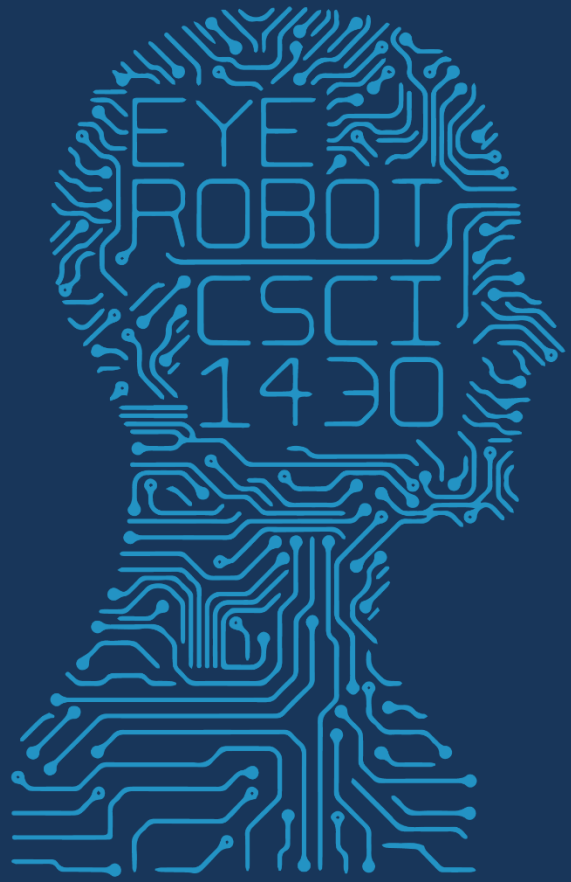
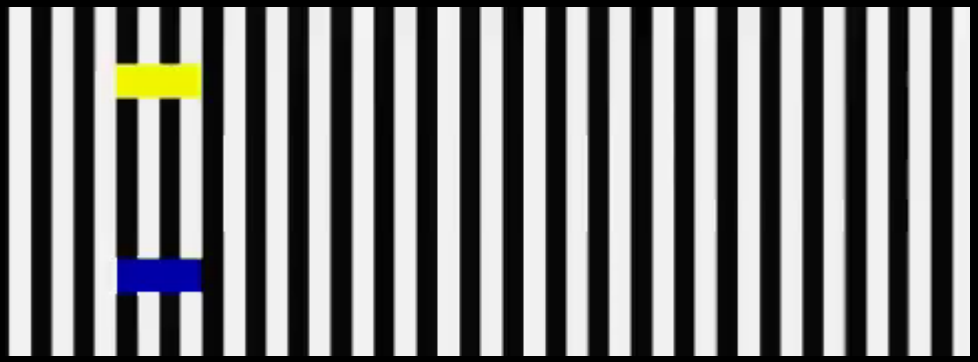
1950

Future Vision

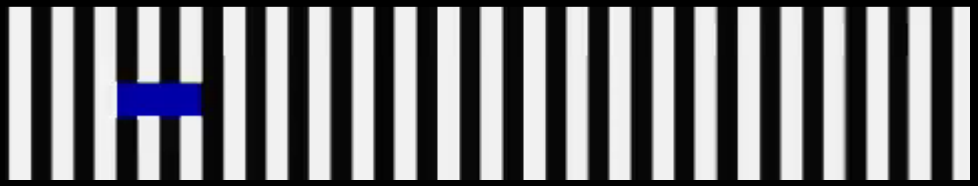


2020

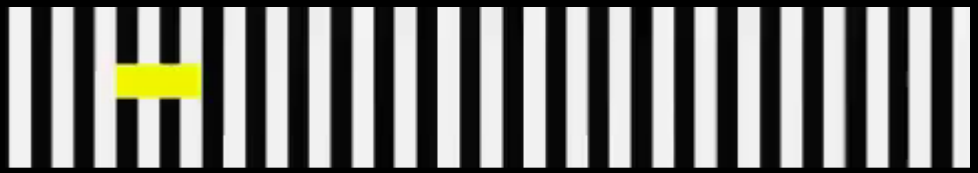
Computer Vision



[Footsteps and Inchworms – Anstis, Perception 2001]



[Footsteps and Inchworms – Anstis, Perception 2001]



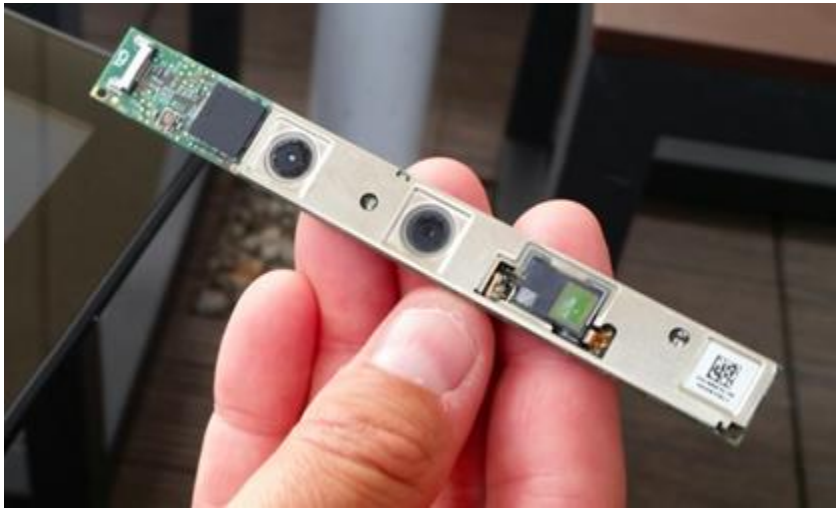
[Footsteps and Inchworms – Anstis, Perception 2001]

https://scratch.mit.edu/projects/188838060/

Demo

How does a depth camera work?

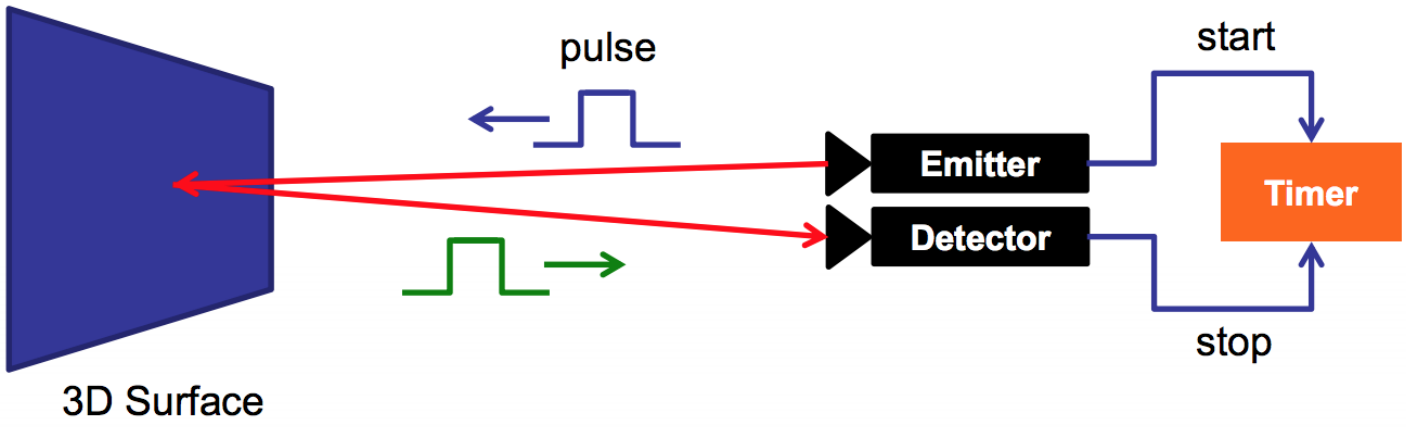
Microsoft Kinect v1



Intel laptop depth camera

Time of Flight (Kinect V2)

- Depth cameras in HoloLens use *time of flight*
  - "SONAR for light"
  - Emit light of a known wavelength, and time how long it takes for it to come back



With either technique…

…I gain depth maps over time.

Optex Depth Camera Based on Canesta Solution

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

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/BodyPartRecognition.pdf

Jamie Shotton et al. (MS Research & Xbox Incubation)

CVPR 2011

Slides by YoungSun Kwon
http://sglab.kaist.ac.kr/~sungeui/IR/Presentation/first/20143050권용선.pdf

2014. 11. 11

Background

Motion Capture (Mocap)

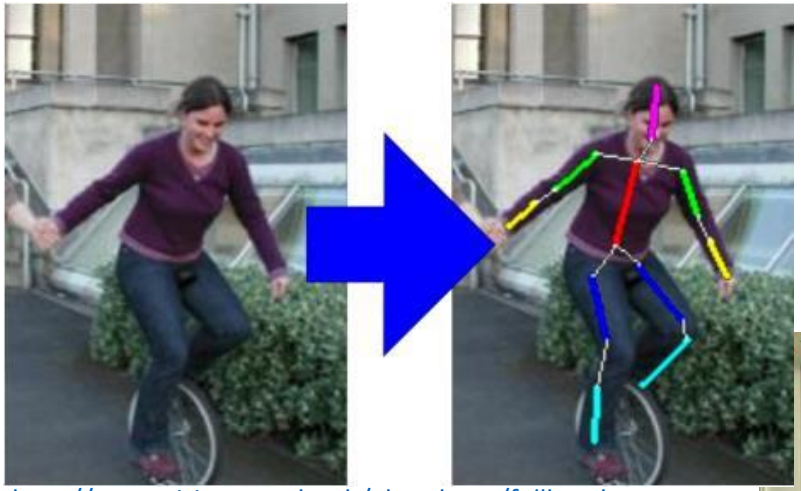
Capture a motion from sensors attached to human body

http://www.neogaf.com/forum/showthread.php?t=824332

Background

Pose Recognition

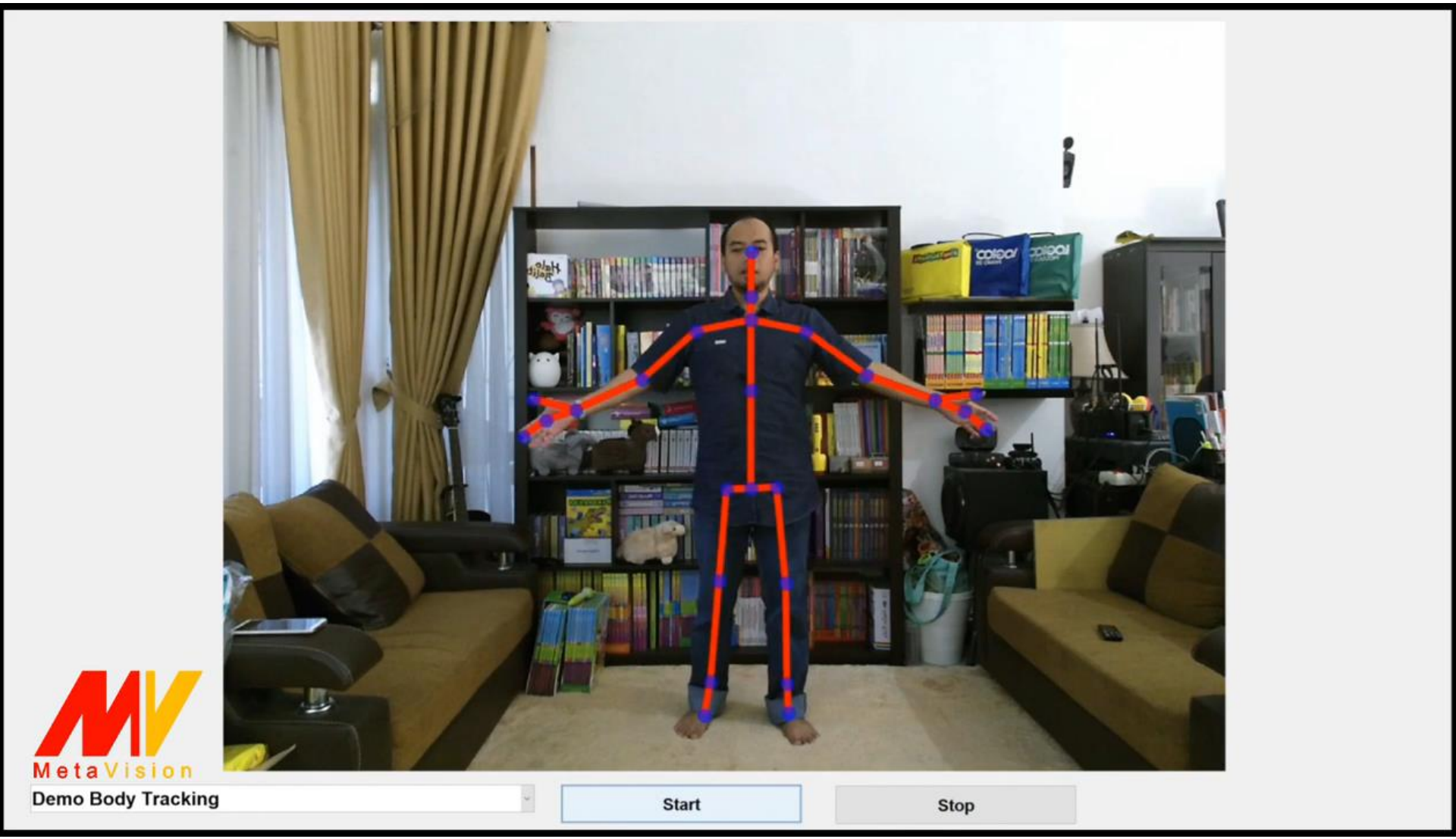
Estimate a pose from images and make a skeletal model



