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

Jamie Shotton, Andrew Fitzgibbon, Mat Cook,
Toby Sharp, Mark Finocchio, Richard Moore,
Alex Kipman, Andrew Blake

CVPR 2011

Microsoft Research

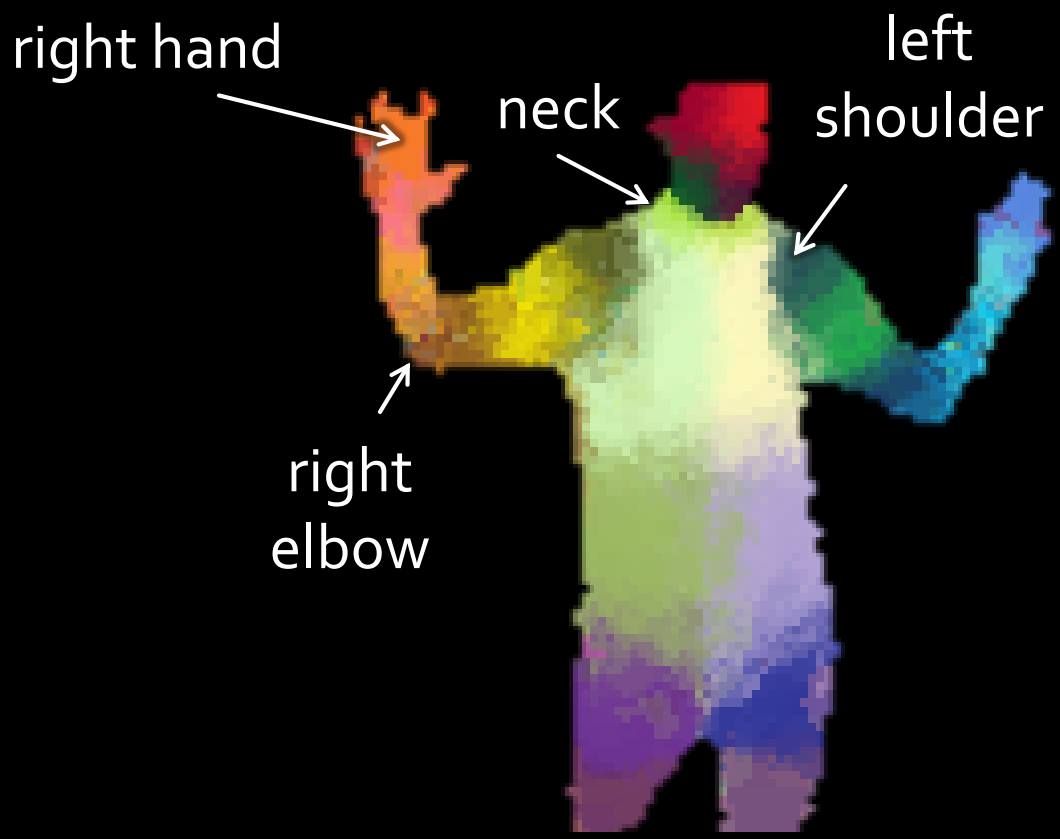
XBOX 360

The mission

Auto-initialize
a tracking algorithm
& recover from failures

- All human poses, shapes & sizes
- Limited compute budget
  - super-real time on Xbox 360
    to allow games to run concurrently

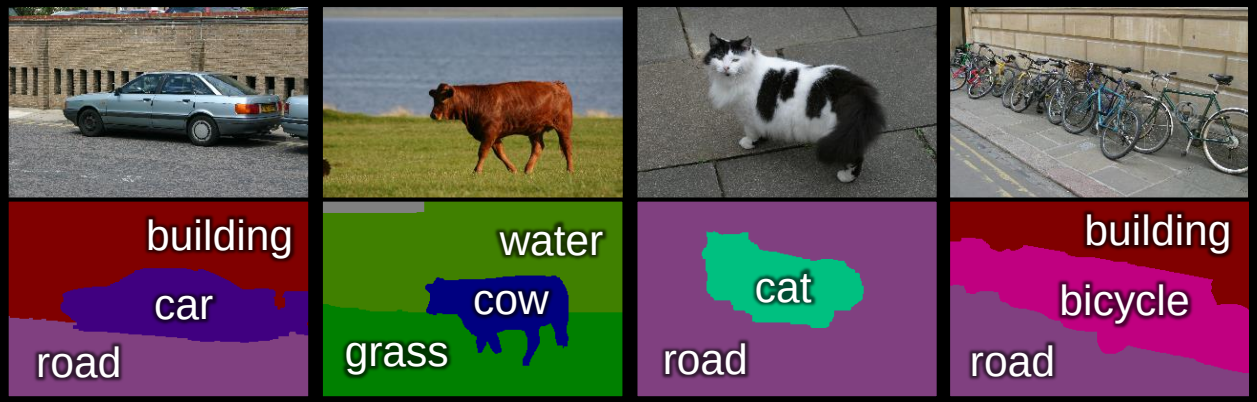
The approach: body part recognition



Body part recognition

- No temporal information
  - frame-by-frame
- Local pose estimate of parts
  - each pixel & each body joint treated independently
  - reduced training data and computation time
- Very fast
  - simple depth image features
  - parallel decision forest classifier

Object segmentation



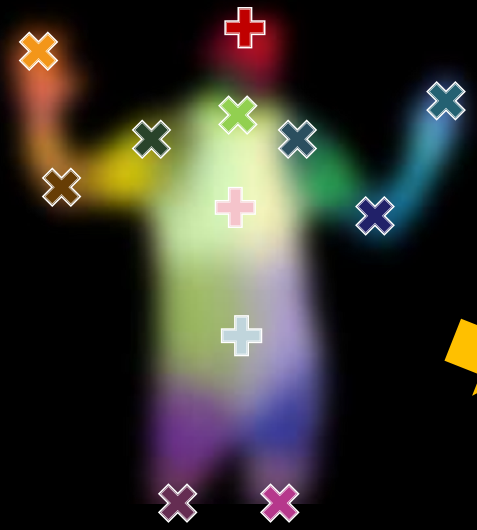
[Shotton, Winn, Rother, Criminisi 06 + 08]
[Winn & Shotton 06]

[Shotton, Johnson, Cipolla 08]

