Random Forests for Kinect

Adrian Barbu

Overview

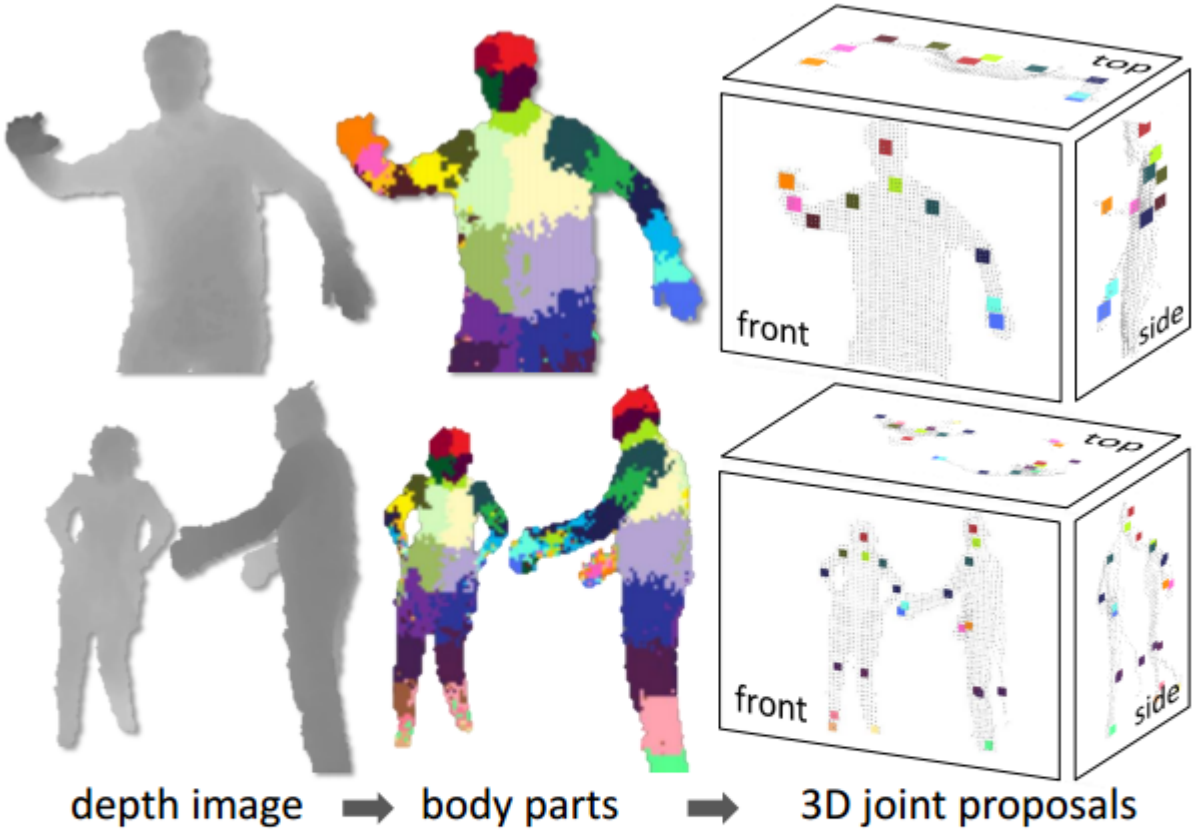


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

Kinect = Depth Camera

- Depth Camera:
  - Pixels indicate calibrated depth in that direction
  - Kinect = 640x480 with 30 fps
  - Obtained using structured infrared light and stereo matching