http://www.vision.ee.ethz.ch/~hpedemo/fullhpedemo.png

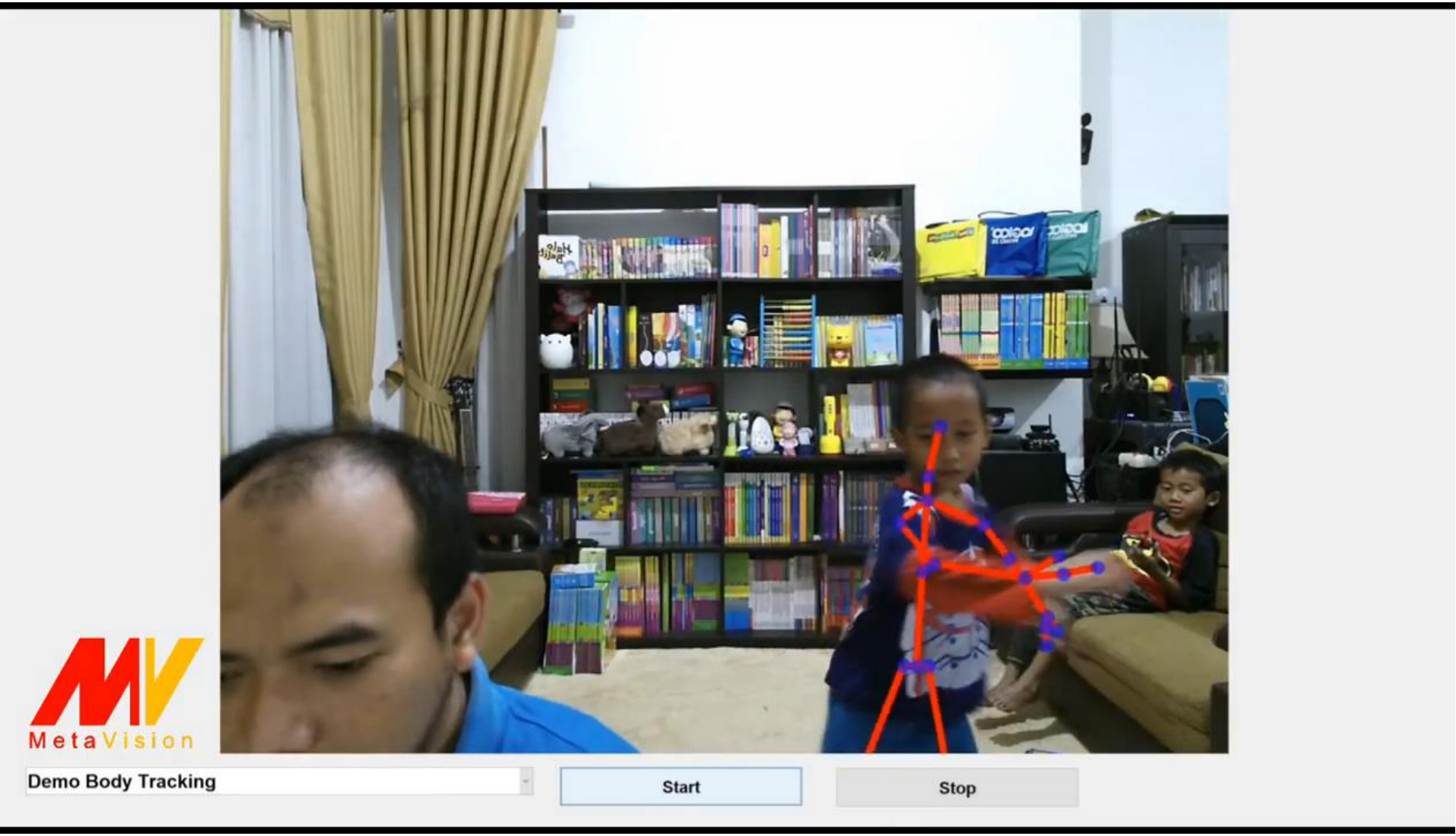
http://www.youtube.com/watch?v=Y-iKWe-U9bY

Kinect v2 body tracking



[MetaVision Studio - https://www.youtube.com/watch?v=CBWxBWQftr4]

Kinect v2 body tracking – multiple people



[MetaVision Studio - https://www.youtube.com/watch?v=mR0a3e7mrKs]

Depth Images

Each pixel has distance information, instead of RGB

Simplifies scene! Less variation

RGB Image

RGB Camera

Depth Image

Depth Camera

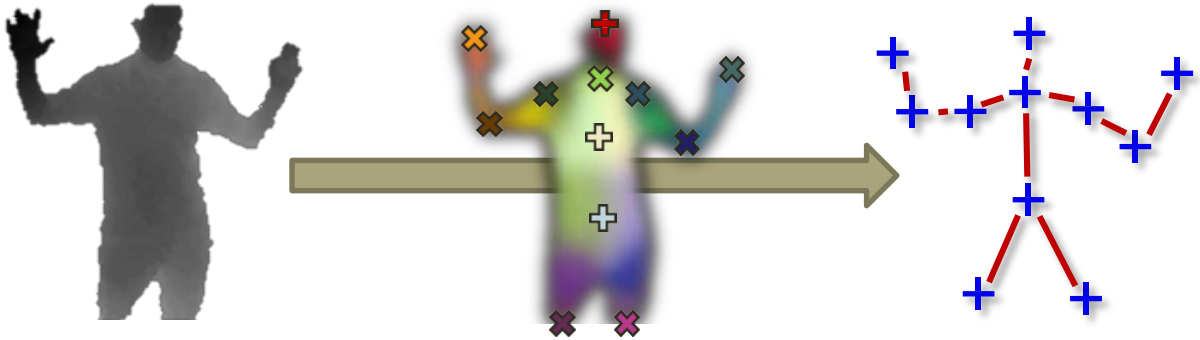
http://userpage.fu-berlin.de/~latotzky/wheelchair/wp-content/uploads/kinect1_cropped.png

Main Contribution

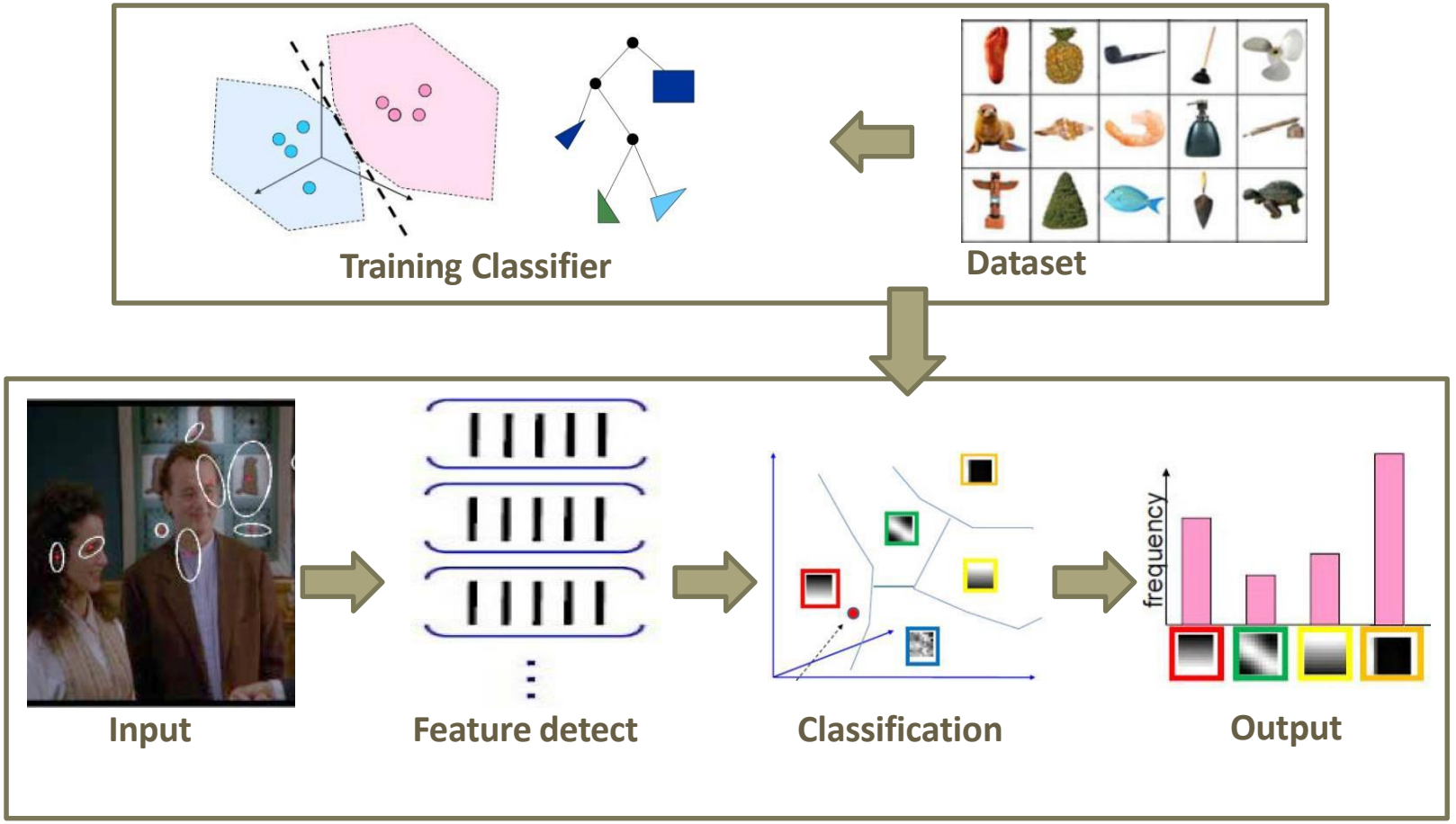
Pose recognition as a classification problem

One application of our image retrieval techniques applied to different data.