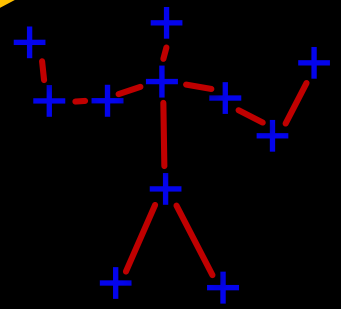
The Kinect pose estimation pipeline

capture
depth image &
remove bg

infer
body parts
per pixel



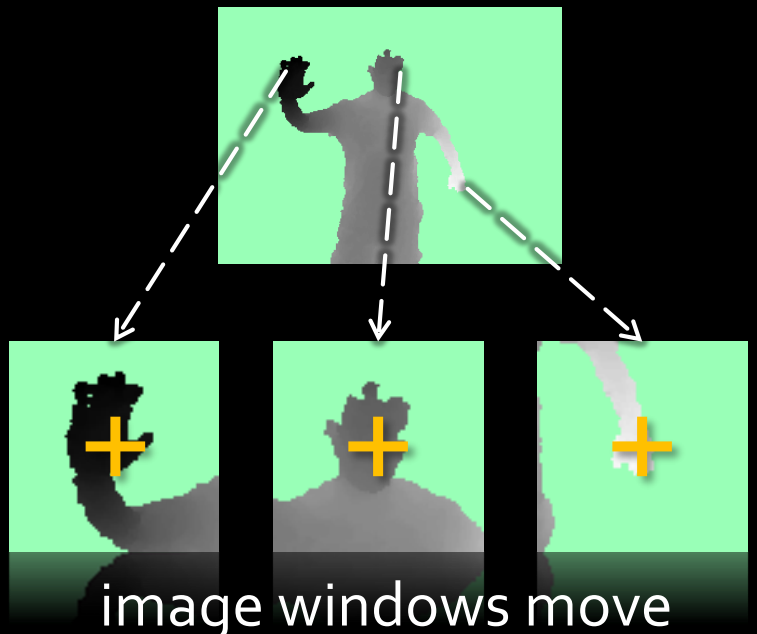
cluster pixels to
hypothesize
body joint
positions



fit model &
track skeleton

Classifying pixels

- Compute $\mathrm{P}(c_i \mid w_i)$
  - pixels $i = (x, y)$
  - body part c_i
  - image window w_i



- Discriminative approach
  - learn classifier $\mathrm{P}(c_i \mid w_i)$ from training data

Synthetic training data

Record mocap

500k frames
distilled to 100k poses

Retarget to several models

Render (depth, body parts) pairs

Train invariance to:

Synthetic vs real data

synthetic
(train & test)

real
(test)