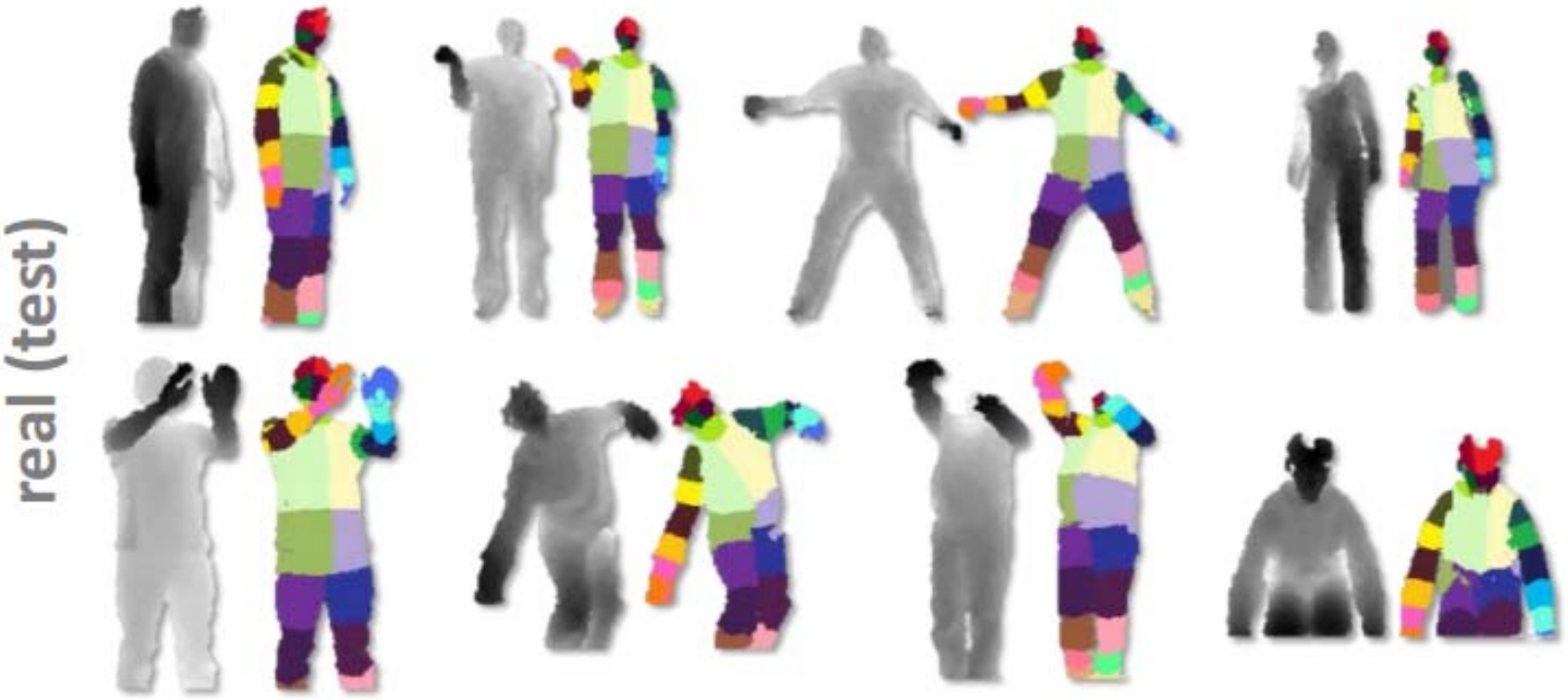
- Advantages over image camera
  - Data has less variability
  - Easier to train

Training/Test Data

- Synthetic Data
  - Obtained using motion capture
  - People wearing special clothes with marked body parts
  - Depth image generated by a 3D rendering algorithm
  - 500k images

Test Data

- Manually annotated label maps
- 31 body parts



Features

- A feature vector is extracted for each pixel of the image
- Depth difference features