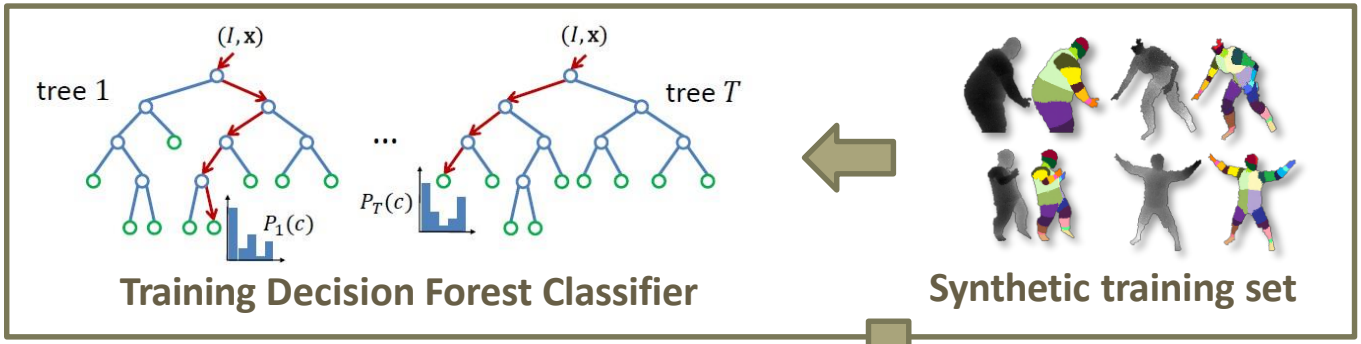
No constraint, More General

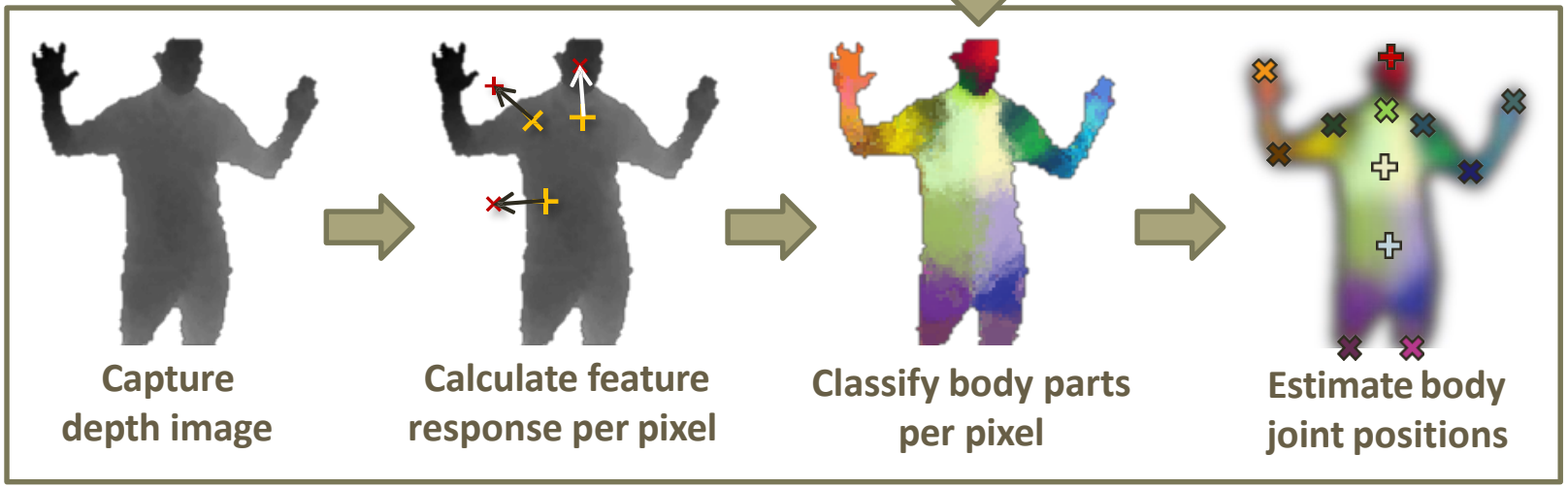


Overview


Training Classifier
Dataset
Input
Feature detect
Classification
frequency
Output

Overview



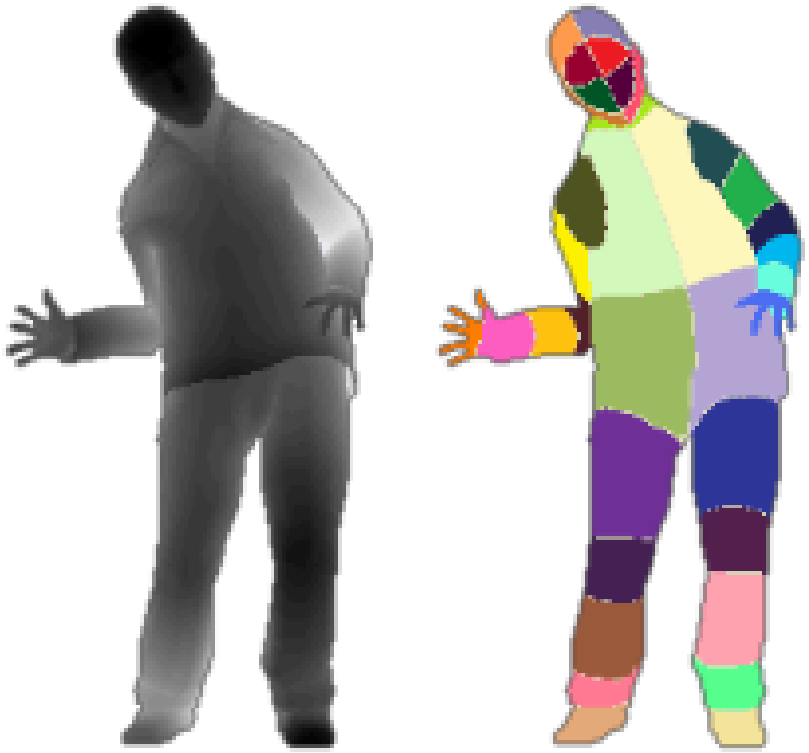


Kinect Slides CVPR2011.pptx

Body Part Representation

31 body parts (classes)

- LU/RU/LW/RW head
- Neck
- L/R shoulder
- LU/RU/LW/RW arm
- L/R elbow
- L/R wrist
- L/R hand
- LU/RU/LW/RW torso
- LU/RU/LW/RW leg
- L/R knee
- L/R ankle
- L/R foot



Synthetic dataset

To account for variations in real world

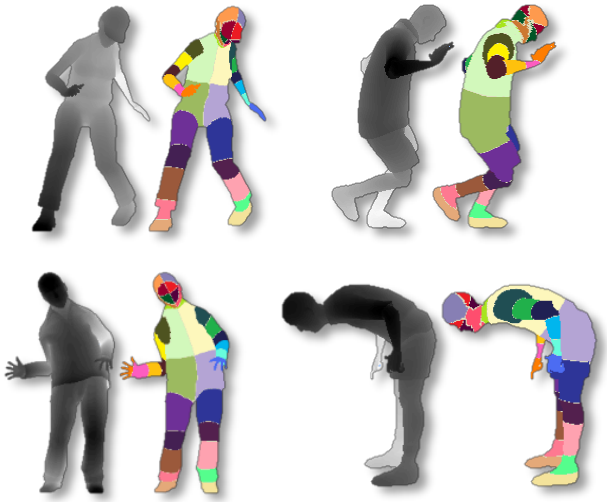
- Rotation & Translation, Hair, Clothing, Height, Camera Pose, etc…

Large scale and variety

Record motion captures
500K frames and
extract **100K poses** among these

Create **several models**
with variations

Render (depth, body parts) pairs



Depth Image Feature Comparison

Calculate feature response for each pixel

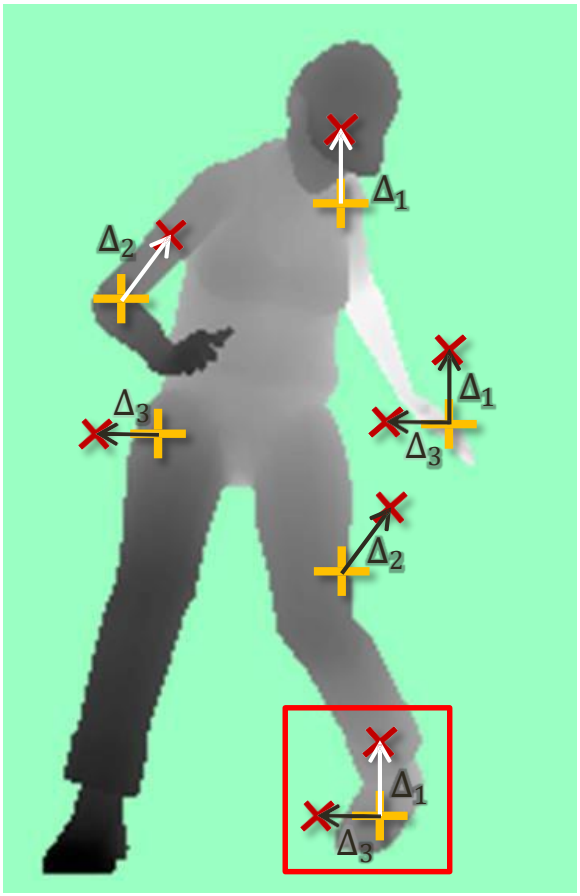
Feature Response Function

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

image, pixel, depth, offset depth

- For example (in red to the right)
  - $\Delta_1 = (0, 1)$ $\Delta_3 = (-1, 0)$
  - $f(I, \boldsymbol{x} | \Delta_1)$ has small value
  - $f(I, \boldsymbol{x} | \Delta_3)$ has large value
- Can be trained in parallel on GPUs

Input depth image



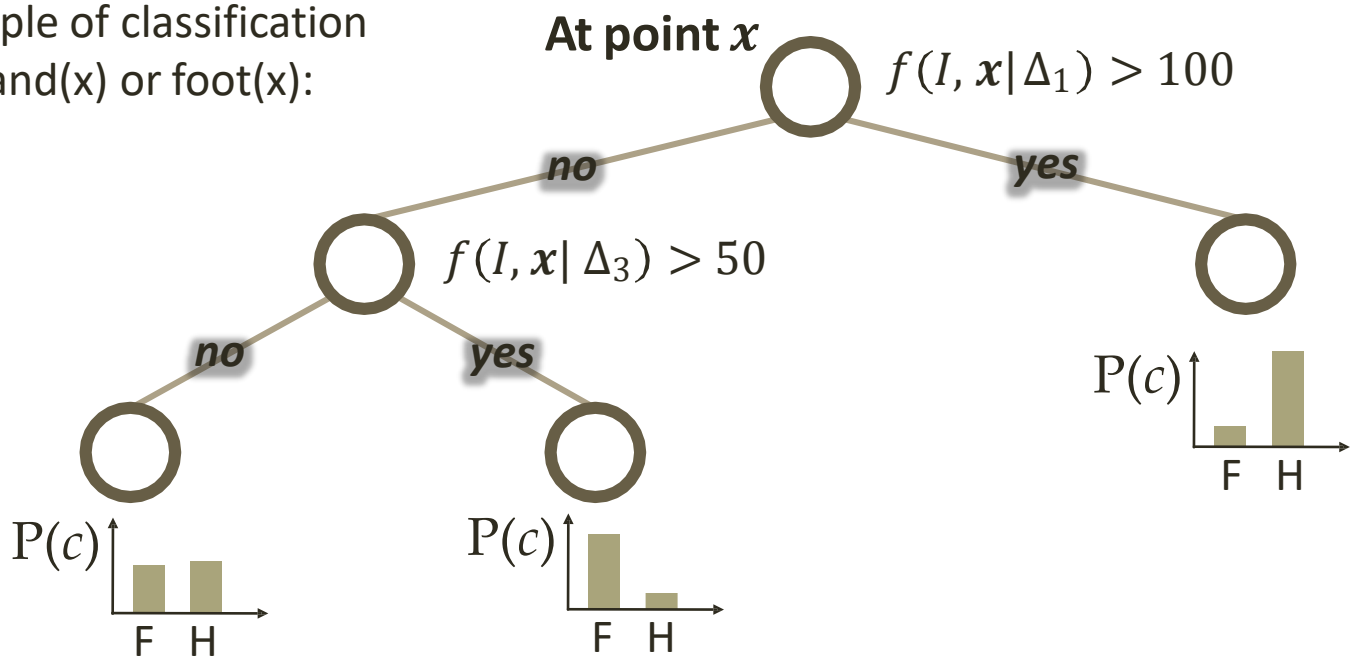
[3] V. Lepetit et al., Randomized trees for real-time keypoint recognition, CVPR, 2005

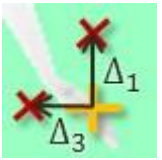
Decision tree classifier

Remember Viola-Jones face detector?

