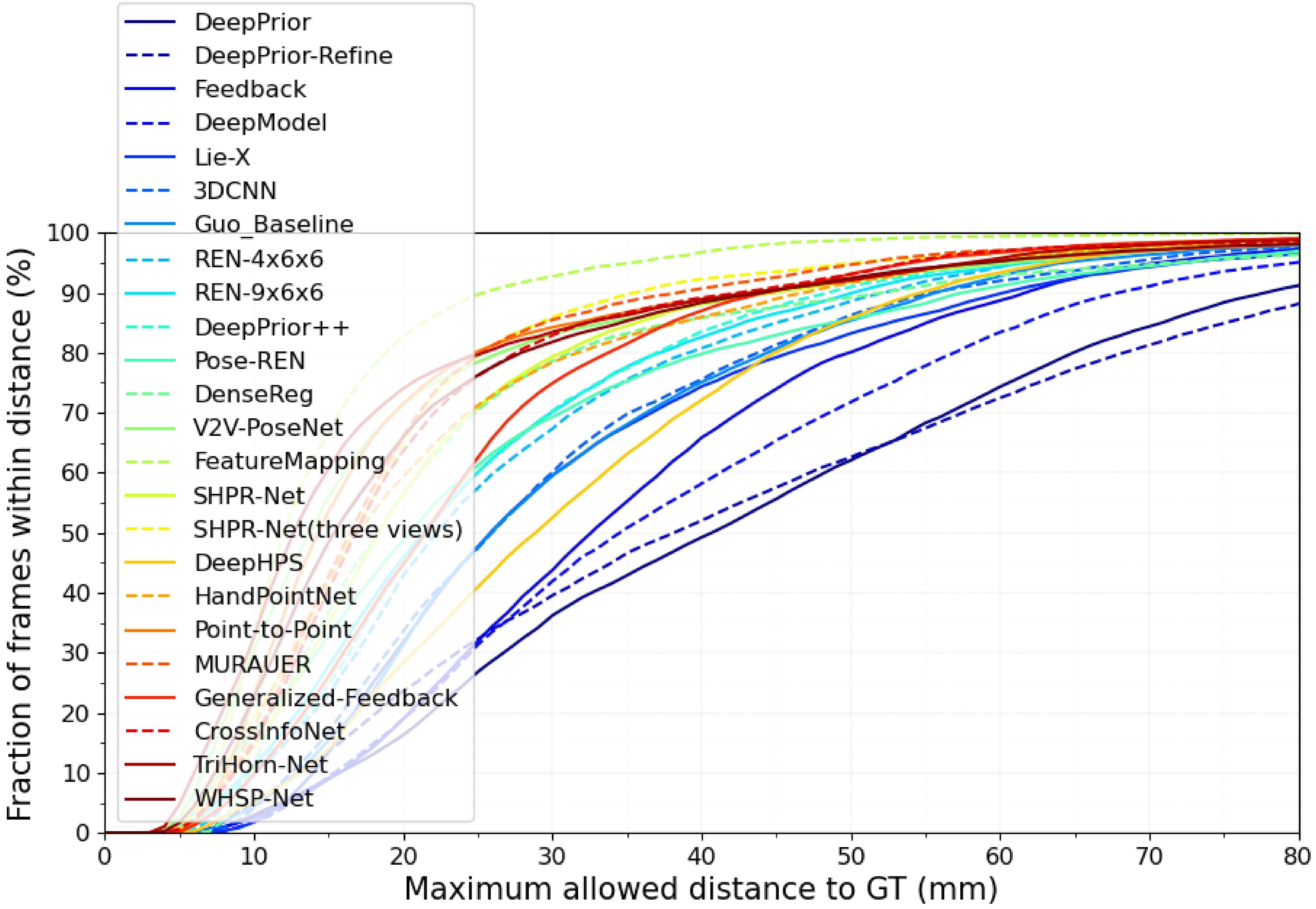


Notable Models Error in avg mm

- TriHorn-Net: 5.73
- JGR-P20: 6.02 (ensemble method)
- P2P: 6.3 (CNN detection based (heatmap))
- V2V-PoseNet: 6.2
- A2J: 6.46(Detection)
- DeepPrior++: 8.1 (CNN regression based)

NYU Error





Notable Models Error in avg mm

- TriHorn-Net: 7.68
- JGR-P20: 8.29 (ensemble method)
- P2P: 9.1 (CNN detection based (heatmap))
- V2V-PoseNet: 8.42
- A2J: 8.61(CNN detection based (heatmap))
- DeepPrior++: 12.24 (CNN regression based)