$$f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right), \quad (1)$$

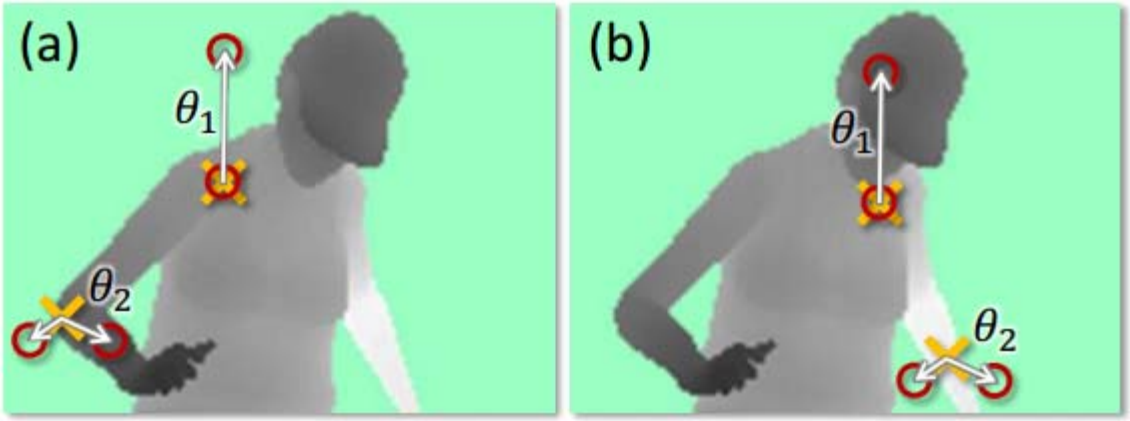


Figure 3. **Depth image features.** The yellow crosses indicates the pixel $\mathbf{x}$ being classified. The red circles indicate the offset pixels as defined in Eq. 1. In (a), the two example features give a large depth difference response. In (b), the same two features at new image locations give a much smaller response.