Many weak classifiers in a decision tree.
Train offset and threshold for each weak classifier

Example of classification for hand(x) or foot(x):



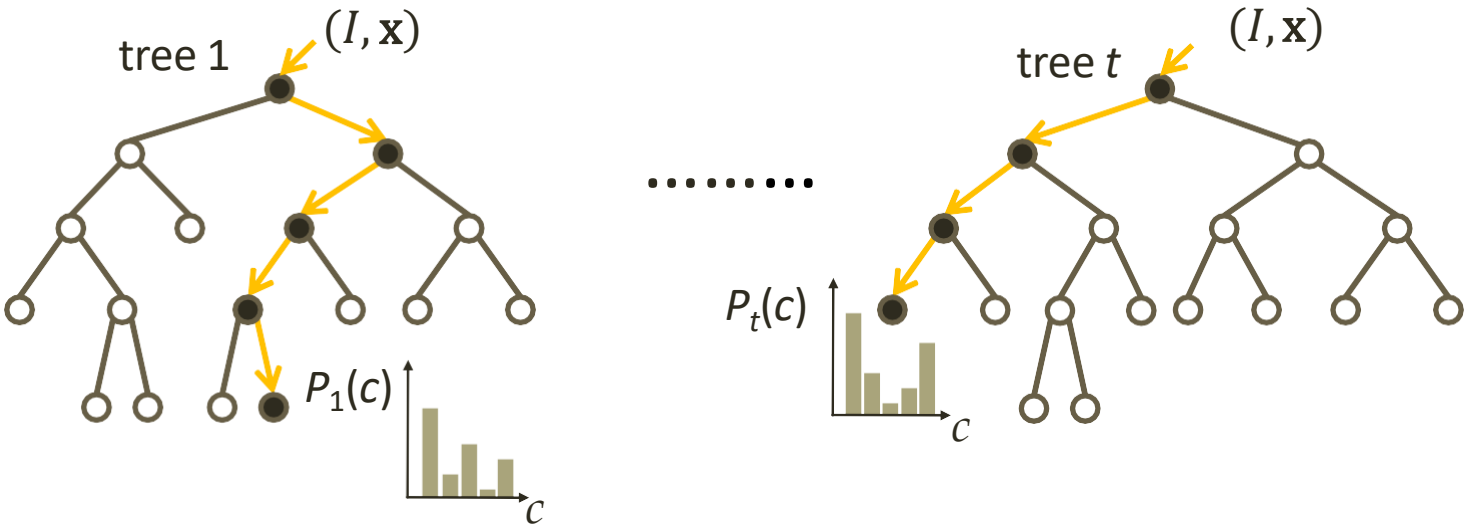


4 T. Amit et al., Shape quantization and recognition with randomized trees, Neural Computation, 1997
5 L. Breiman, Random forests, Mach. Learning, 2001
6 F. Moosmann et al., Fast discriminative visual codebooks using randomized clustering forests, NIPS, 2006

Decision Forest Classifier

In training step, generate *many trees T* to build a decision *forest*

In testing step, check all trees and compute average probability

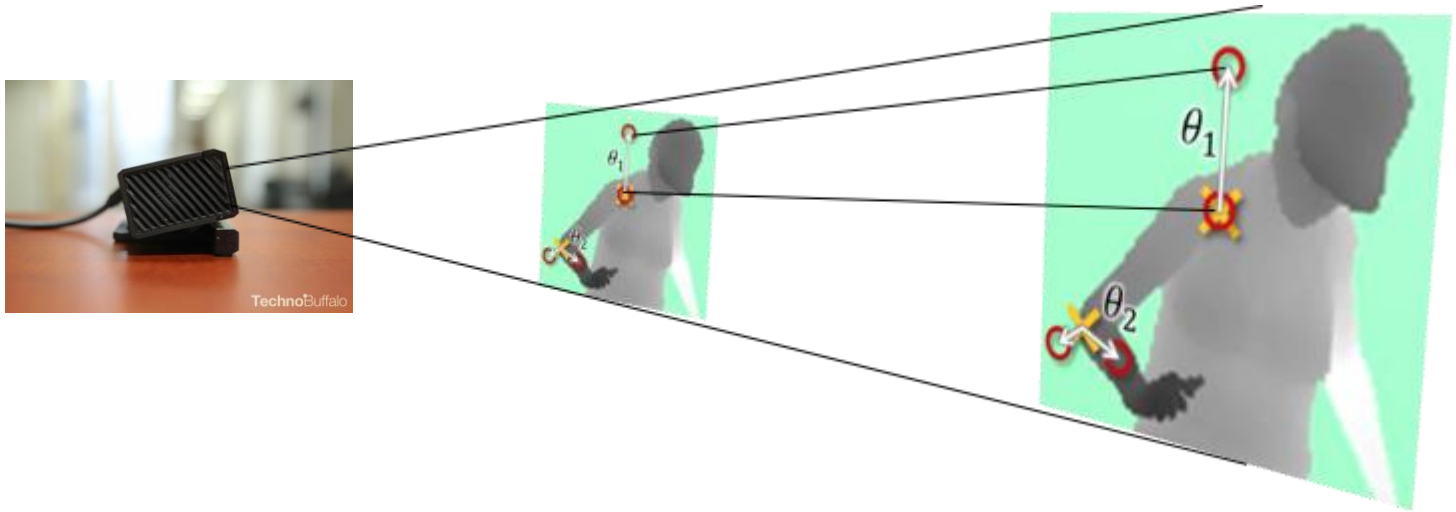


$$P(c|I, \mathbf{x}) = \frac{1}{T}\sum_t^T P_t(x, I|c)$$

Two details – depth invariance and two offsets

Offset 1 Offset 2

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



Yisheng Zhou

Joint Position Proposal

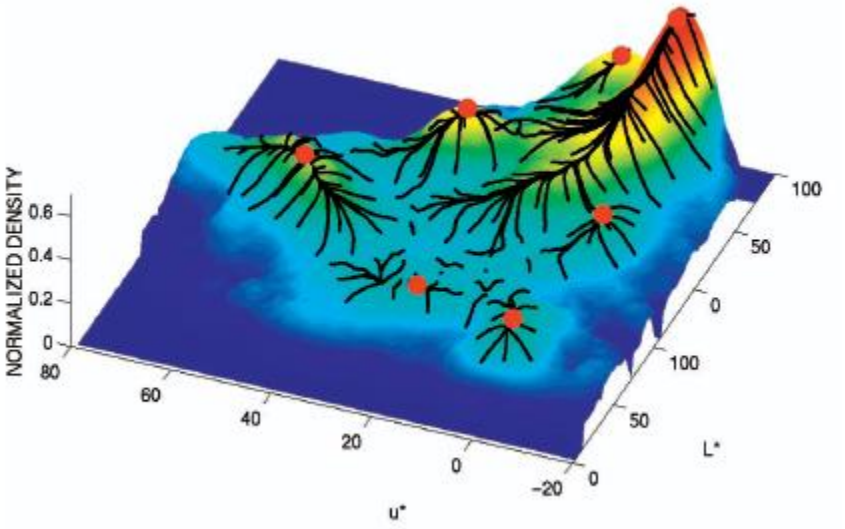
- **Find mode using mean shift algorithm**
  - With weighted Gaussian kernel
  - Using class probabilities for each pixel, find **representative positions** of classes

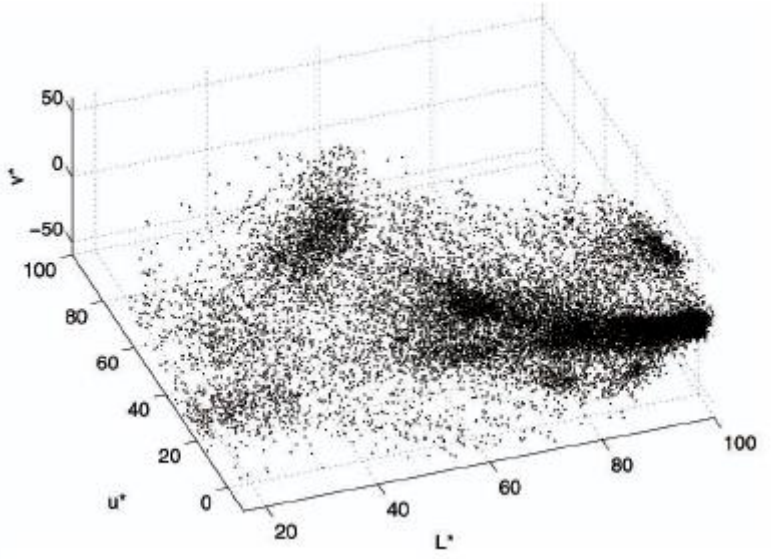
Estimate body joint positions

[7] S. Belongie et al., Mean shift: A robust approach toward feature space analysis, PAMI, 2002

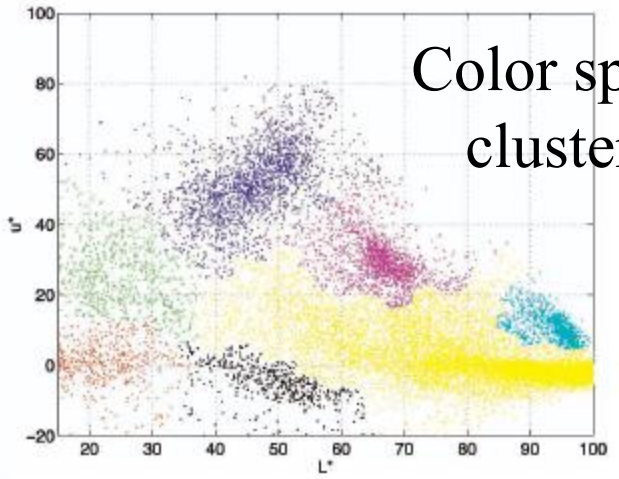
Mean shift algorithm

Try to find *modes* of a non-parametric density.



