Real-Time Human Pose Recognition in Parts from Single Depth Images

PRESENTATION: JESSE DAVIS

CS 3710 VISUAL RECOGNITION

Outline

- Data Gathering
  - Real Data vs Synthetic Data
- Body Part Inference and Joint Proposals
  - Intermediate Body Parts Representation
  - Depth image comparison and features
  - Random Decision Forests/Trees
  - Joint Positioning
- Experiments

Main Idea/Problem Addressed

- ✓Real-time human pose recognition using consumer quality machine
- ✓Invariance to:
  - ✓Shape
  - ✓Size
  - ✓Pose
  - ✓Clothing
  - ✓Etc.

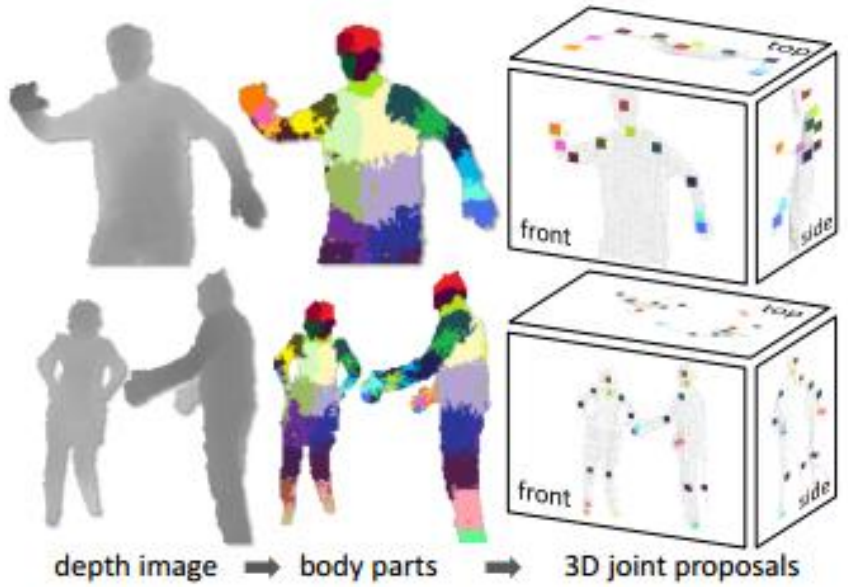


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of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

Problem Relevance

- Provides opportunity for:
  - monetary gain
  - quality of life improvement
  - improved machine-computer interaction
- Examples: Microsoft Kinect, Pedestrian Tracking, Computer-Interfaces

Challenges

- Human Body is capable of **many** different poses
- Need a way to generate many of these poses with different body shapes, scales, clothing types, etc. in mind
- Must decide how to identify parts of the body, and how detailed our labeling should be (i.e. leg vs. lower leg, knee, and upper leg)
- Backgrounds, light levels, color, and texture invariance

Previous Approaches

- Using Conventional Intensity Cameras (CPU expensive)
  - Examine/learn an initial pose => Learn variations of that pose
  - Estimate/search for locations of body segments from which to build the body
- **Using Depth Cameras**
  - Base their search off of 3D models, divide the model into parts, base their search off of those parts

Their Approach

- Large and varied dataset
  - Both synthetic and motion captured data
- Object Recognition Approach (by parts)
- Intermediate Body Part Representation
- Pose Estimation reduced to Per-Pixel Classification
- Create scored proposals of body joints

Data Gathering

- Depth Imaging Benefits
  - Color and texture invariance
  - Does well in low light levels
  - Gives accurate scaling estimates
  - Resolves silhouette ambiguity
  - Background subtraction
  - Creates a base for synthesizing additional depth images

Data Types

synthetic
(train & test)

real
(test)

Real Data (Motion Capture Data)

- Large Database built using motion capture and human actions related to target application (dancing, running, etc.)
- Classifier expected to generalize unseen poses
- Wide range of poses vs. all possible combinations
  - Many redundant poses are discarded based on initial data and “furthest neighbor” clustering
  - 100k poses (frames) used s.t. no poses are within 5 cm of one another

Synthetic Data

- Goals: **Realism and Variety**
- Use of **randomized rendering pipeline**, which produces samples that are fully labeled and can be trained on
- Learning is used to provide invariance towards: camera pose, body pose, body size, and body shape
  - Other slight variations accounted for: height, weight, mocap frame, camera noise, clothing, hairstyle

The Rendering Pipeline

- Necessary to account for pose variation and model variation
- Start with: Base Character and Pose
  - Transform on: Rotation and Translation; Hair and Clothing; Weight and Height Variations; Camera Position and Orientation; Camera Noise
  - Add transformations to dataset