Fast depth image features

- Depth comparisons
  - very fast to compute

feature response

$$f(I, \mathbf{x}) = d_I(\mathbf{x}) - d_I(\mathbf{x} + \Delta)$$

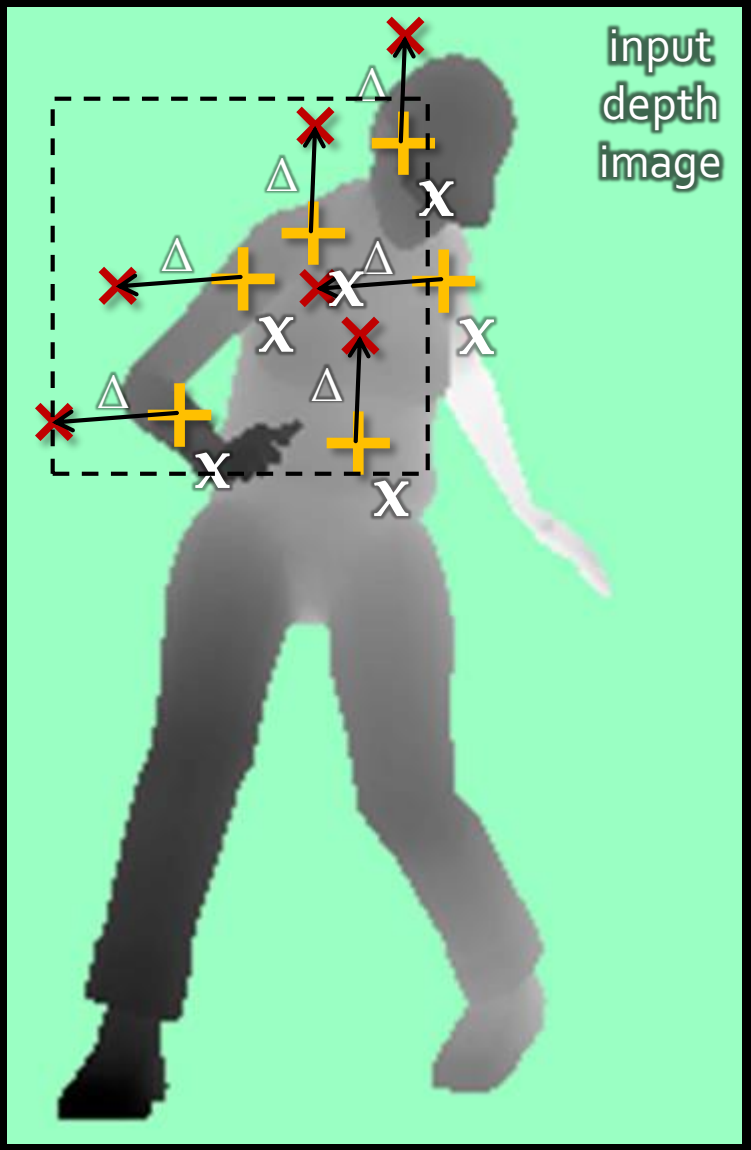
image coordinate

image: I; depth: $d_I(\mathbf{x})$; offset depth: $d_I(\mathbf{x} + \Delta)$

$$\Delta = \frac{\mathbf{v}}{d_I(\mathbf{x})}$$

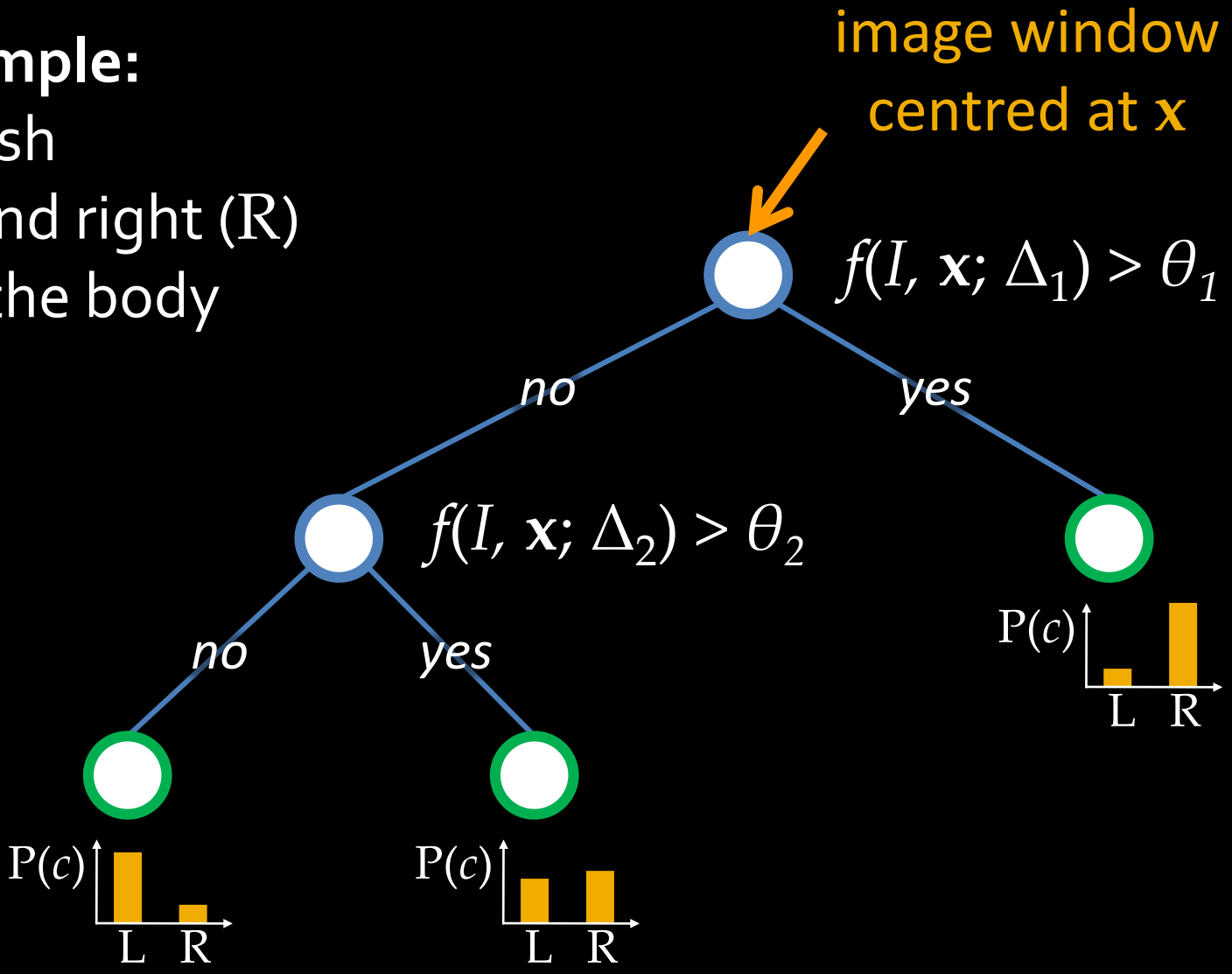
scales inversely with depth

Background pixels
d = large constant



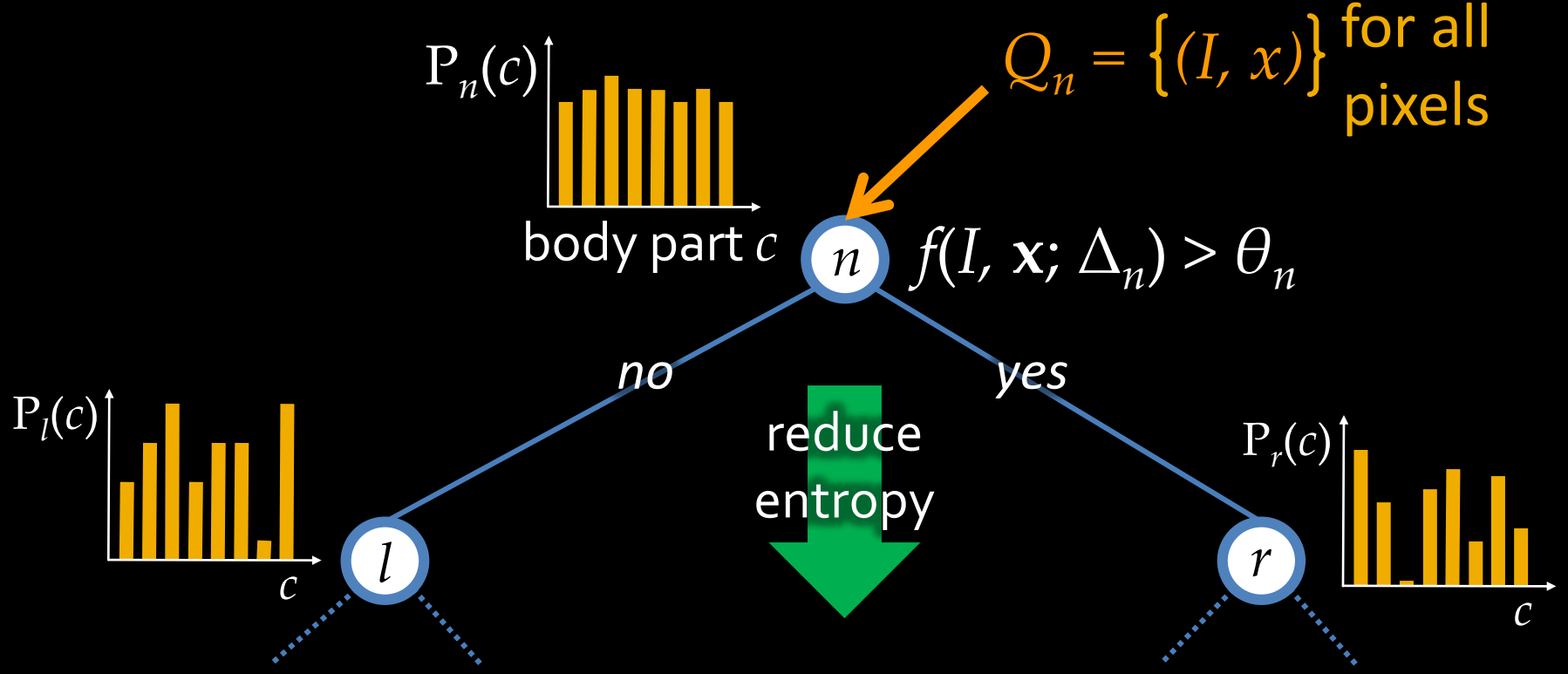
Decision tree classification

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:

$$\Delta E = -\frac{|Q_l|}{|Q_n|} E(Q_l) - \frac{|Q_r|}{|Q_n|} E(Q_r)$$

Goal: drive entropy at leaf nodes to zero

Depth of trees

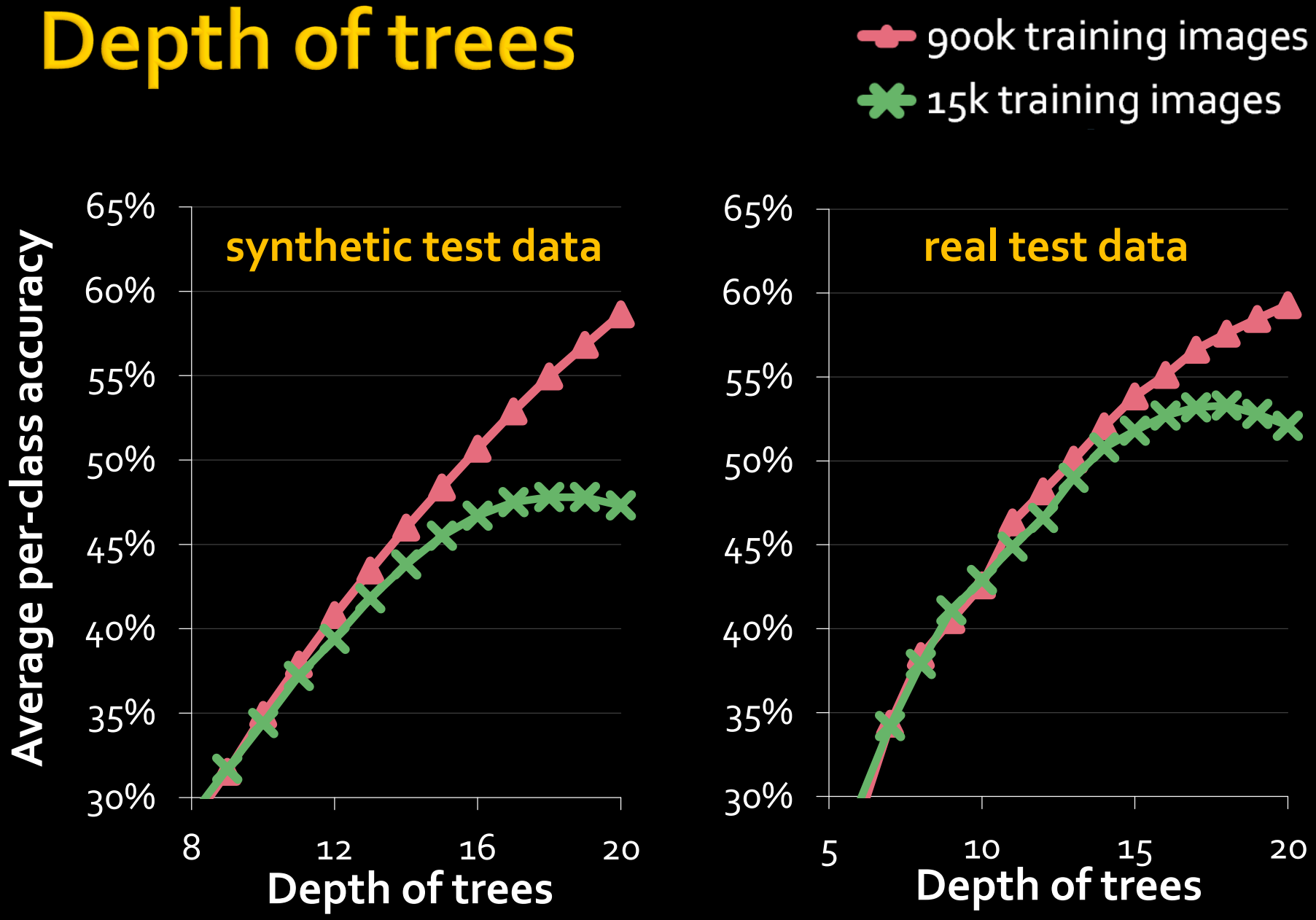
input depth

ground truth parts

inferred parts (soft)