Random Forest

- Can be implemented on the GPU

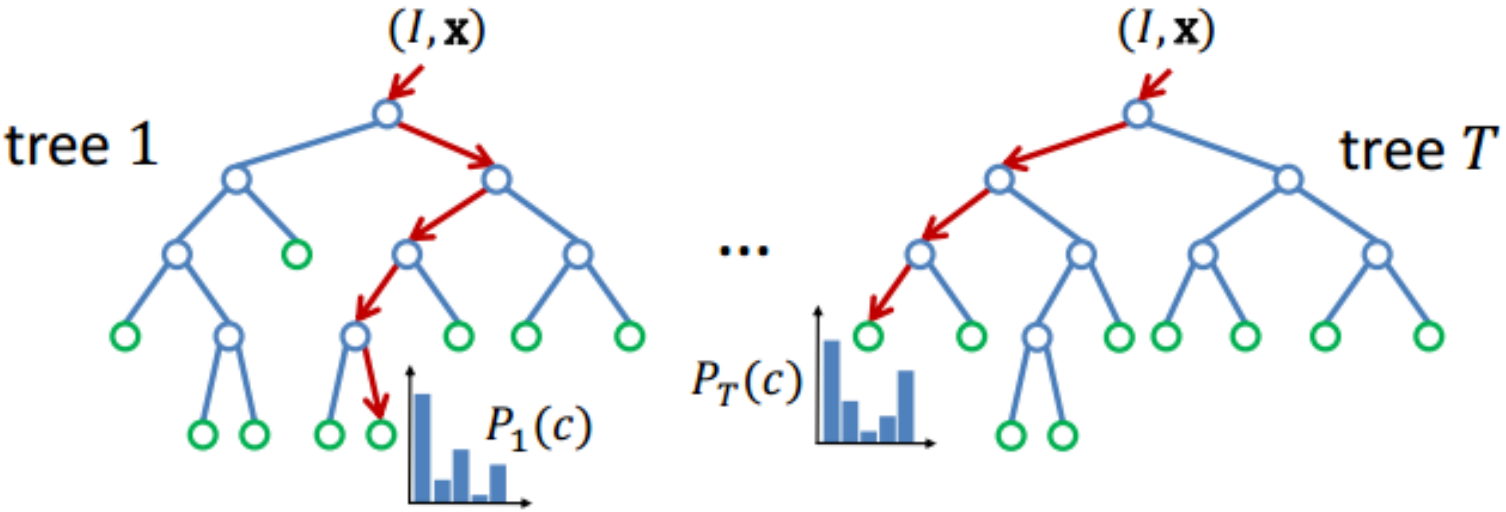


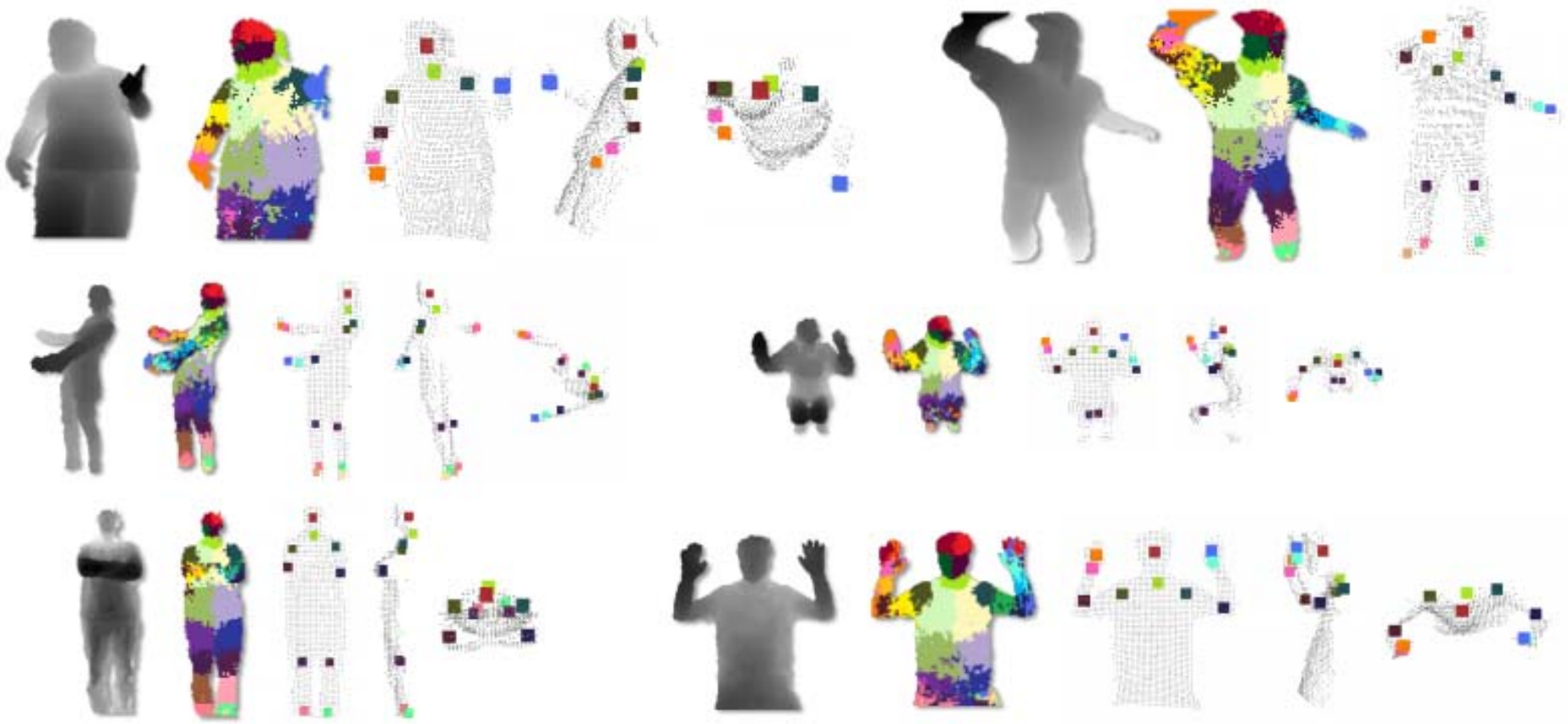
Figure 4. **Randomized Decision Forests.** A forest is an ensemble of trees. Each tree consists of split nodes (blue) and leaf nodes (green). The red arrows indicate the different paths that might be taken by different trees for a particular input.