Different Renderings

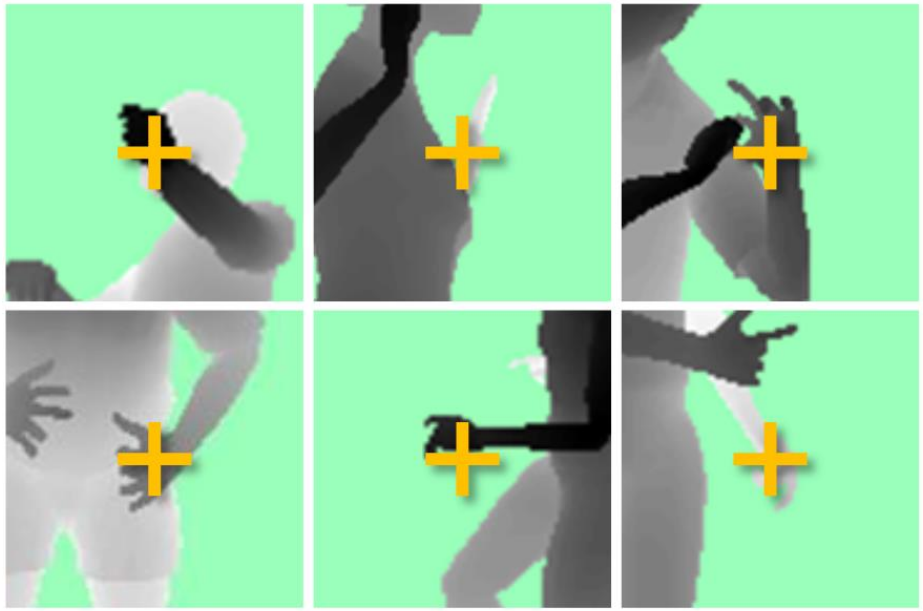


Figure 1. **Differences in local appearance within one body part.**

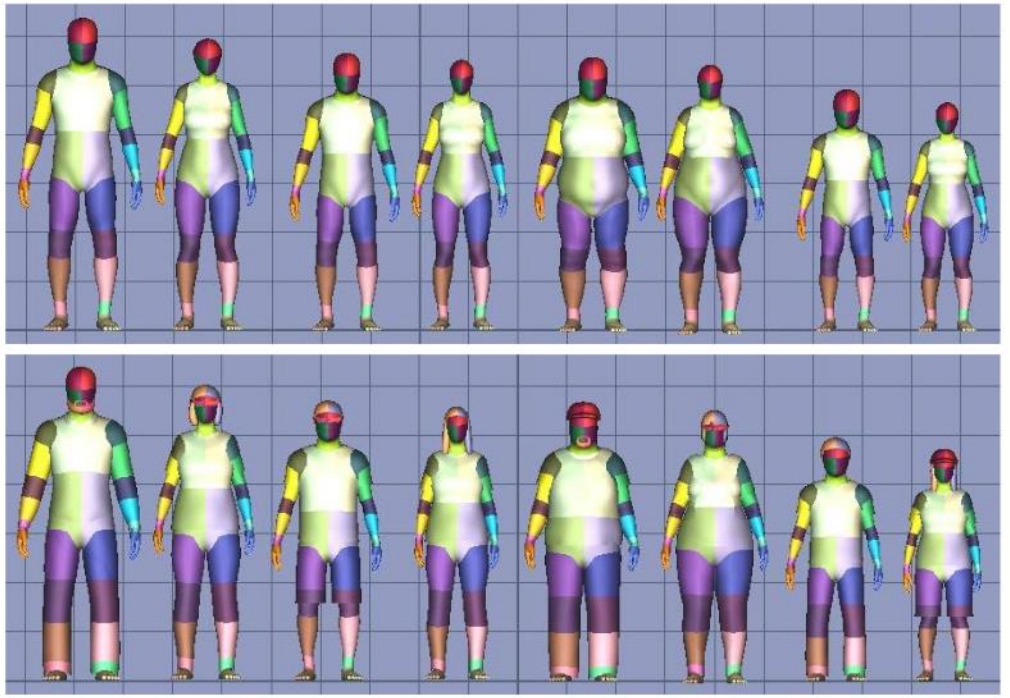


Figure 2. **Renders of several base character models.** Top row: without skinning. Bottom row: with random skinning of hair and clothing.

Rendering Pipeline…

- Steps:
  1) Sample a set of parameters (randomly)
  2) Render depth and body part images based on 3D meshes
  3) Using Autodesk Motionbuilder, the mocap retargets to the base 15 meshes of the body's shapes and sizes
  4) Further randomized variations to parameters (to expand pose coverage)

Body Part Labeling

- Body parts broken down into an intermediate body part representation of 31 body parts (object by parts)
- Observations:
  - Parts should be small to enough to localize different body joints
  - Parts should be small in numbers such that no classifier space is wasted

Depth Image Features

For a given pixel x, the features are computed via:

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

$d_I(x)$: the depth at pixel x in Image

$\theta = (u, v)$: describes offsets u and v

$\frac{1}{d_I(x)}$: ensures features are depth invariant

Depth Image Features…

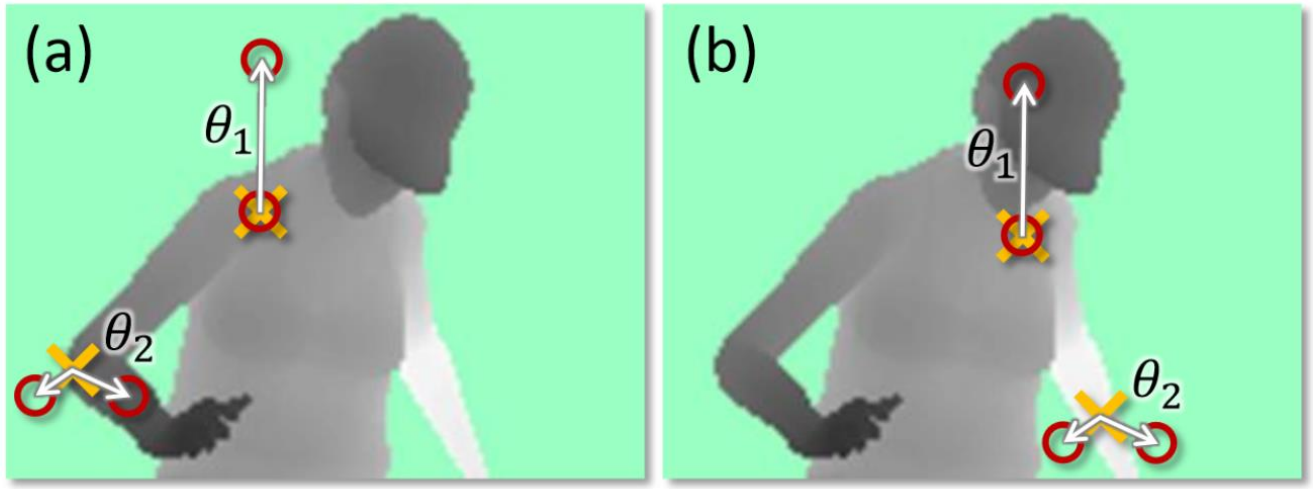


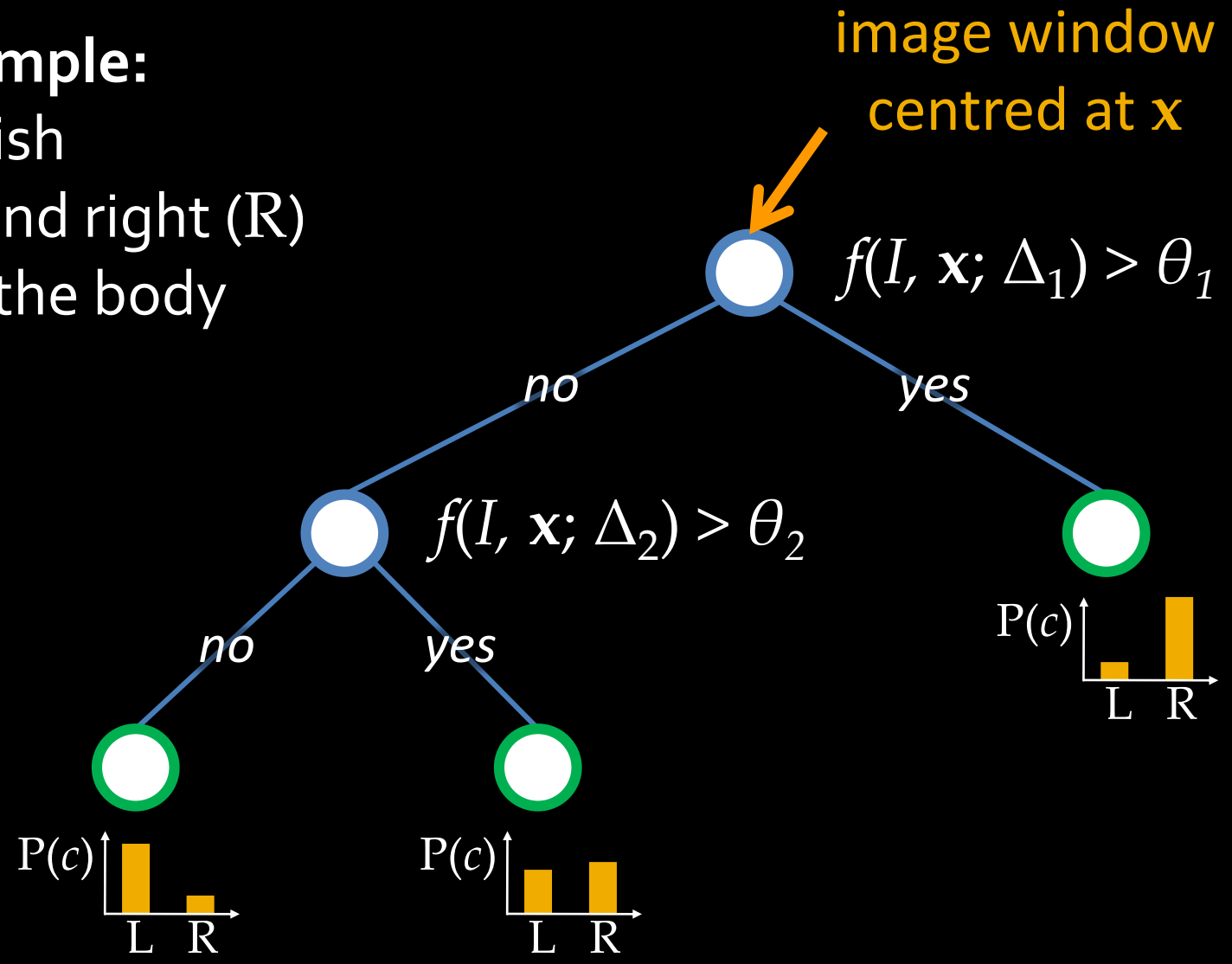
Figure 3. **Depth image features.** The yellow crosses indicates the pixel $\mathbf{x}$ being classified. The red circles indicate the offset pixels as defined in Eq. 1. In (a), the two example features give a large depth difference response. In (b), the same two features at new image locations give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