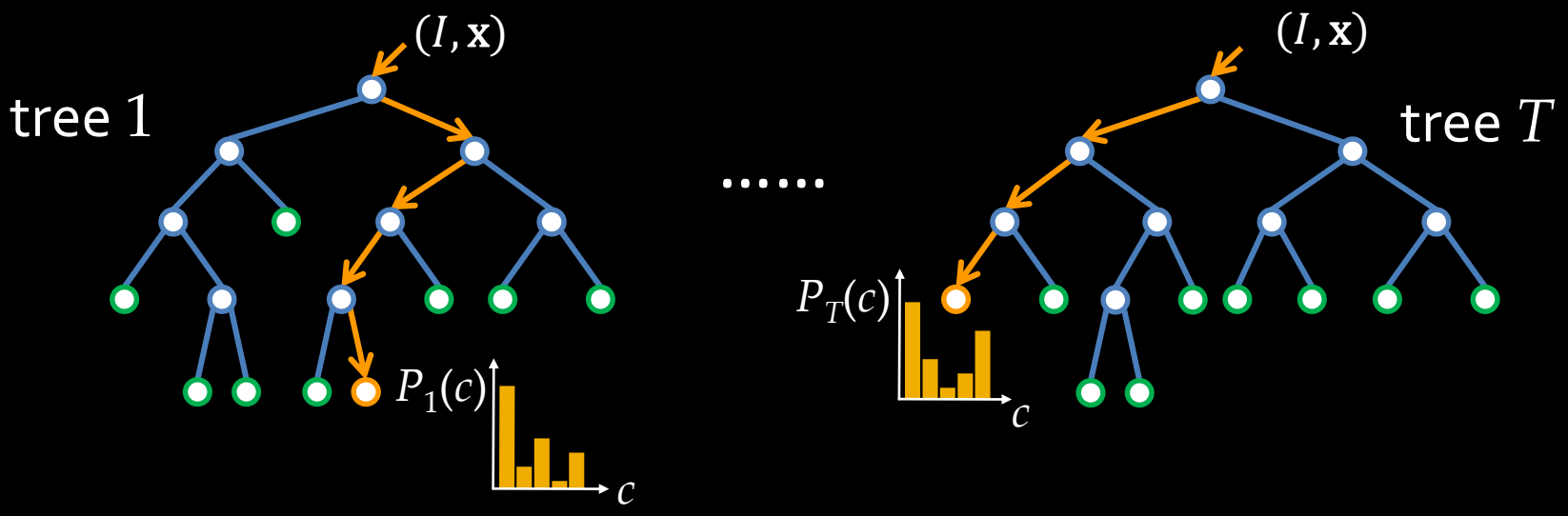
depth 18

Depth of trees



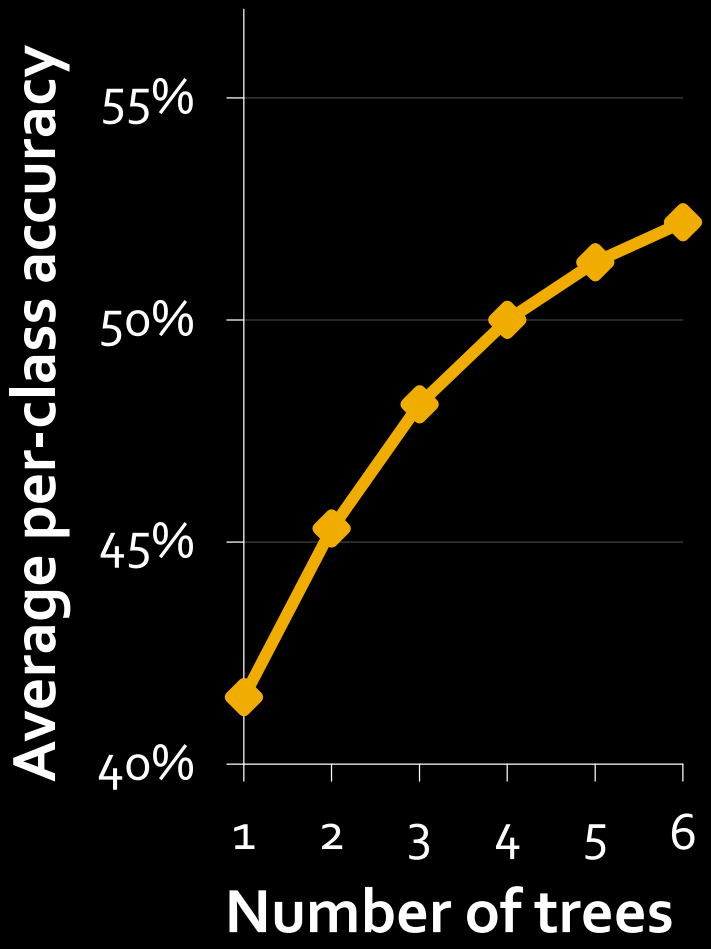
Decision forest classifier

[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

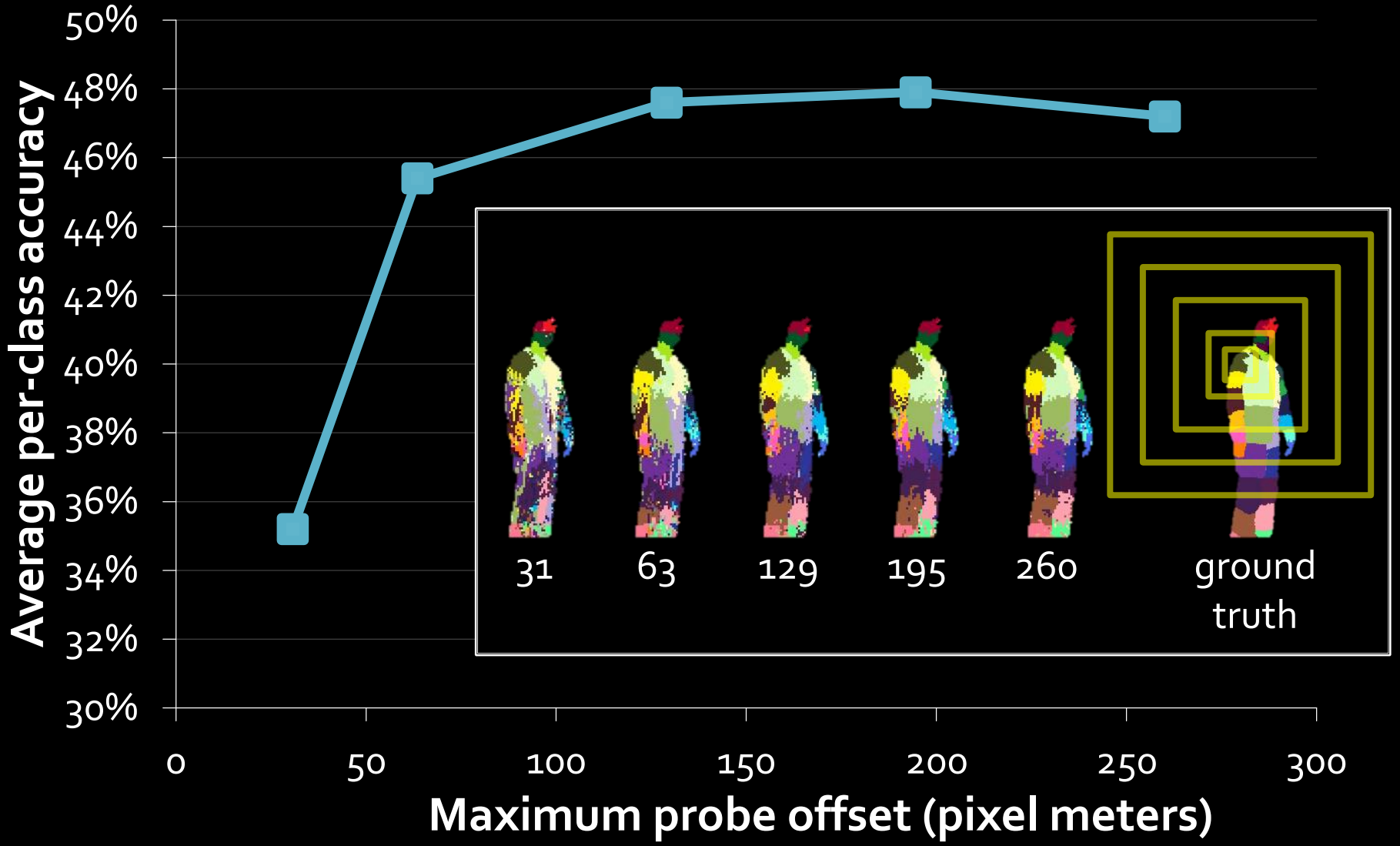
inferred body parts (most likely)