RF Training

- Each tree is trained on a different set of images
- 2000 pixels for each image
- At each node
  - A set of features and thresholds are chosen randomly
  - The best attribute is selected if
    - The information gain is above a threshold
    - The node depth is below a maximum value
- Training details
  - 3 trees of depth 20
  - 300k images per tree
  - 2000 features, 50 thresholds
  - 1 day on a 1000 core cluster

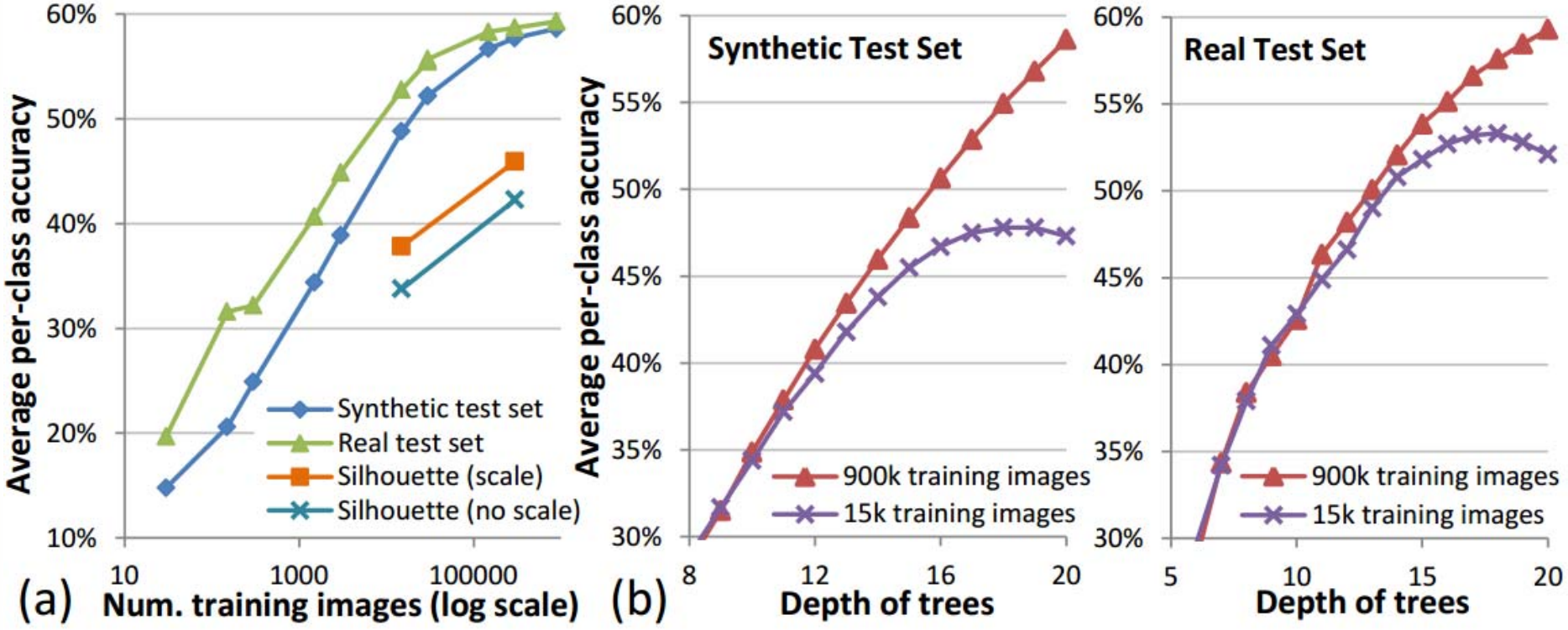
Joint Positions

- Local modes from pixel response maps by Mean Shift