Depth Image Features…

- Features are weak individually
- Solution: Combine with a decision forest
- Solution is efficient:
  - One feature reads **at most** 3 image pixels
  - Performs **at most** 5 arithmetic operations
  - Can be implemented on a GPU

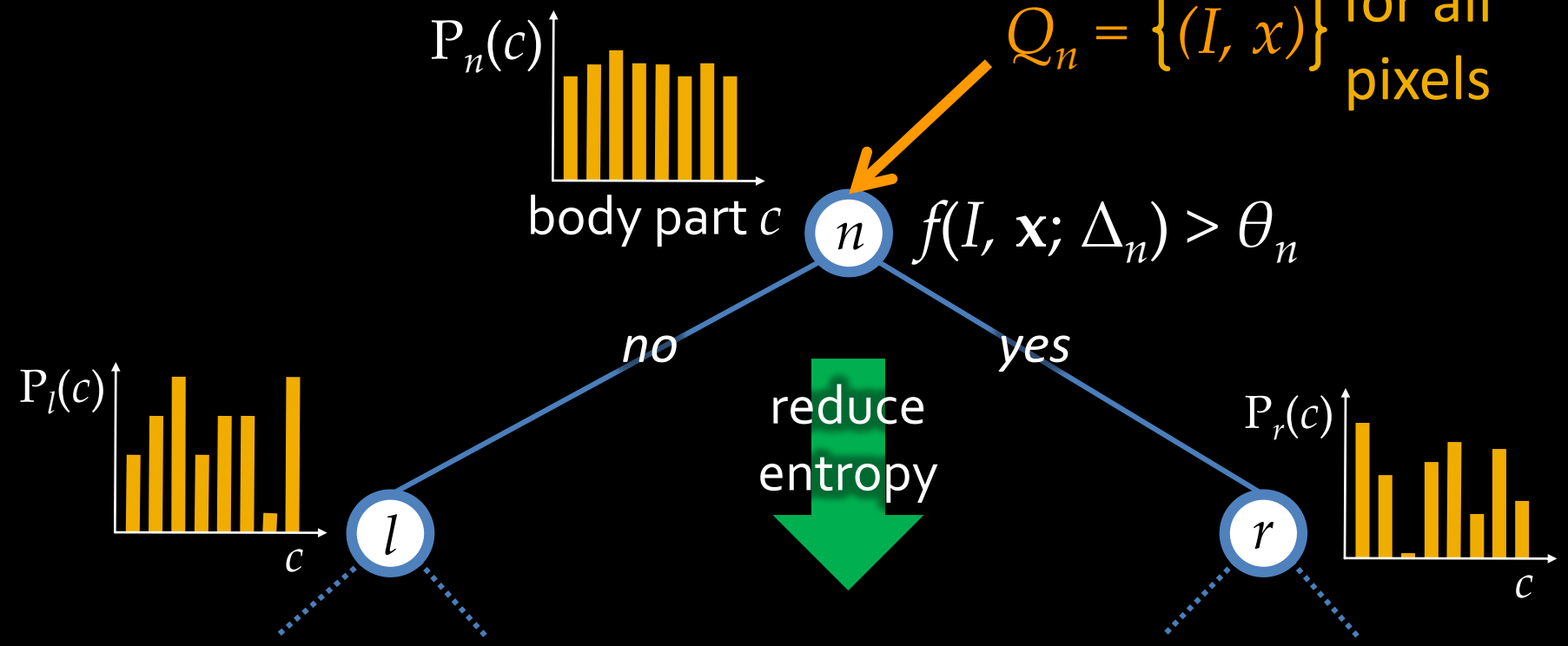
Decision tree classification

Toy example:
distinguish
left (L) and right (R)
sides of the body



Training decision trees

[Breiman *et al.* 84]