1 tree

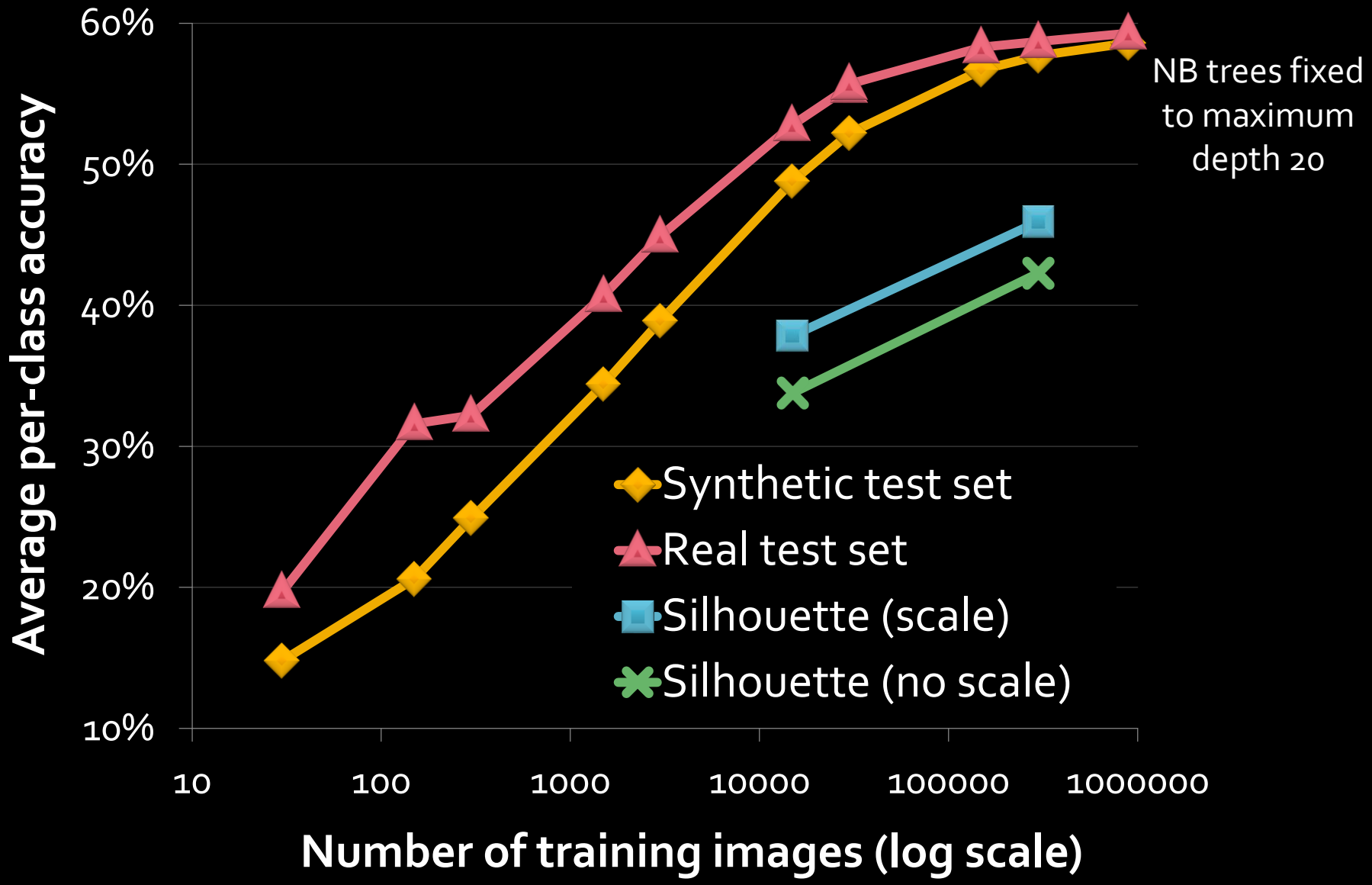
3 trees

6 trees

Feature window size



Number of training images



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

$$w_{ic} = \underbrace{P(c|I, \mathbf{x}_i)}_{\text{inferred probability}} \cdot \underbrace{d_I(\mathbf{x}_i)^2}_{\text{depth at } i^{\text{th}} \text{ pixel}}$$

- Mean shift for mode detection

3. hypothesize body joints

...