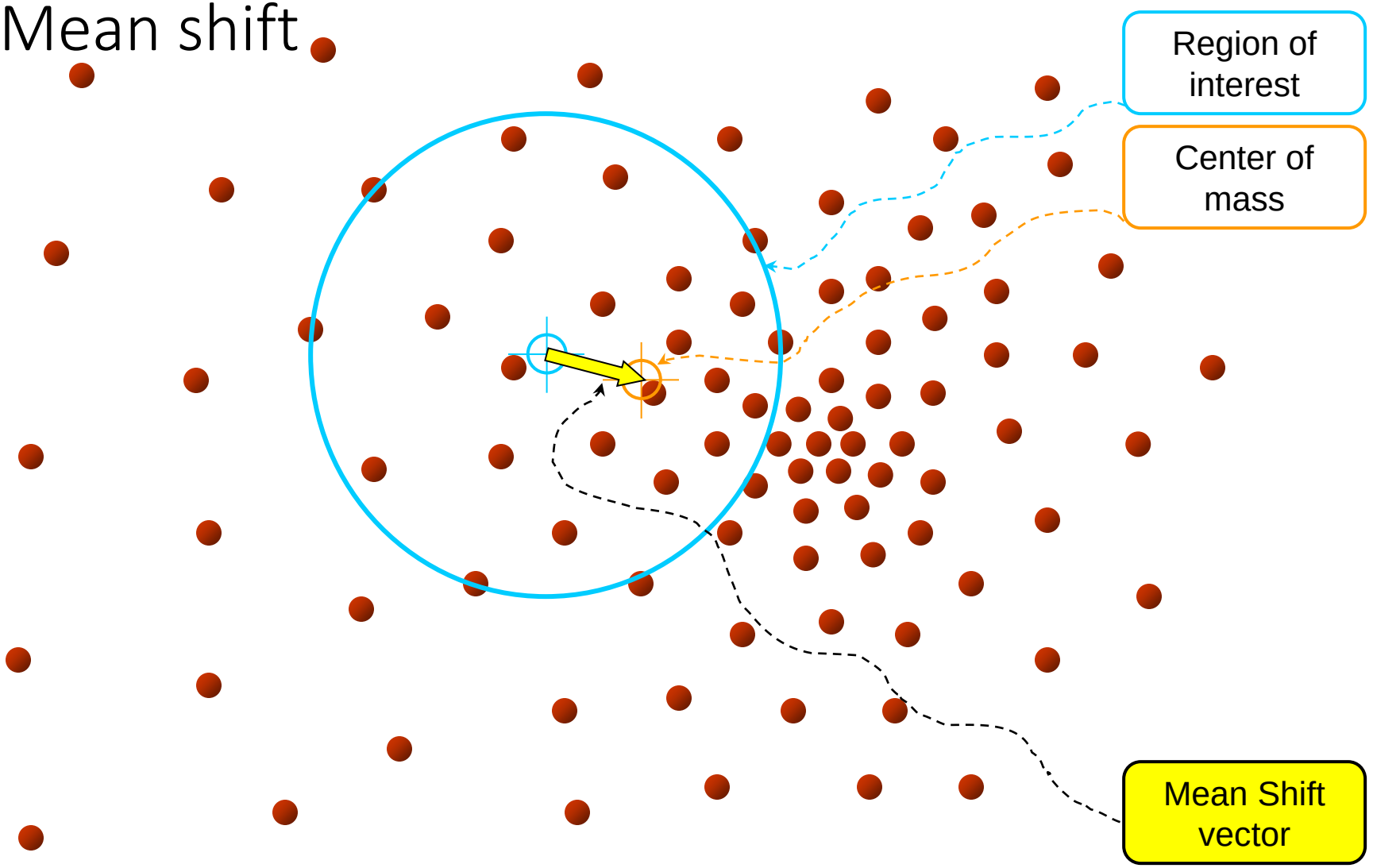


Color space



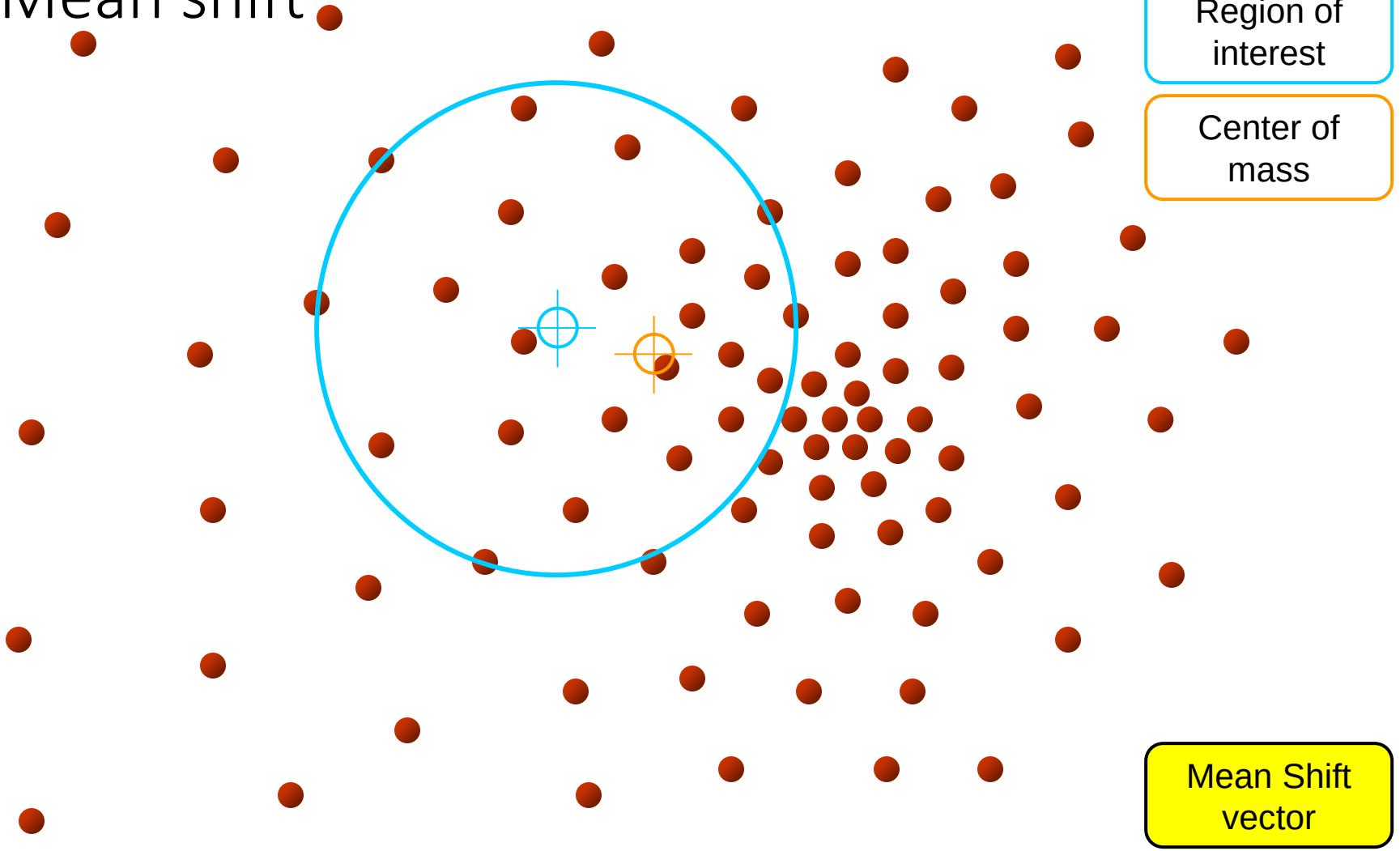
Color space clusters

Mean shift



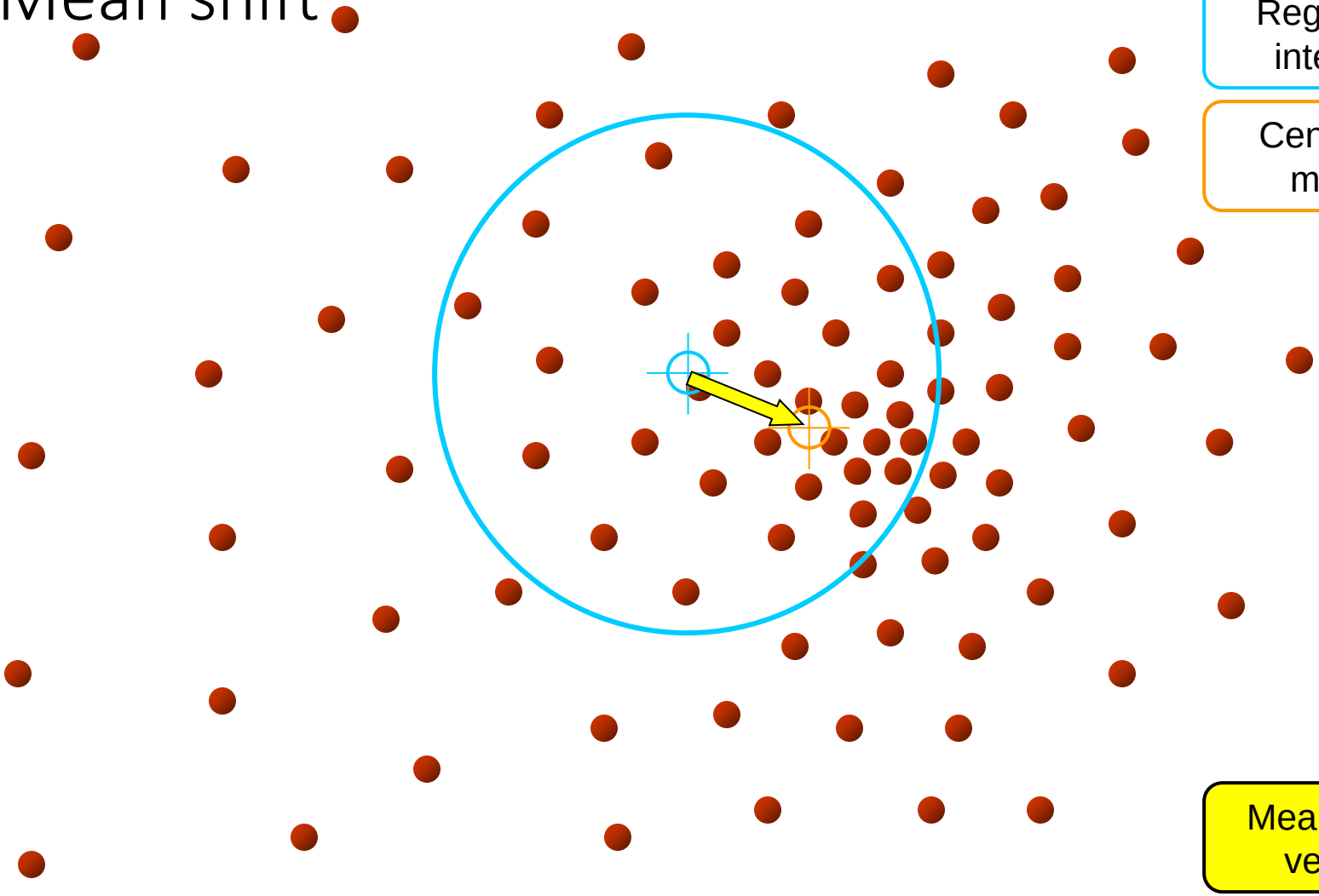
Slide by Y. Ukrainitz & B. Sarel

Mean shift



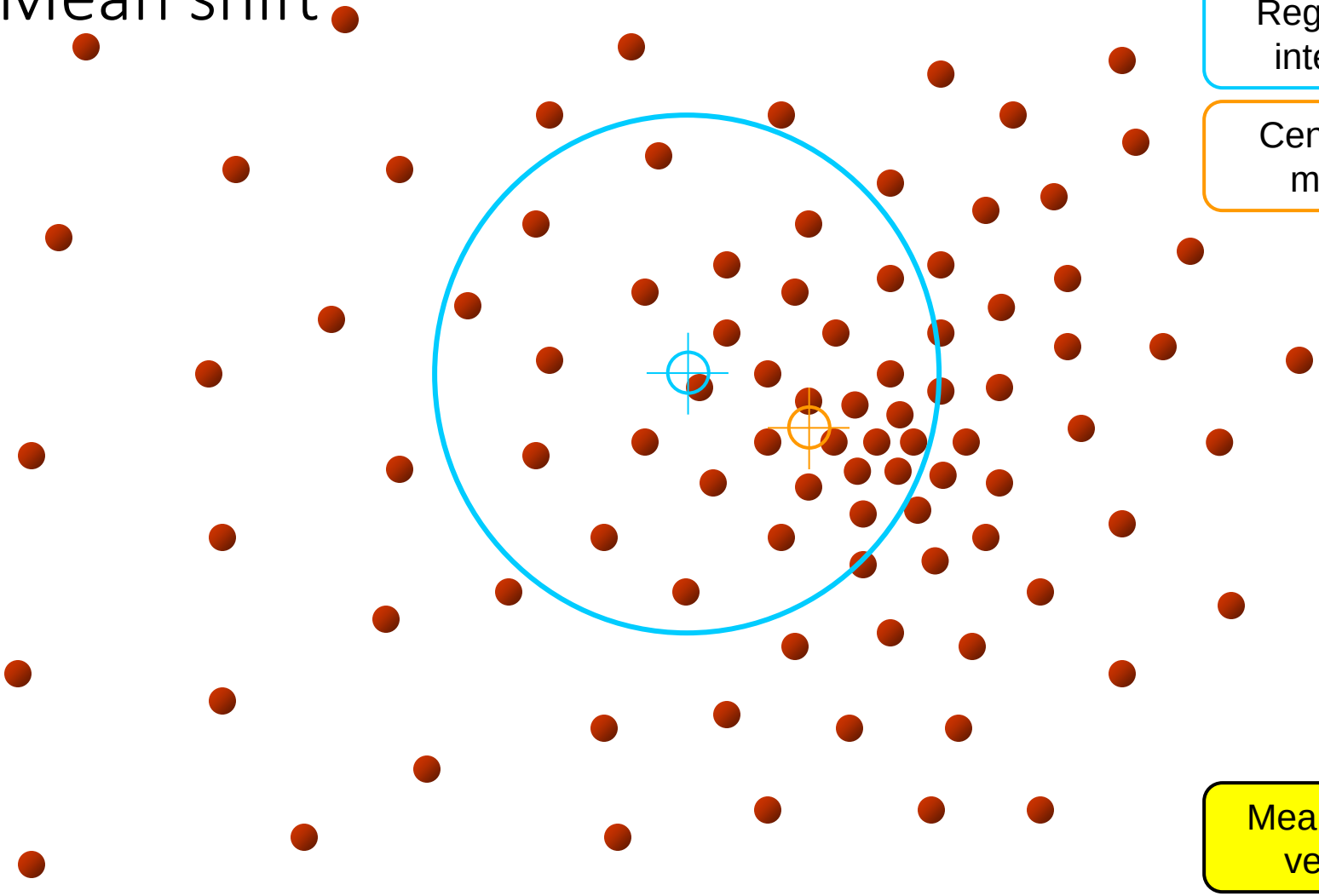
Slide by Y. Ukrainitz & B. Sarel

Mean shift



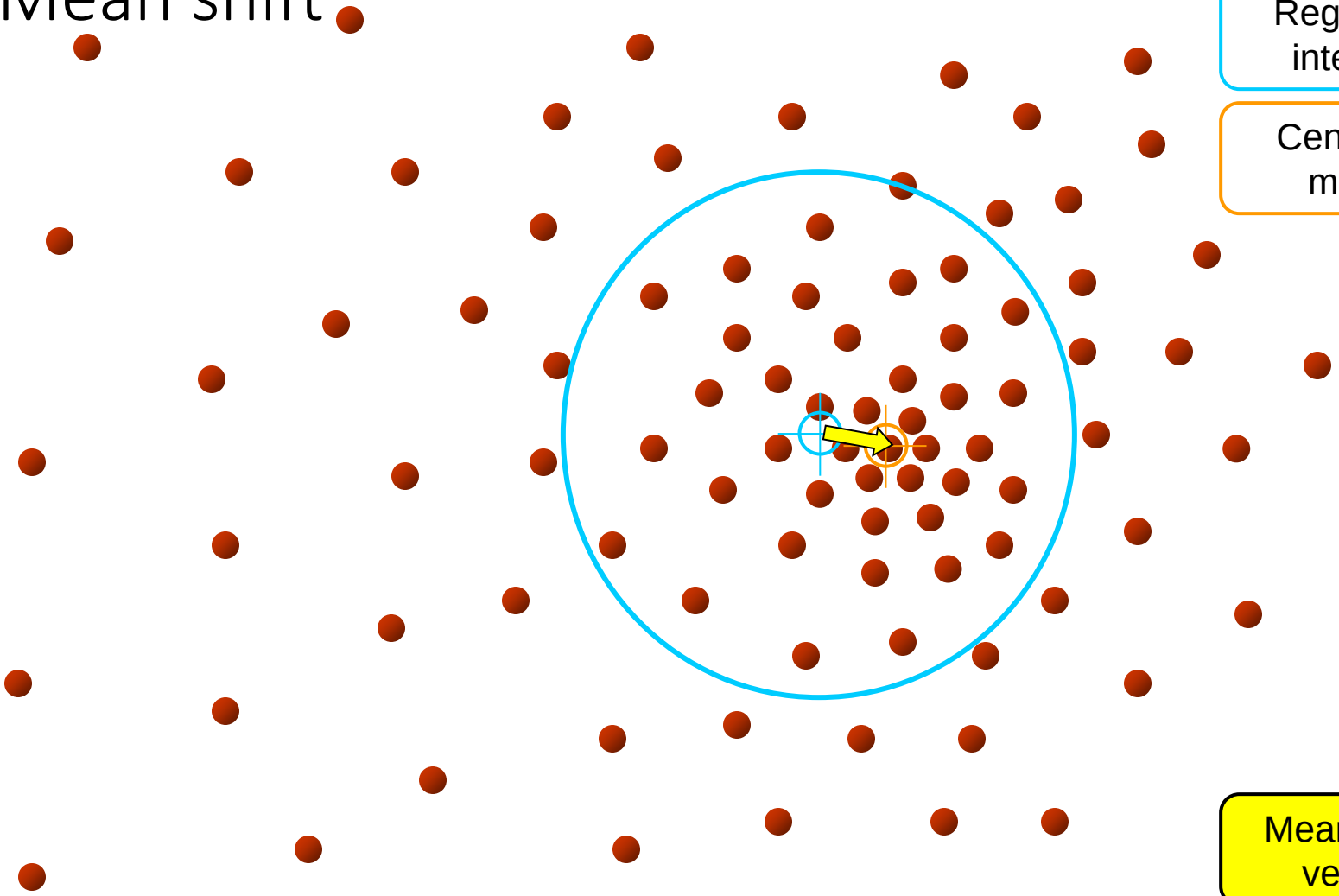
Slide by Y. Ukrainitz & B. Sarel

Mean shift



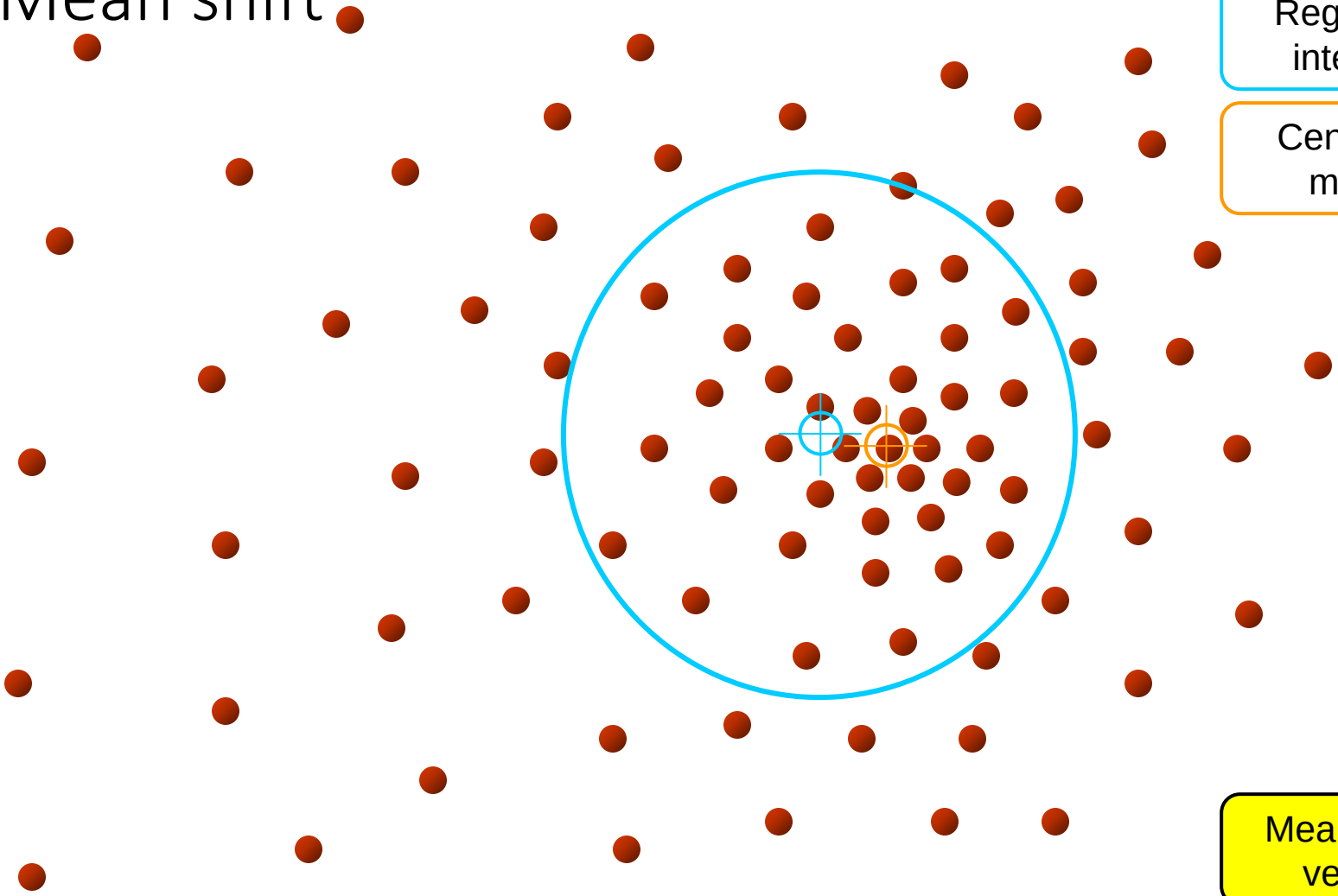
Slide by Y. Ukrainitz & B. Sarel

Mean shift



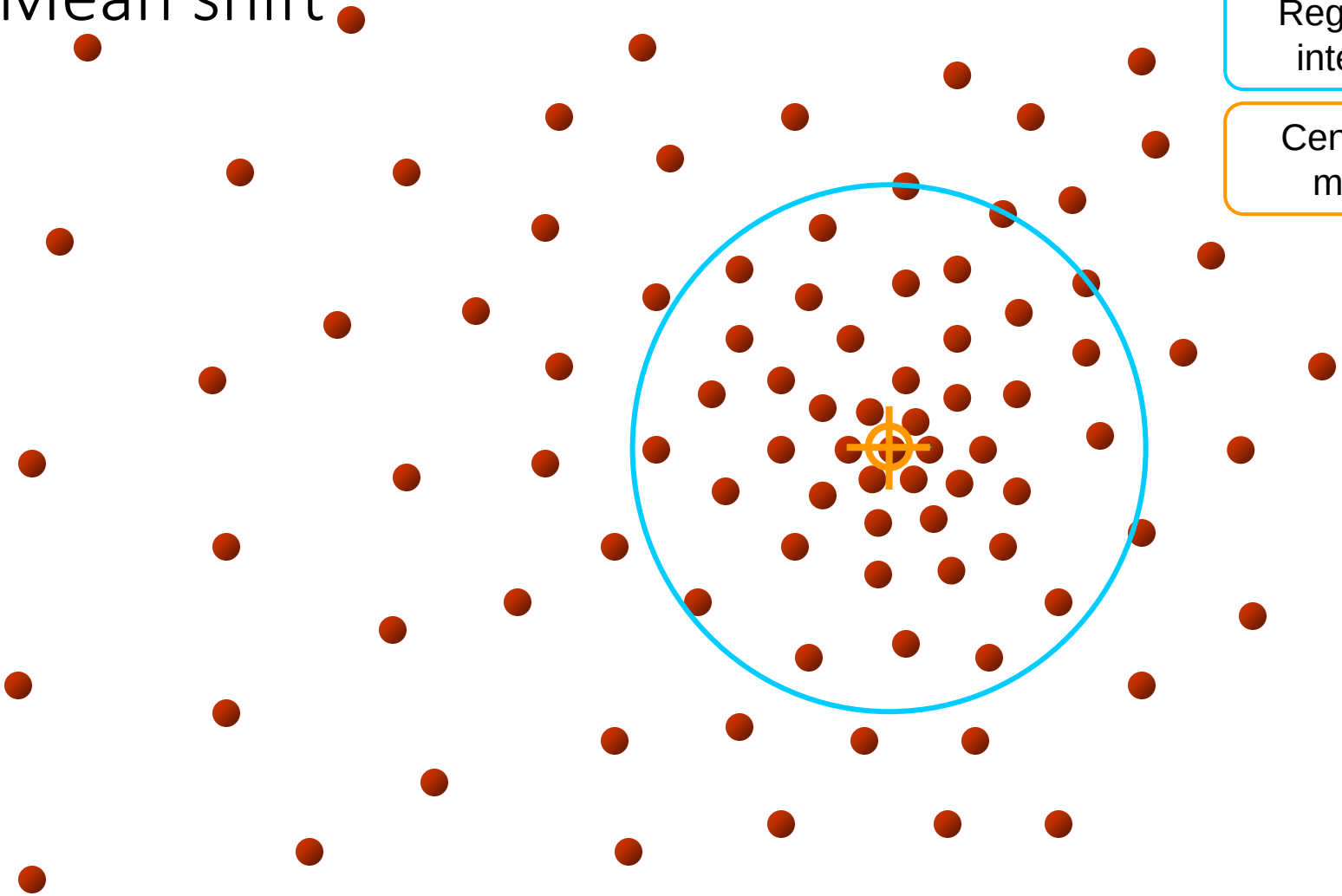
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Kernel density estimation

Kernel density estimation function

$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

n = number of points assessed
h = 'bandwidth', or normalization for size of region

Gaussian kernel

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Mean shift clustering