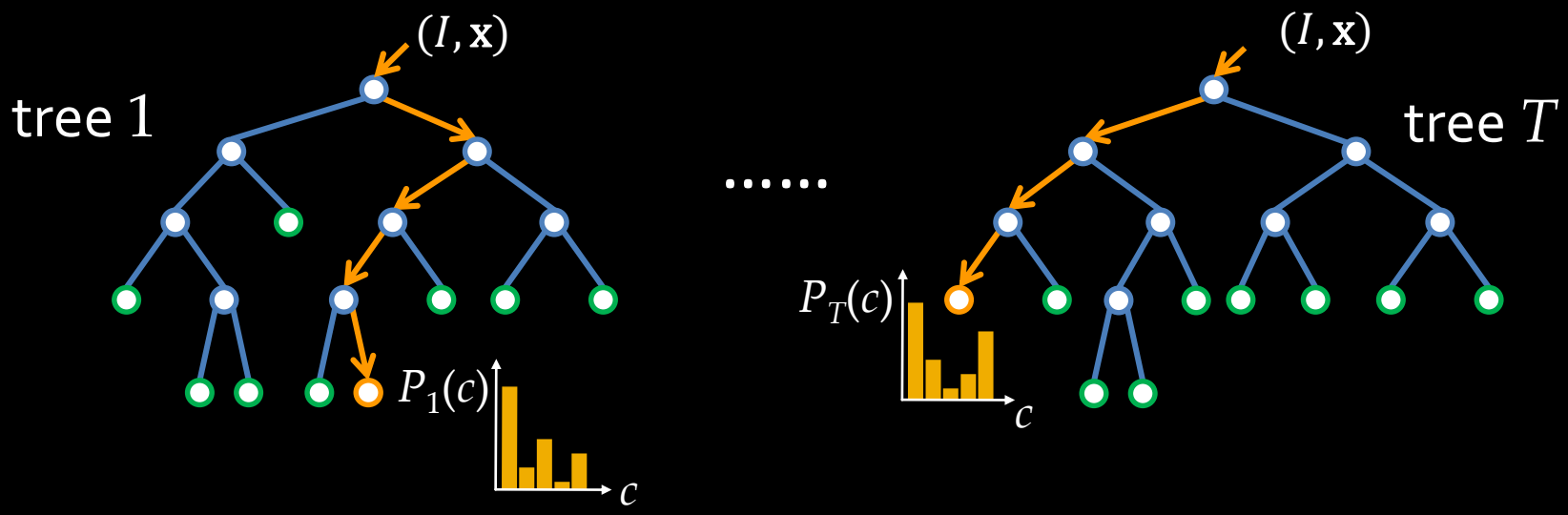
Take (Δ, θ) that maximises information gain:

$$\Delta E = -\frac{|Q_\mathrm{l}|}{|Q_n|} E(\mathrm{Q_l}) - \frac{|Q_\mathrm{r}|}{|Q_n|} E(\mathrm{Q_r})$$

Goal: drive entropy at leaf nodes to zero

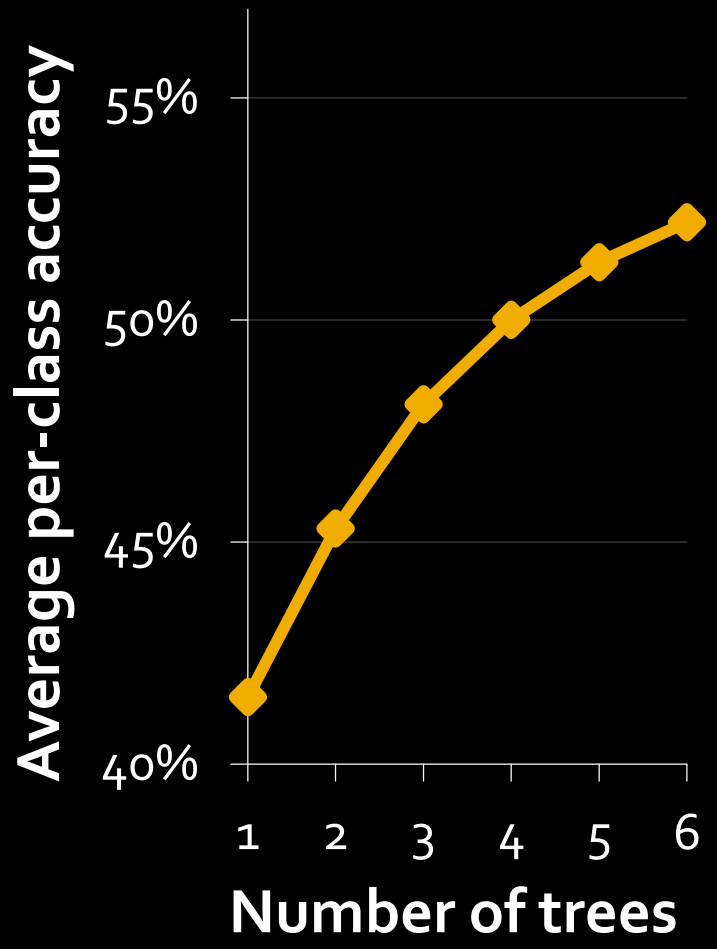
Decision forest classifier

[Amit & Geman 97]
[Breiman 01]
[Geurts *et al.* 06]



- Trained on different random subset of images
  - "bagging" helps avoid over-fitting
- Average tree posteriors $P(c|I,\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,\mathbf{x})$

Number of trees



ground truth

inferred body parts (most likely)

1 tree

3 trees

6 trees

Randomized Decision Forests...