Results

- 5000 synthetic frames
- 8808 real depth images of 15 subjects



Results

- Avg precision = average of the precision for 10 recall values: 0.1,0.2,…,1.0

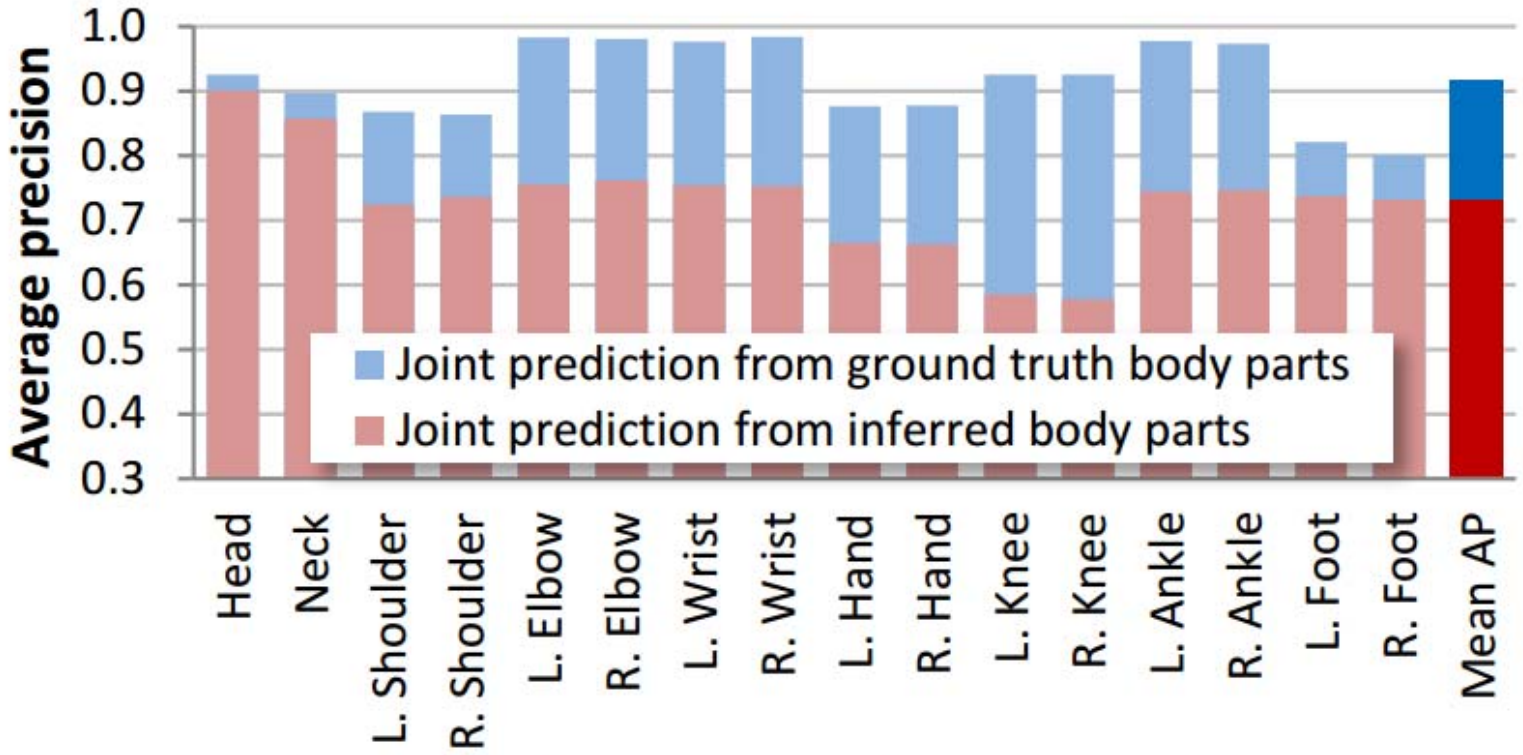
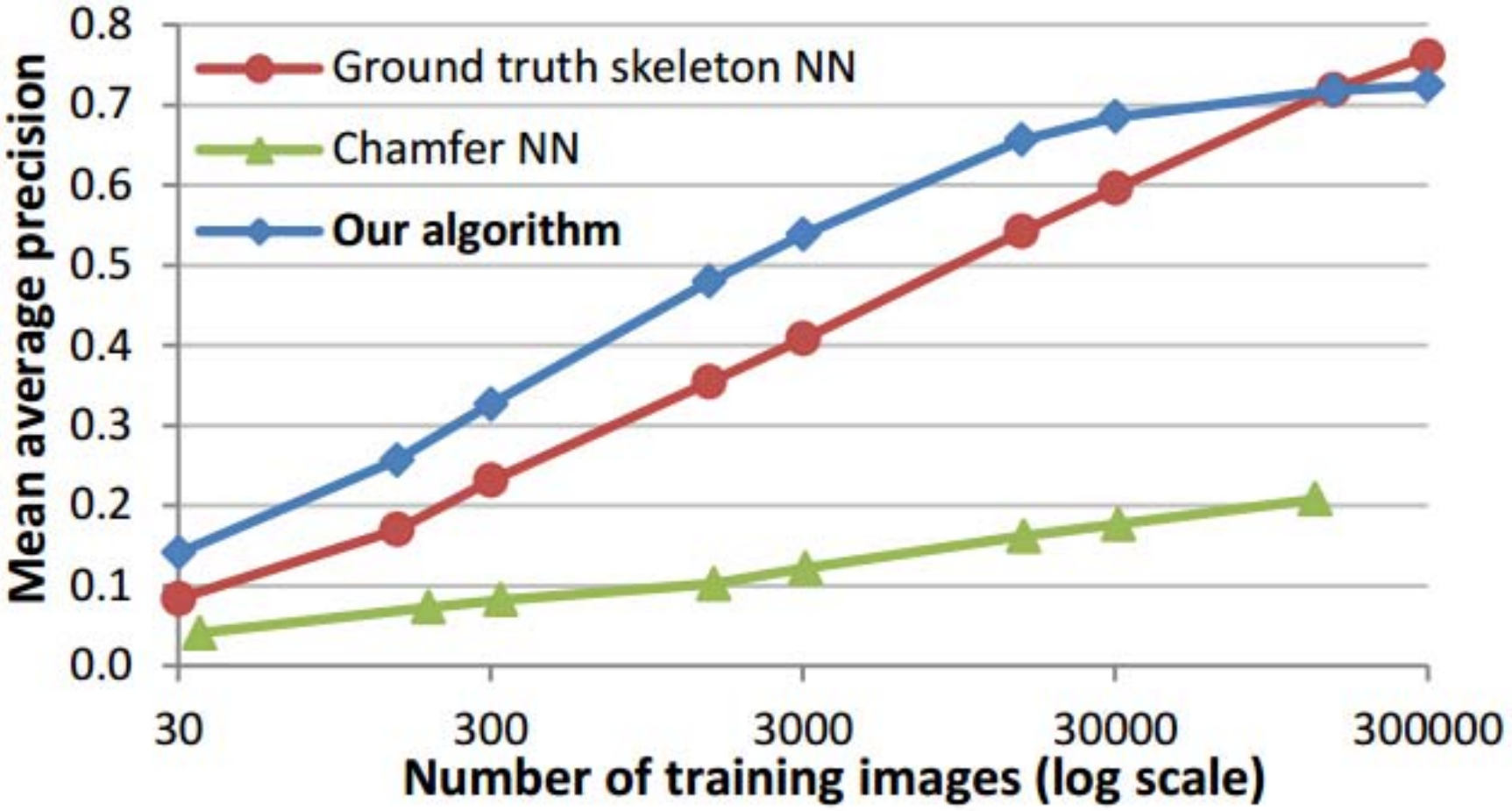