The mean shift algorithm seeks *modes* of the given set of points

1. Choose kernel and bandwidth
2. For each cluster initial point:
    a) Center a window on that point
    b) Compute the mean of the data in the search window
    c) Center the search window at the new mean location
    d) Repeat (b,c) until convergence
3. Assign points that lead to nearby modes to the same cluster

Joint Position Proposal

Find mode using mean shift algorithm

- With weighted Gaussian kernel
- Using class probabilities for each pixel, find **representative positions** of classes

3D position of class / pixel weight / 3D position of i pixel

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

pixel index i / bandwidth

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

class probability / depth at i pixel

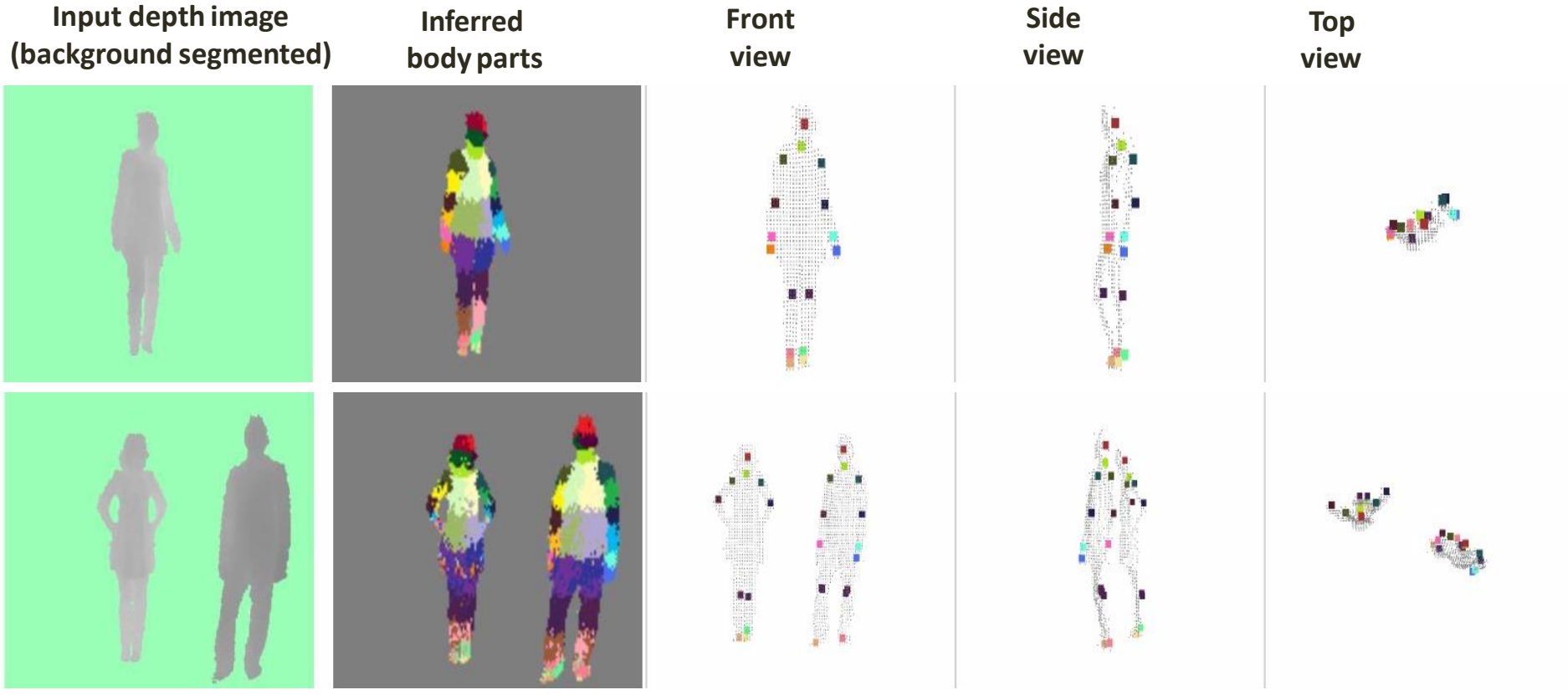
Why this depth term?
"world surface area"
-> each pixel is worth a different amount