DeepPrior++

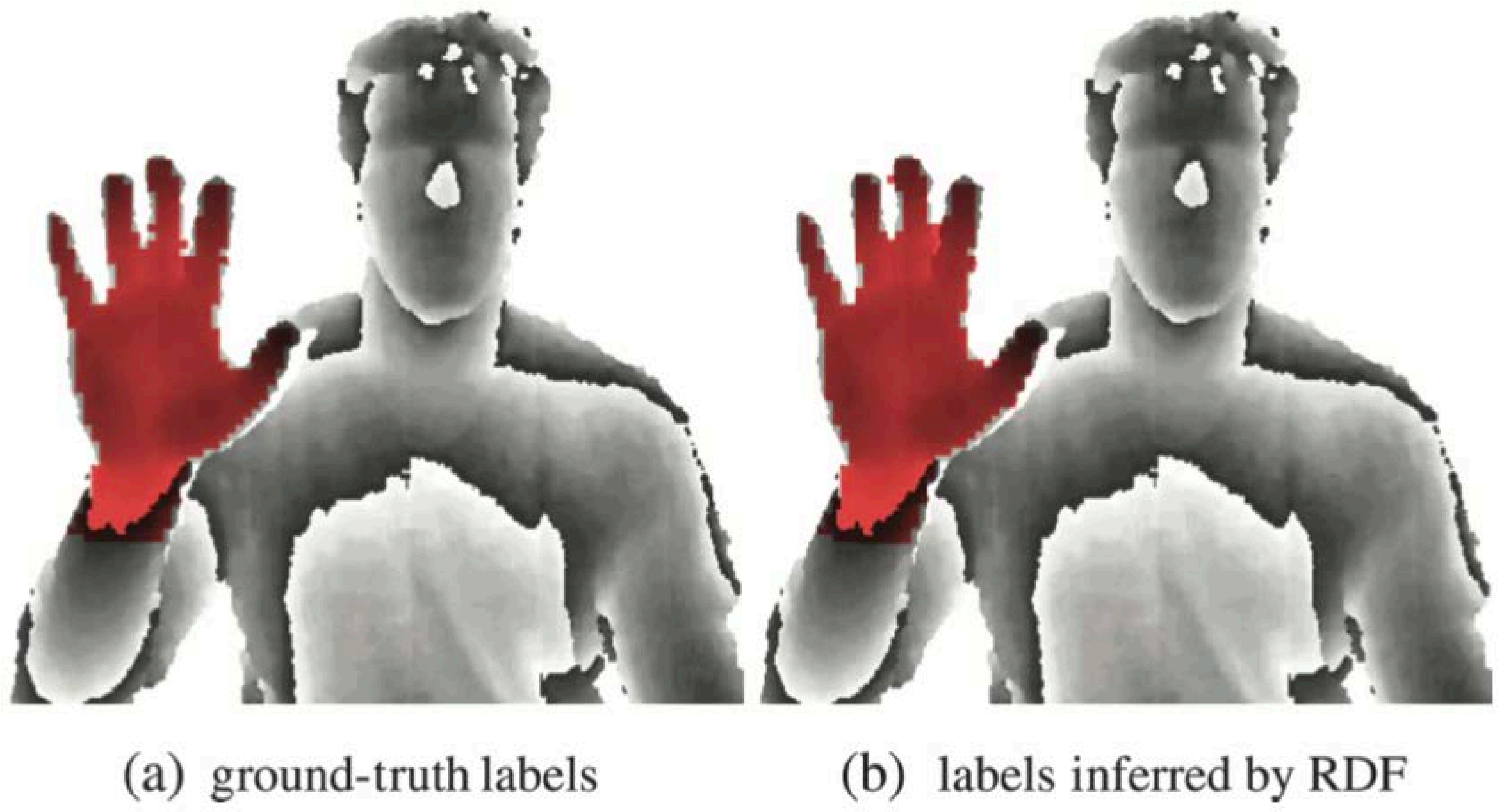
Ablation Experiment

Localization	Avg. 3D pose error	Loc. 3D error
CoM	13.8mm	28.1mm
Refined CoM	12.3mm	8.6mm
Ground truth	10.8mm	0.0mm

- Shows the average error improvement with different segmentation methods
- Using the ground truth to segment the hand improves accuracy by 2mm
- Shows the importance of segmenting the hand with as little error as possible in the process of the problem

Segmentation Problem

- Implementing the segmentation procedure in the Tompson paper
- Create manually crafted data features from a random sampling of the input image
- Use RDF to segment the hand from the rest of the image



Tompson Paper Figure 2

RDF segmentation

Tompson Feature Formula

$$I\left(u + \frac{\Delta u}{I(u,v)}, v + \frac{\Delta v}{I(u,v)}\right) - I(u,v) \geq d_t, \qquad (1)$$

- Don’t just throw Neural Nets at every problem
- Manually decide upon feature patterns
- Decide upon feature calculation metric
- Can pick feature pattern or do so randomly

Shotton Fig. 4