- Consist of T decision trees
- Split nodes:
  - Have a feature f(θ) and threshold τ
  - To classify pixel x from image I, you start at the root and evaluate equation (1), branching left or right according to the set t
- Leaf nodes:
  - A learned distribution (left side of equation) of body part labels c is stored
  - Distributions are averaged for all trees to give the final classification decision

$$P(c|I,\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,\mathbf{x})\ . \qquad (2)$$

Training using the decision forest

Trees are trained on different sets of randomly synthesized images using the following algorithm:

1. Randomly propose a set of splitting candidates $\phi = (\theta, \tau)$ (feature parameters θ and thresholds τ).
2. Partition the set of examples $Q = \{(I, \mathbf{x})\}$ into left and right subsets by each ϕ:

$$Q_{\mathrm{l}}(\phi) = \{ (I, \mathbf{x}) \mid f_\theta(I, \mathbf{x}) < \tau \} \quad (3)$$

$$Q_{\mathrm{r}}(\phi) = Q \setminus Q_{\mathrm{l}}(\phi) \quad (4)$$

3. Compute the ϕ giving the largest gain in information:

$$\phi^\star = \operatorname*{argmax}_\phi G(\phi) \quad (5)$$

$$G(\phi) = H(Q) - \sum_{s \in \{\mathrm{l},\mathrm{r}\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)) \quad (6)$$

where Shannon entropy $H(Q)$ is computed on the normalized histogram of body part labels $l_I(\mathbf{x})$ for all $(I, \mathbf{x}) \in Q$.

4. If the largest gain $G(\phi^\star)$ is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets $Q_{\mathrm{l}}(\phi^\star)$ and $Q_{\mathrm{r}}(\phi^\star)$.

Joint Position Proposal

- Body Part Recognition provides per-pixel information
  - Must utilize this to generate reliable proposals for joint proposals
- Method: local mode-finding approach based on mean shift with a weighted Gaussian kernel
- Density estimator per body part equation:

$$f_c(\hat{\mathbf{x}}) \propto \sum_{i=1}^{N} w_{ic} \exp\left(-\left\|\frac{\hat{\mathbf{x}} - \hat{\mathbf{x}}_i}{b_c}\right\|^2\right), \qquad (7)$$

$\hat{x}$: a coordinate in 3D space

$\hat{x}_i$: re-projection of image pixel x_i into space given

N: the number of image pixels

b_c: learned per-part bandwidth

w_{ic}: pixel weighting

Joint Position Proposal...

Pixel weighting w_{ic} is defined by:

$$w_{ic} = P(c|I, \mathbf{x}_i) \cdot d_I(\mathbf{x}_i)^2 \,. \qquad (8)$$

- Accounts for:
  - Inferred Body Part Probability: $P(c|I, x_i)$
  - World surface area of the pixel: $d_I(x_i)^2$

Body parts to joint hypotheses

- Define 3D world space density:

$$f_c(\hat{\mathbf{x}}) \propto \sum_{i=1}^{N} w_{ic} \exp\left(-\left\| \frac{\hat{\mathbf{x}} - \hat{\mathbf{x}}_i}{b_c} \right\|^2\right)$$

3D coord

pixel weight

3D coord of *i* th pixel

pixel index *i*

bandwidth

$$w_{ic} = P(c|I, \mathbf{x}_i) \cdot d_I(\mathbf{x}_i)^2$$

inferred probability

depth at *i* th pixel

- Mean shift for mode detection

3. hypothesize body joints

...

Joint Position Proposal (cont)

- Key Details:
  - Density estimates are depth invariant (8)
  - Depending on the application, inferred body parts can be pre-accumulated, e.g. multiple body parts over the same area can be merged to form a localized joint
  - Mean shift finds modes efficiently
  - Pixels above a selected learned probability curve are used for starting points of c (7, 8)
  - Final joint estimate: sum of the pixel weights reaching their given mode

Experiments

- Test Data:
  - Set of 5000 synthesized depth images (original poses used to generate are left out)
  - Real dataset of 8808 frames from more than 15 different subjects
  - Depth data from [13] (28 depth image sequences ranging from short motion to full actions)

Parameters and Metrics

- 3 trees, depth of 20, 300k training images/tree, 2000 training example pixels/image, 2000 candidate features (θ), 50 candidate thresholds (τ)/feature
- Quantification of classification and joint prediction accuracy
  - Classification: average per-class accuracy (all body parts =)
  - Joint Prediction: average precision per joint using mean avg precision

Joint Prediction

- Given a ground truth position: the first joint proposal within D meters (D = 0.1m) is assumed to be the true positive, while all others are false positives
- Joint proposals outside D are also false positives
- All proposals counted, not just confident ones
- Invisible joints not penalized