Estimate body joint positions

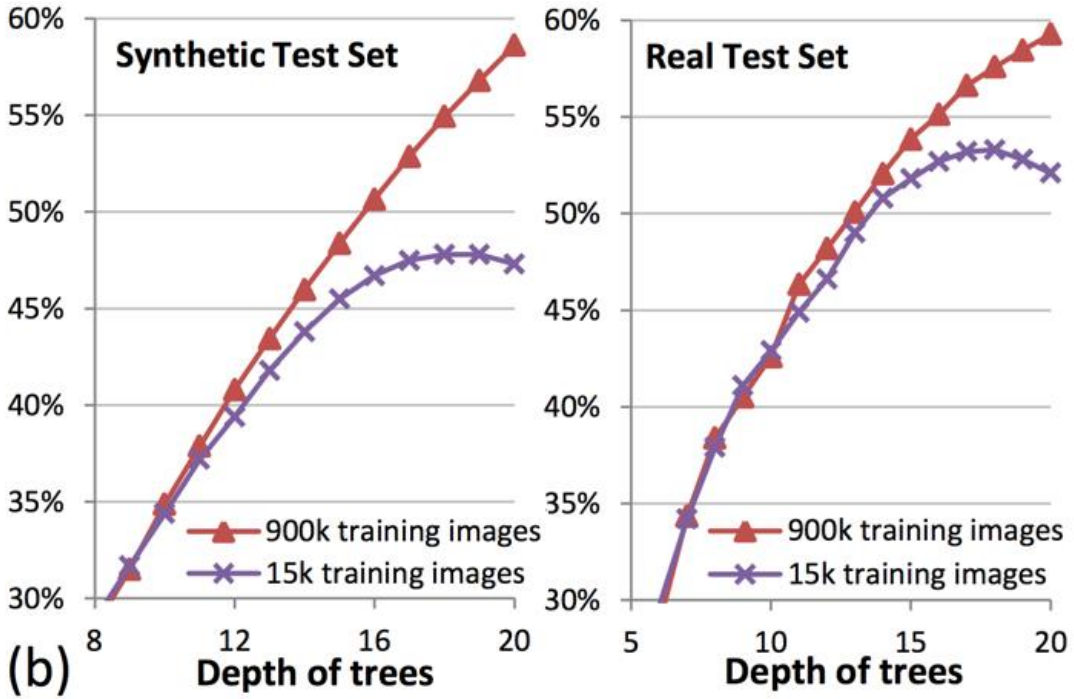
[7] S. Belongie et al., Mean shift: A robust approach toward feature space analysis, PAMI, 2002

Results

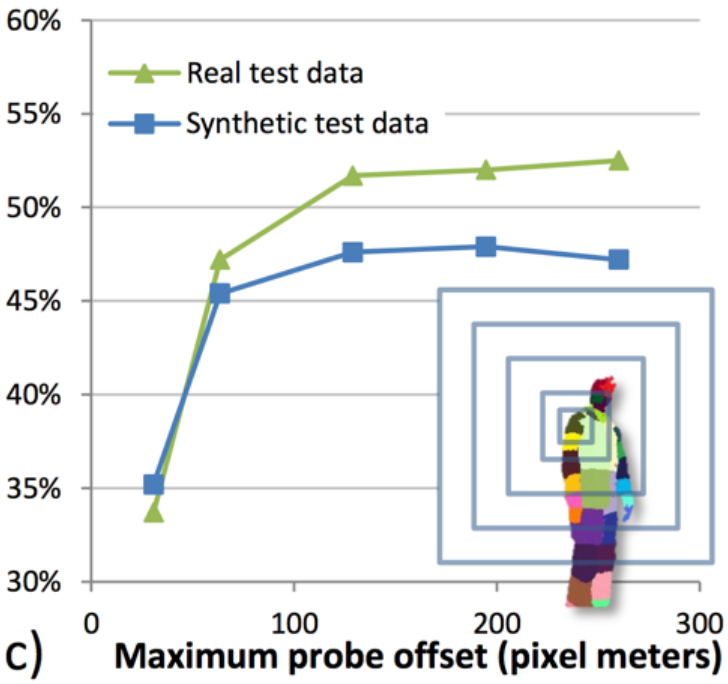
- **Fast Joint Proposals**
  - **Max. 200 FPS on Xbox 360 GPU, 50 FPS on 8 core CPU**
    - Previous work was 4-16 FPS



Depth of trees

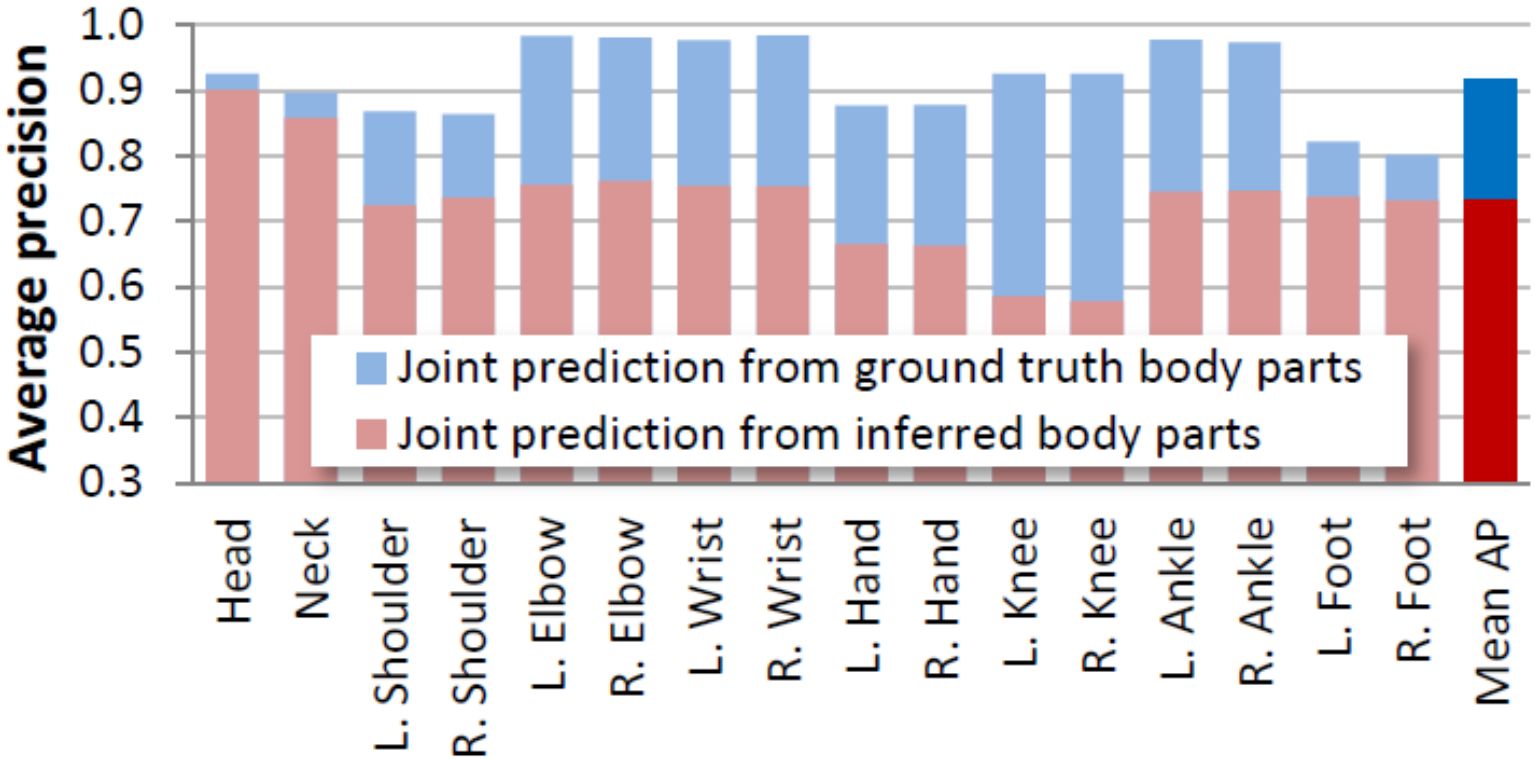


Offset Size


60%
55%
50%
45%
40%
35%
30%
Real test data
Synthetic test data
0
100
200
300
C)
Maximum probe offset (pixel meters)

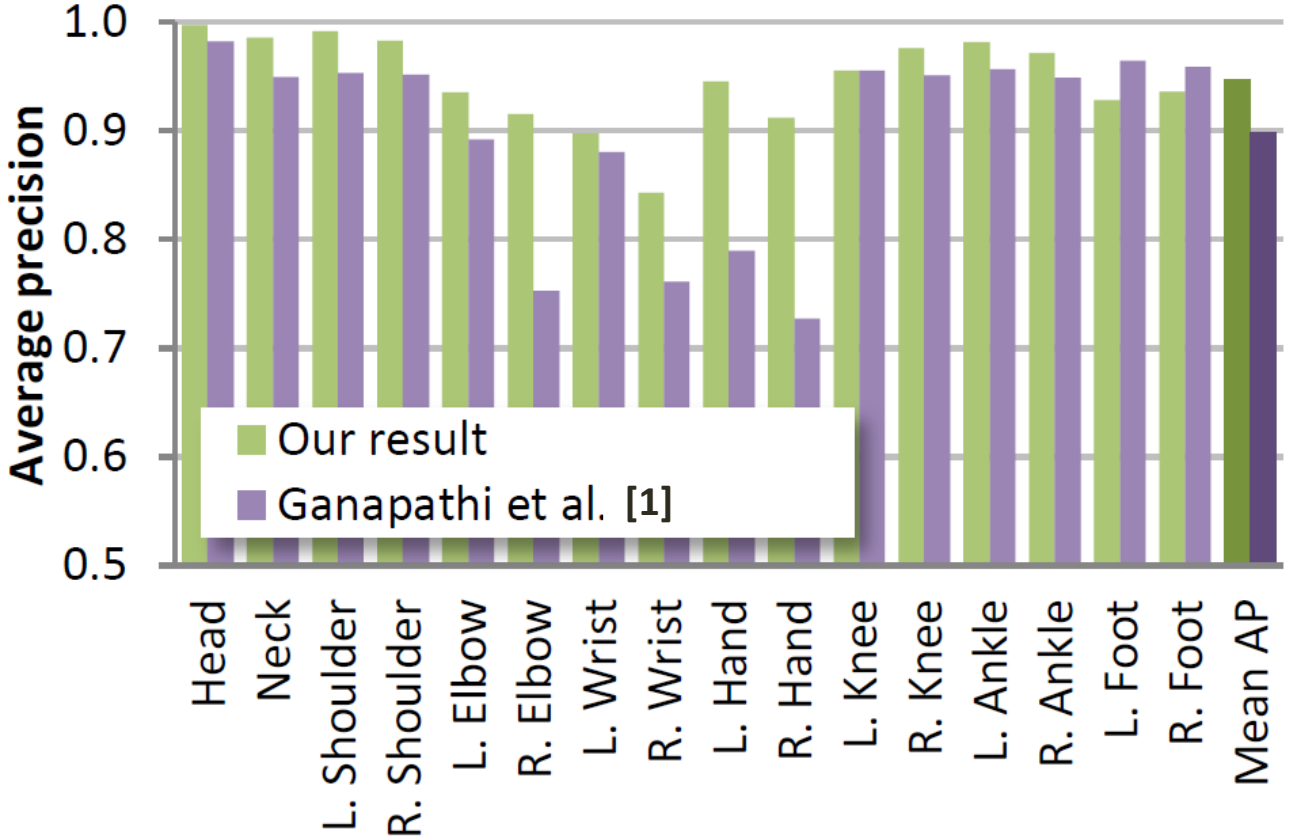
Results

- **Body Parts Classification Accuracy on synthetic test set**
  - **GT body parts (0.914 mAP) vs Our Algorithm (0.731 mAP)**