input depth

inferred body parts

front view

side view

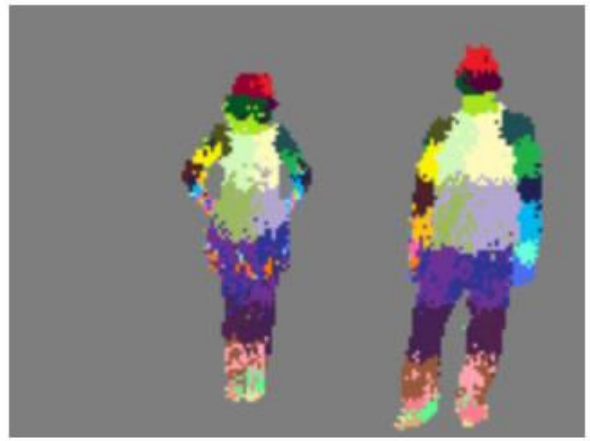
top view

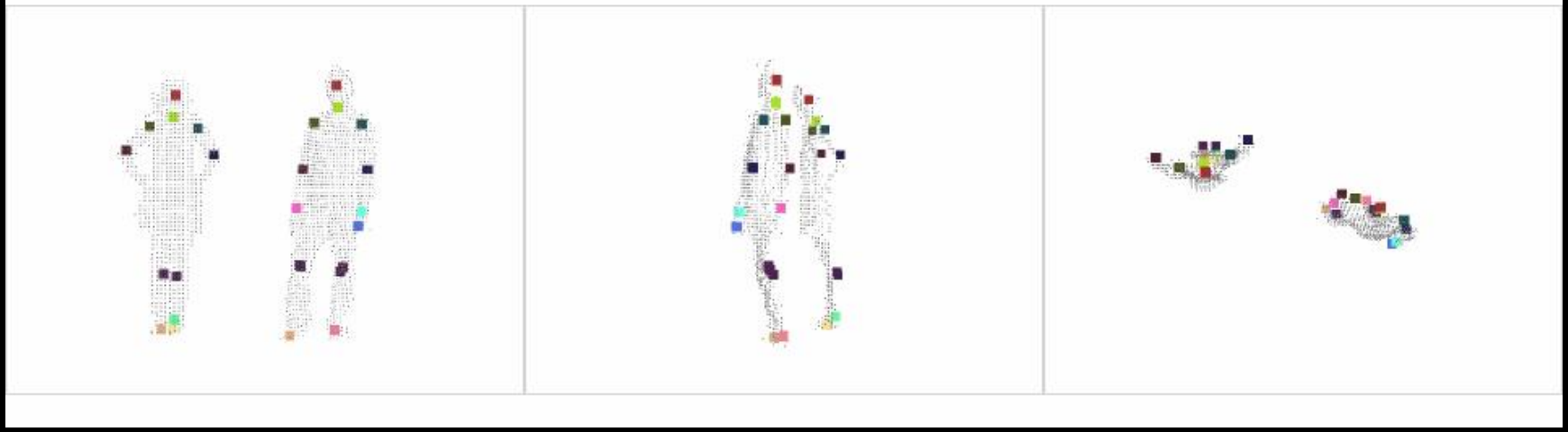
inferred joint positions

no tracking or smoothing

input depth

inferred body parts





front view

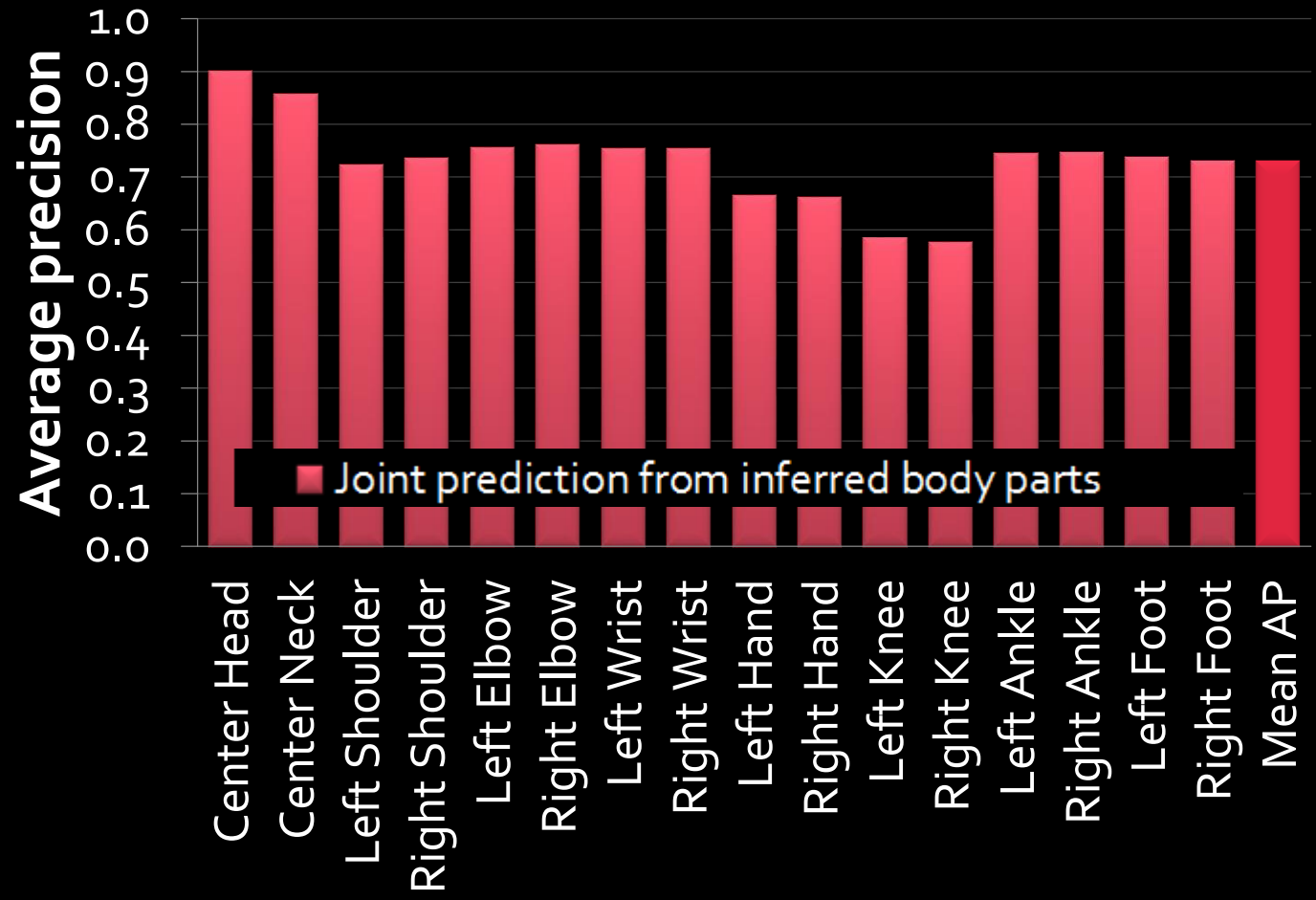
side view

top view

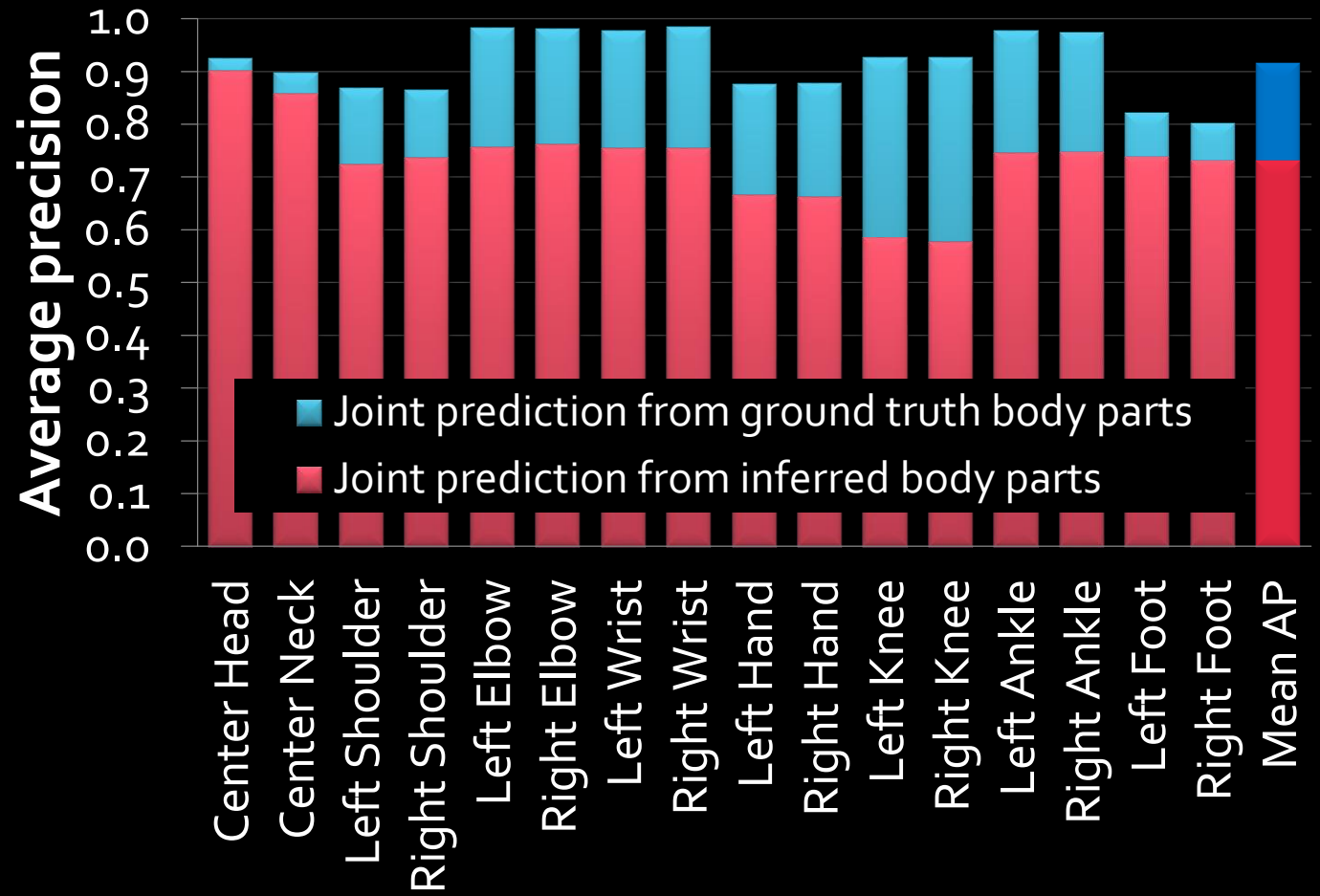
inferred joint positions

no tracking or smoothing

Joint prediction accuracy

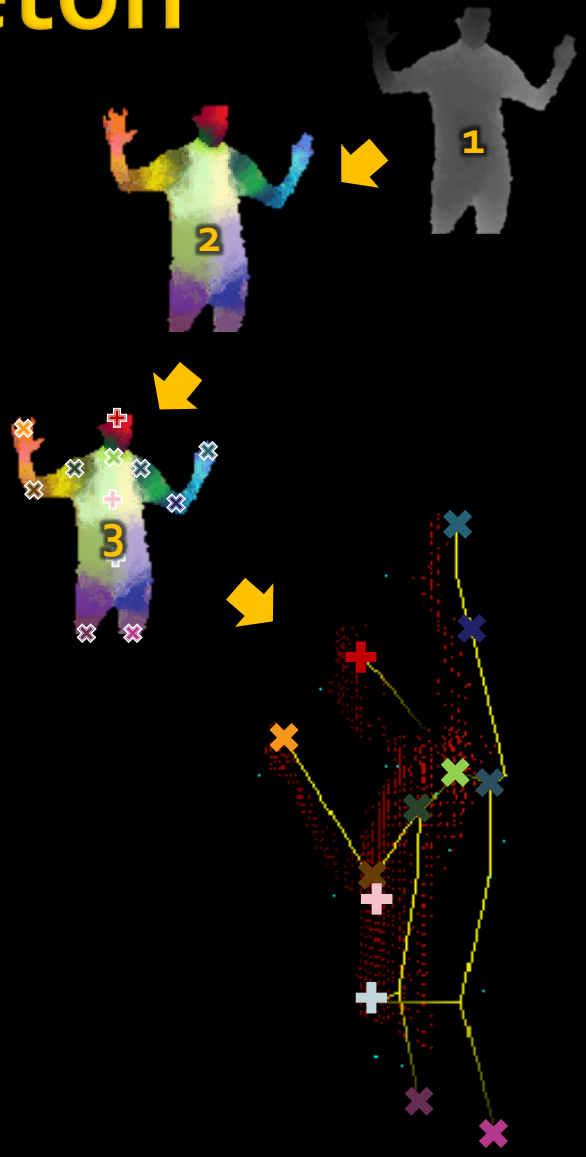


Joint prediction accuracy



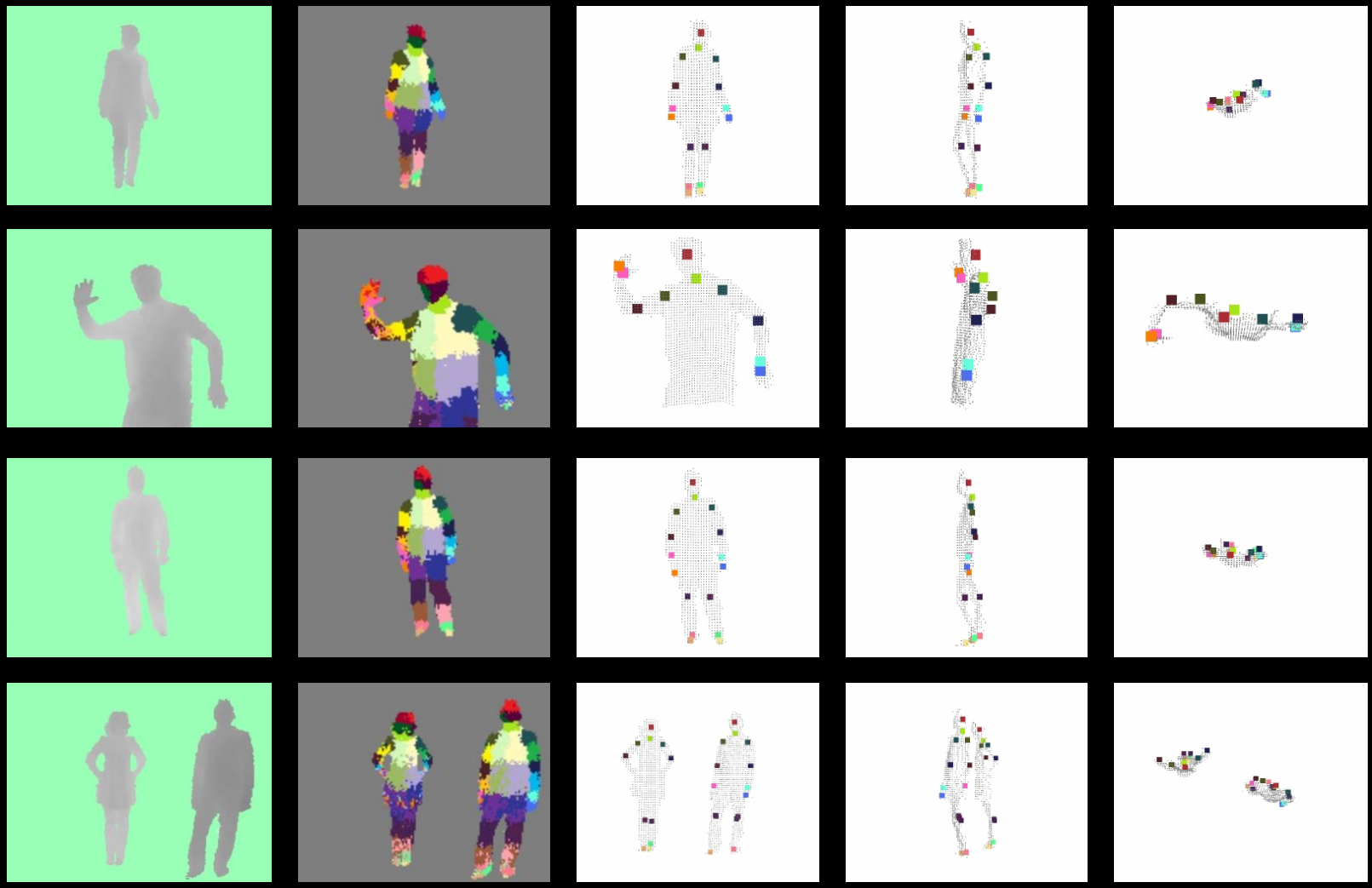
From proposals to skeleton

- Use…
  - 3D joint hypotheses
  - kinematic constraints
  - temporal coherence
- … to give
  - full skeleton
  - higher accuracy