Qualitative Results

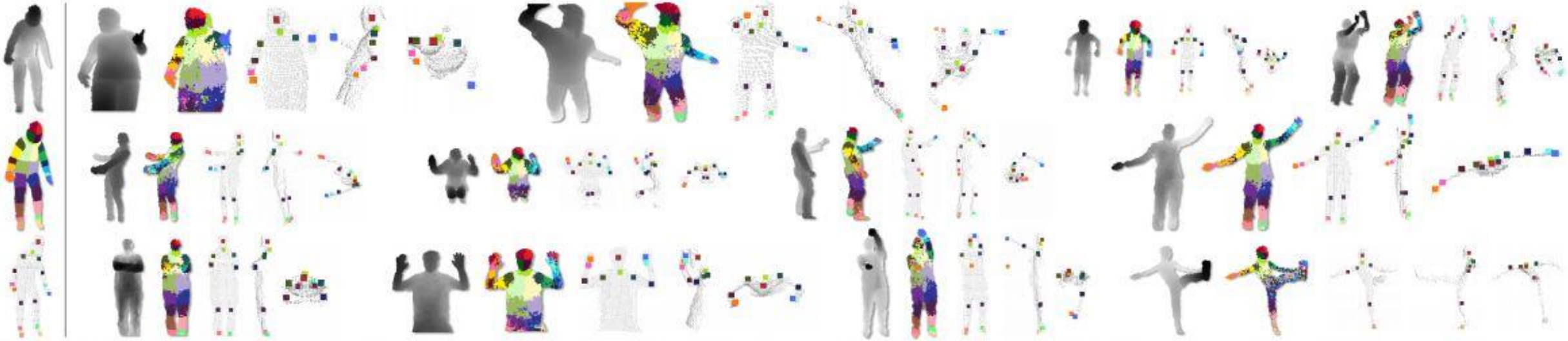


Figure 5. **Example inferences.** Synthetic (top row); real (middle); failure modes (bottom). Left column: ground truth for a neutral pose as a reference. In each example we see the depth image, the inferred most likely body part labels, and the joint proposals show as front, right, and top views (overlaid on a depth point cloud). Only the most confident proposal for each joint above a fixed, shared threshold is shown.

Classification Accuracy and Metrics

- Metrics:
  - # of training images
  - Synthesized silhouette images
  - Depth of trees
  - Max probe offset

Classification Accuracy and Metrics…

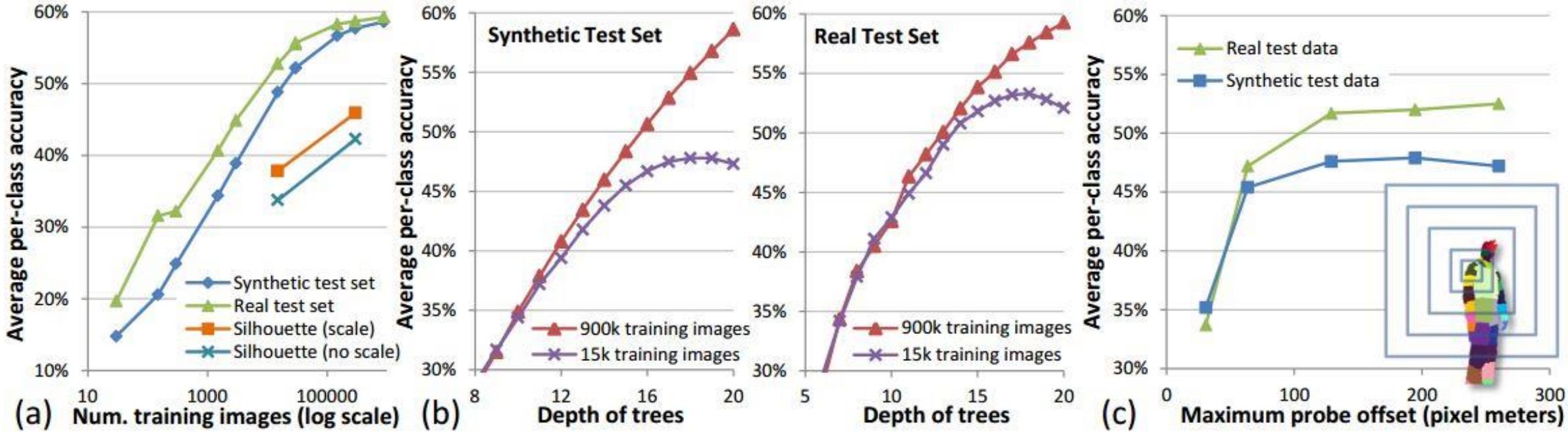


Figure 6. **Training parameters *vs*. classification accuracy.** (a) Number of training images. (b) Depth of trees. (c) Maximum probe offset.