Figure 7. **Joint prediction accuracy.** We compare the actual performance of our system (red) with the best achievable result (blue) given the ground truth body part labels.

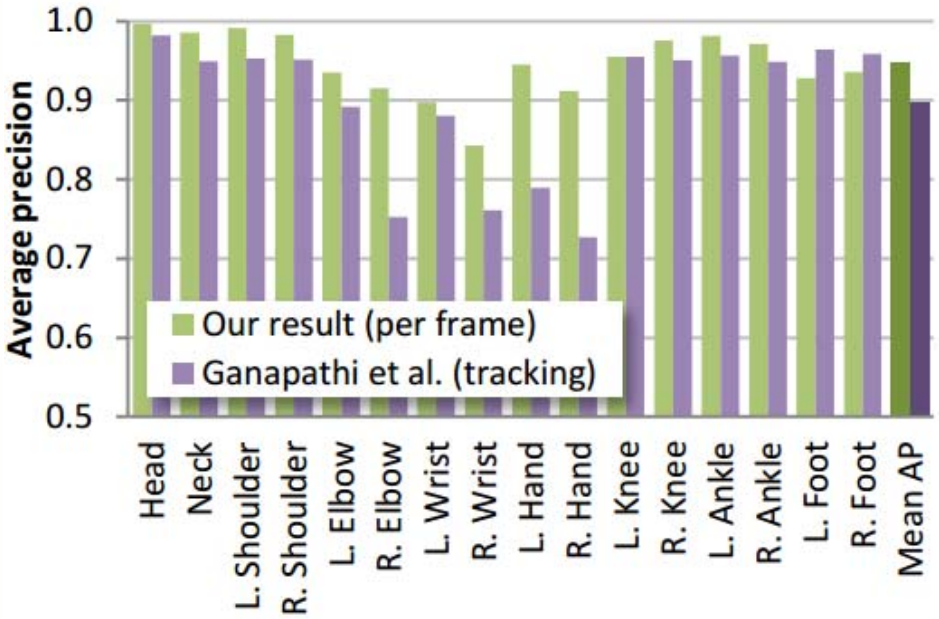
Comparison with Nearest Neighbor

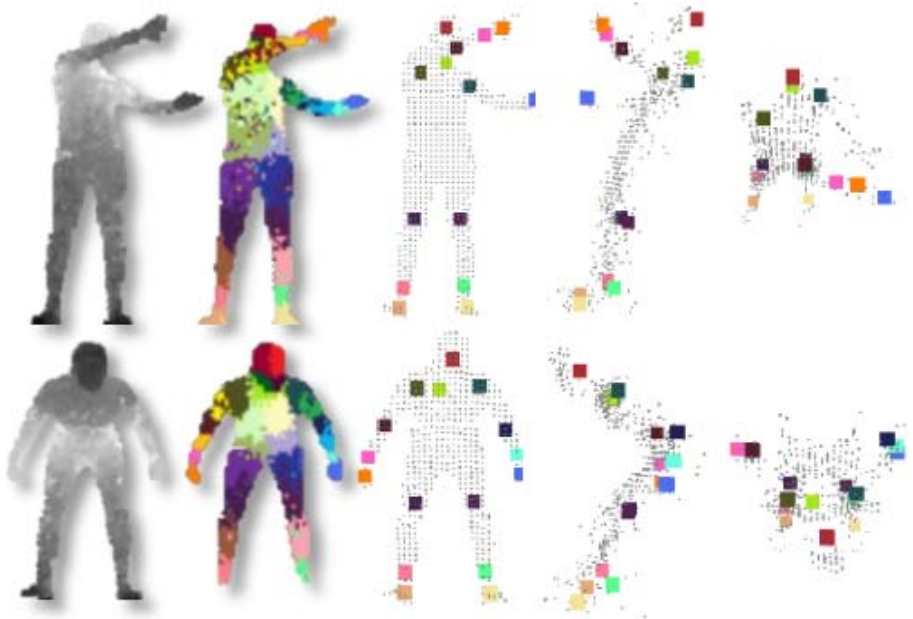
- Compares the joint configuration of the test image with configurations in training images.
- NN is slow (needs to search all training set)



Comparison

- V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. CVPR, 2010





Conclusion

- Fast joint position estimation from depth images
- Fast
  - 3 random trees of depth 20
  - At most 60 feature evaluations per pixel
  - 200 fps on Xbox
  - 50 fps on a computer
- Accurate
  - Trained on 900k images
  - Tested on 8k images

References

- J Shotton, T Sharp, A Kipman, A Fitzgibbon… - Real-time human pose recognition in parts from single depth images. Communications of the ACM, 2013