  - invisible joints
  - multi-player



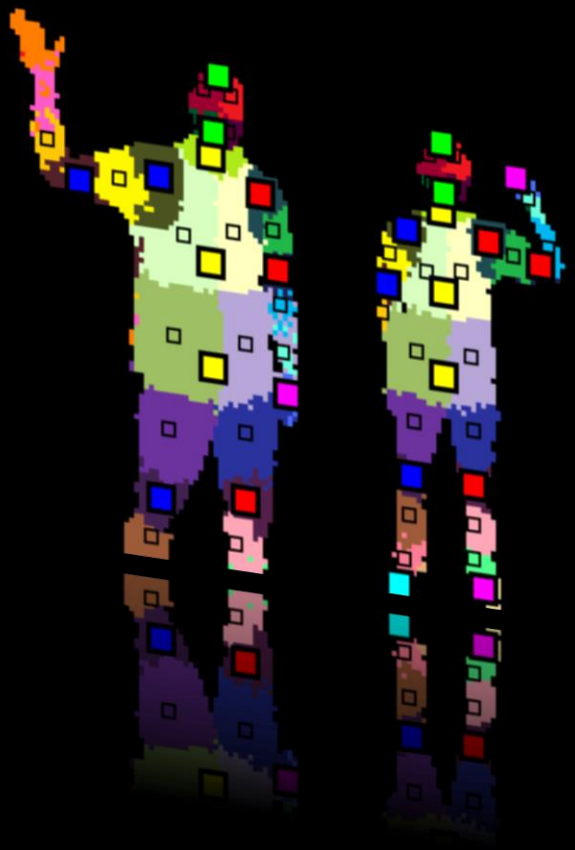
Summary

- Frame-by-frame gives robustness
- Body parts representation for efficiency
- Fast, simple machine learning
- Significant engineering to scale to a massive, varied training data set



Microsoft®
Research

XBOX 360



With thanks to:

Microsoft Research

Andrew Fitzgibbon, Mat Cook, Andrew Blake, Toby Sharp, Ollie Williams, Sebastian Nowozin, Antonio Criminisi, Mihai Budiu, Ross Girshick, Duncan Robertson, John Winn, Shahram Izadi, Pushmeet Kohli

XBOX 360

The whole Kinect team, especially: Alex Kipman, Mark Finocchio, Ryan Geiss, Richard Moore, Robert Craig, Momin Al-Ghosien, Matt Bronder, Craig Peeper