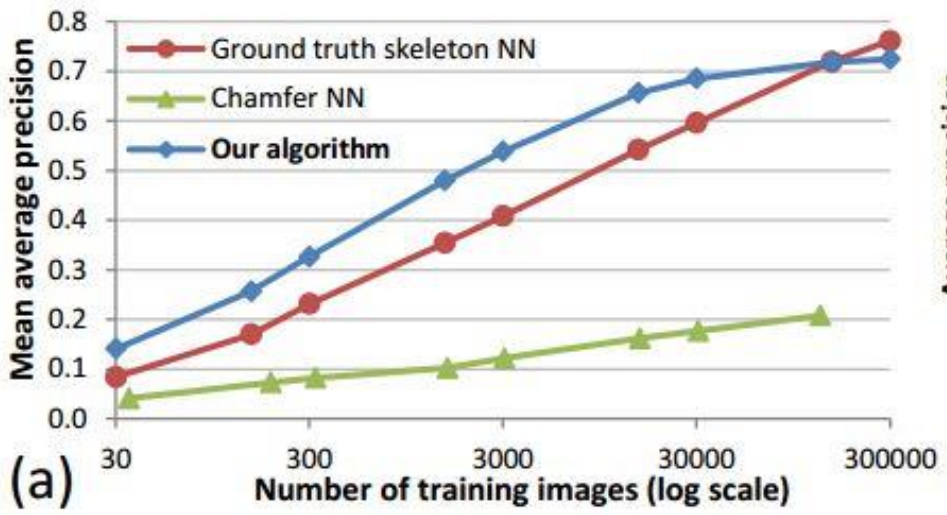
Joint Prediction Accuracy Comparisons

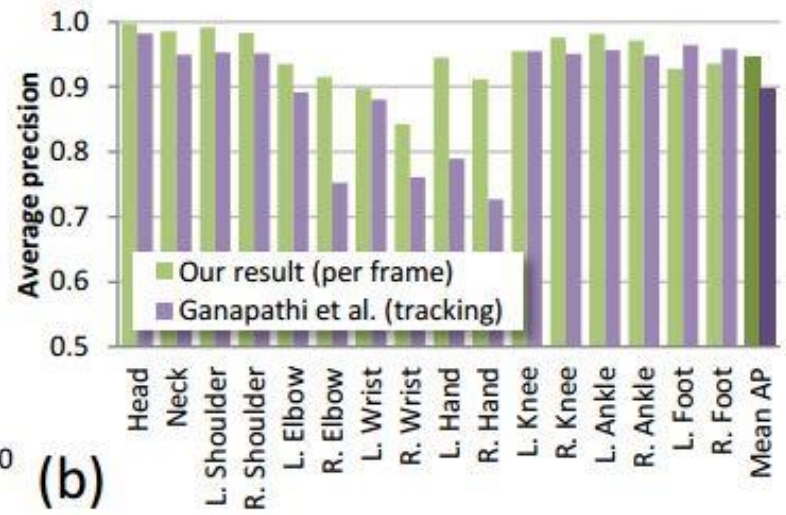
- Nearest Neighbor Comparison, two variants:
  - Var. 1: matches ground truth test skeleton to training skeletons with translational alignment in 3D space (note: no access to test skeleton in practice)
  - Var. 2: chamfer matching [14]: uses depth edges and 12 orientation bins

Joint Prediction Accuracy Comparisons...

- Versus [13]
  - [13]:
    - Uses body part proposals from [28]
    - Tracks the skeleton with kinematic and temporal info.
    - Data from time-of-flight depth camera
  - This paper's algorithm has improved joint prediction (using AP), and runs at least 10x faster (no values given)

Joint Prediction Accuracy Comparisons…





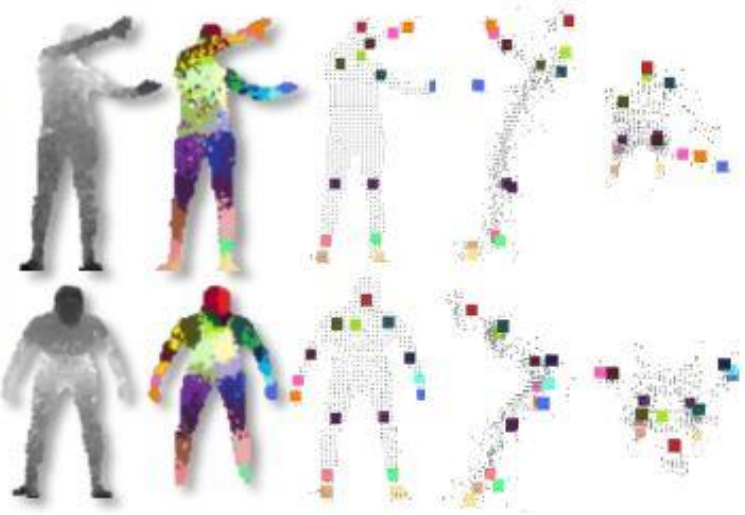


Figure 8. **Comparisons.** (a) Comparison with nearest neighbor matching. (b) Comparison with [13]. Even without the kinematic and temporal constraints exploited by [13], our algorithm is able to more accurately localize body joints.

Full Rotations and Multiple People

- Decrease in mAP, but was still able to achieve an impressive accuracy of 65.5%
- Trained on 900k images with full rotations, 5k synthetic images with full rotations

Discussion

- Improvements?
- Weaknesses?
- Applications?
- Future work?

Resources/References

1. Articulated Pose Estimation using Flexible Mixtures of Parts. Y. Yang and D. Ramanan. CVPR 2011.
2. Real-Time Human Pose Recognition in Parts from a Single Depth Image. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. CVPR 2011.
3. *Openc Dev Team. "Meanshift and Camshift." Meanshift and Camshift — OpenCV 3.0.0-dev Documentation*. Openc Dev Team, n.d. Web. 20 Feb. 2015
4. Random Decision Forests. Tin Kam Ho.
5. V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In Proc. CVPR, pages 2:775–781, 2005.