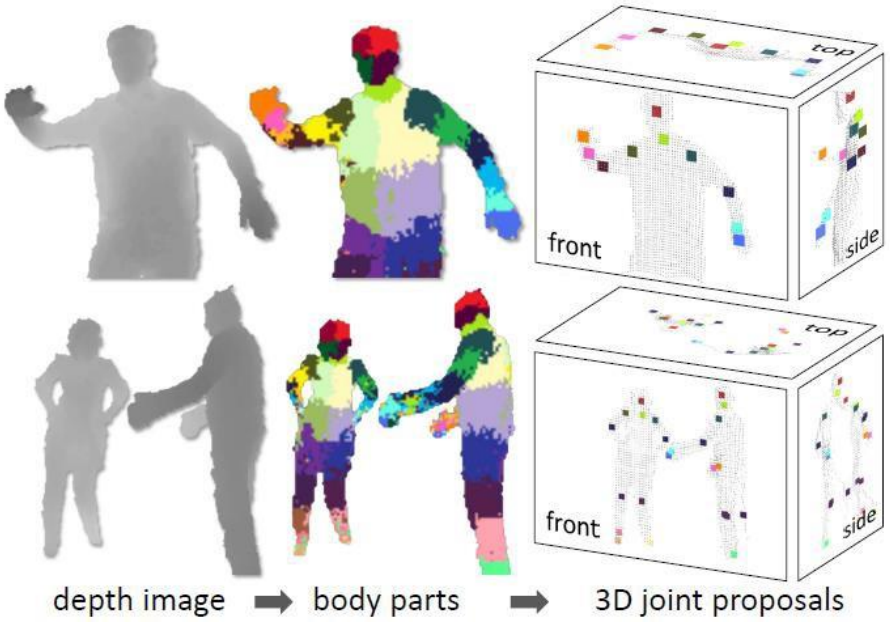
Results

- **Joint Prediction Accuracy**
  - **How well body joint position is predicted**

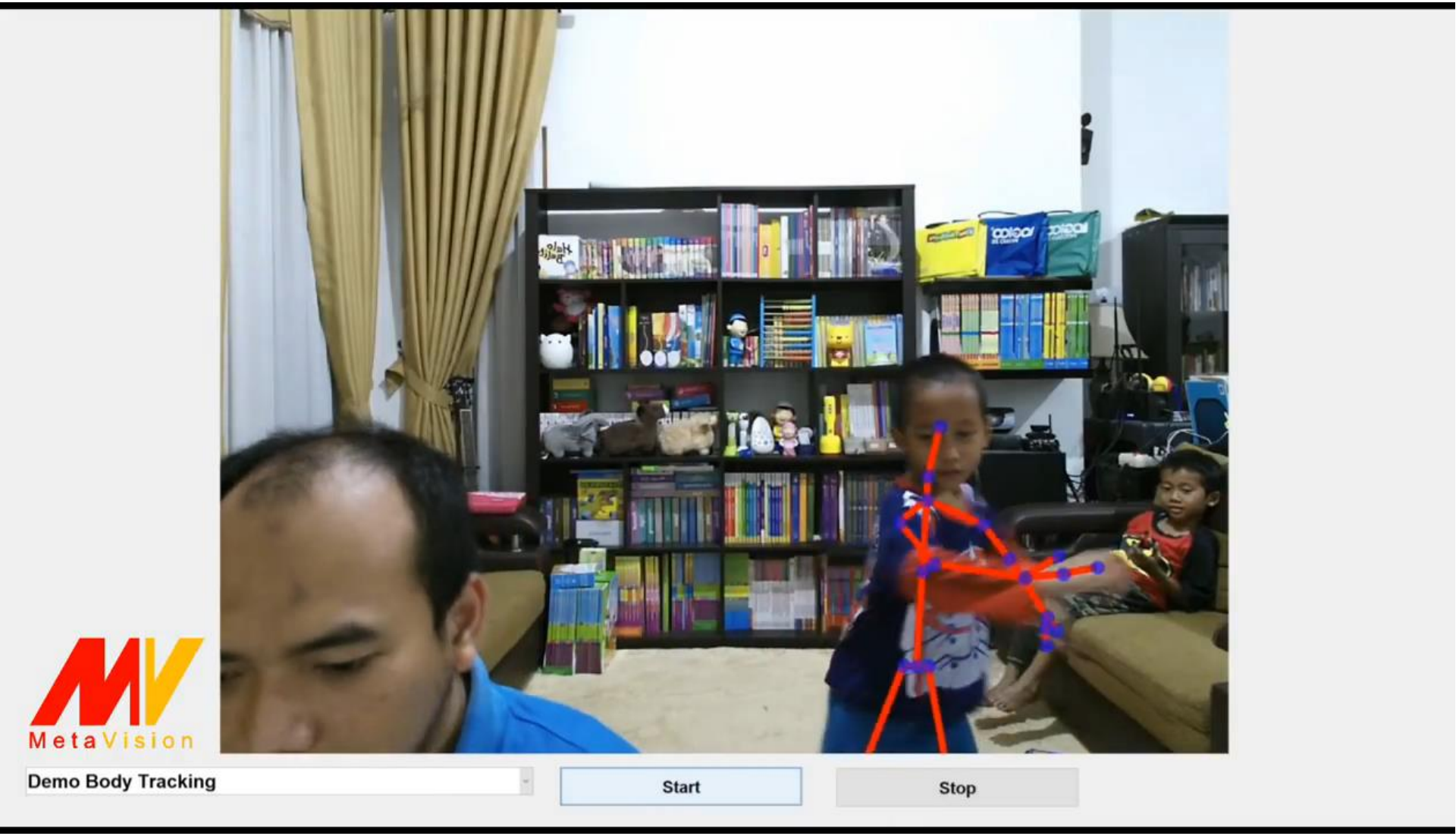


Summary

- **Body parts representation for efficiency**
- **Fast, simple machine learning – Decision Forest**
- **No constraint, high generality**
- **Significant engineering to scale to a massive, varied training dataset**



Kinect v2 body tracking – multiple people



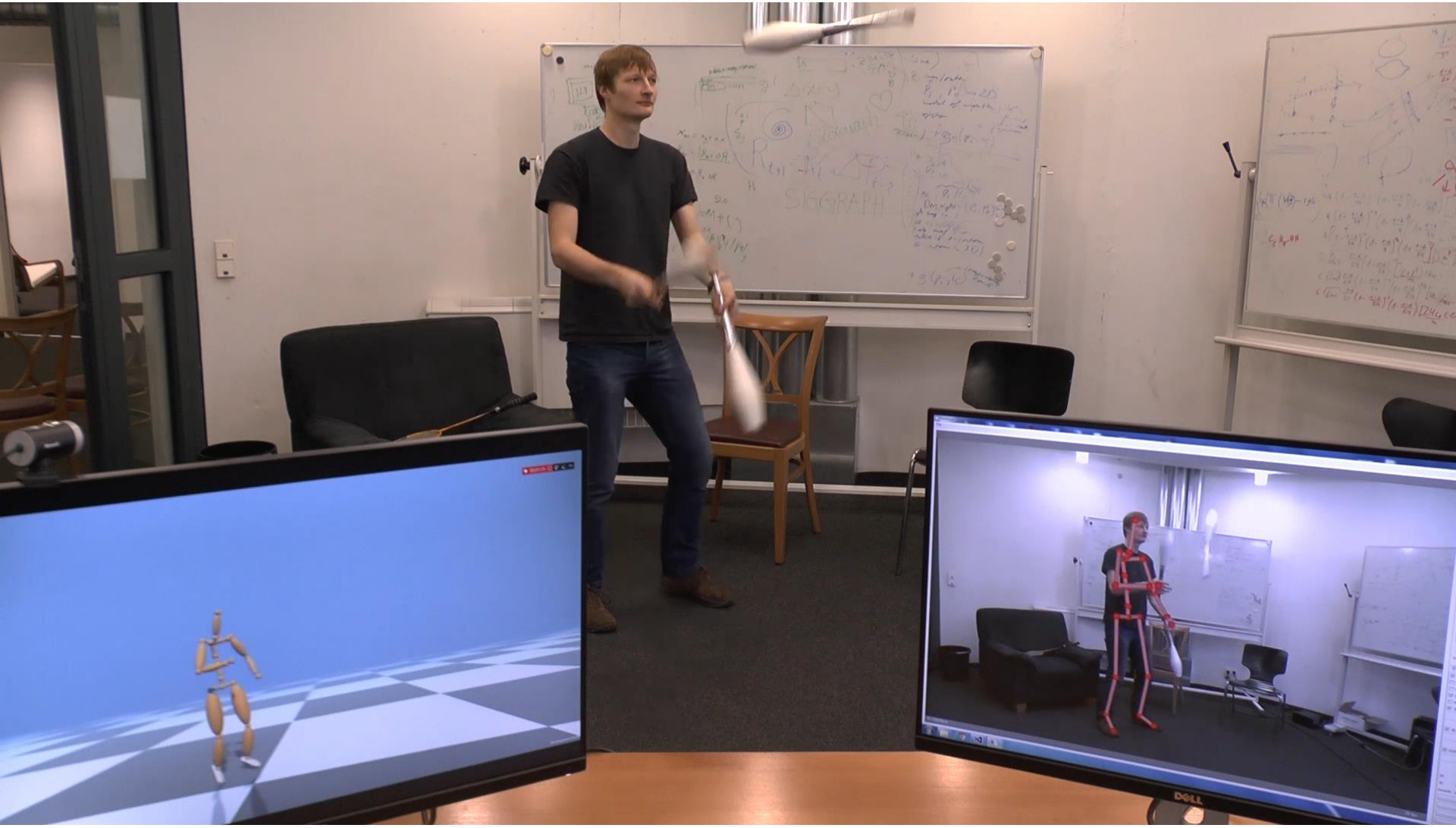
[MetaVision Studio - https://www.youtube.com/watch?v=mR0a3e7mrKs]

VNect – Mehta et al.

Depth information is rich…

…but do we always need it?

Can we learn to predict joint locations from RGB data?


SIGGRAPH
DELL

Pipeline

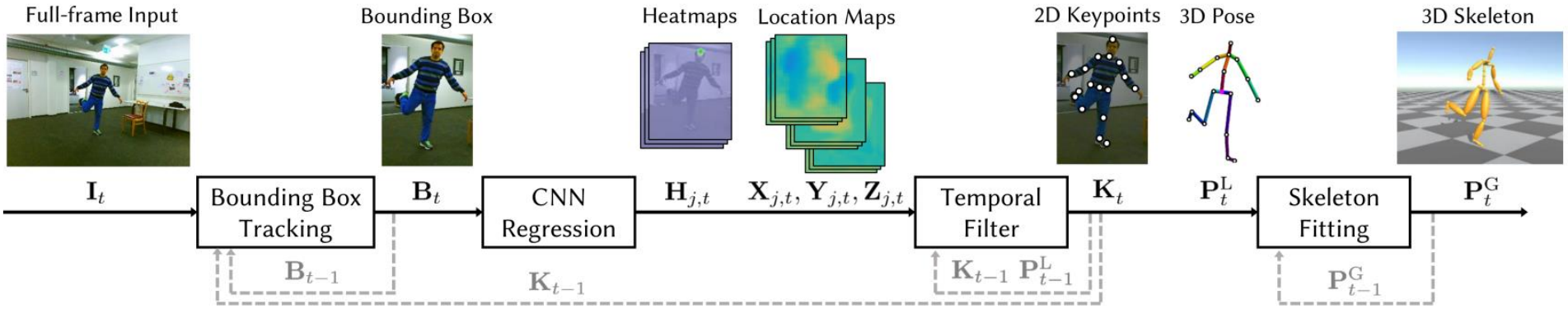


Fig. 2. Overview. Given a full-size image $\mathbf{I}_t$ at frame t, the person-centered crop $\mathbf{B}_t$ is efficiently extracted by bounding box tracking, using the previous frame's keypoints $\mathbf{K}_{t-1}$. From the crop, the CNN jointly predicts 2D heatmaps $\mathbf{H}_{j,t}$ and our novel 3D *location-maps* $\mathbf{X}_{j,t}$, $\mathbf{Y}_{j,t}$ and $\mathbf{Z}_{j,t}$ for all joints j. The 2D keypoints $\mathbf{K}_t$ are retrieved from $\mathbf{H}_{j,t}$ and, after filtering, are used to read off 3D pose $\mathbf{P}_t^{\mathrm{L}}$ from $\mathbf{X}_{j,t}$, $\mathbf{Y}_{j,t}$ and $\mathbf{Z}_{j,t}$. These per-frame estimates are combined to stable global pose $\mathbf{P}_t^{\mathrm{G}}$ by skeleton fitting. Information from frame $t-1$ is marked in gray-dashed.

Joint position encoding

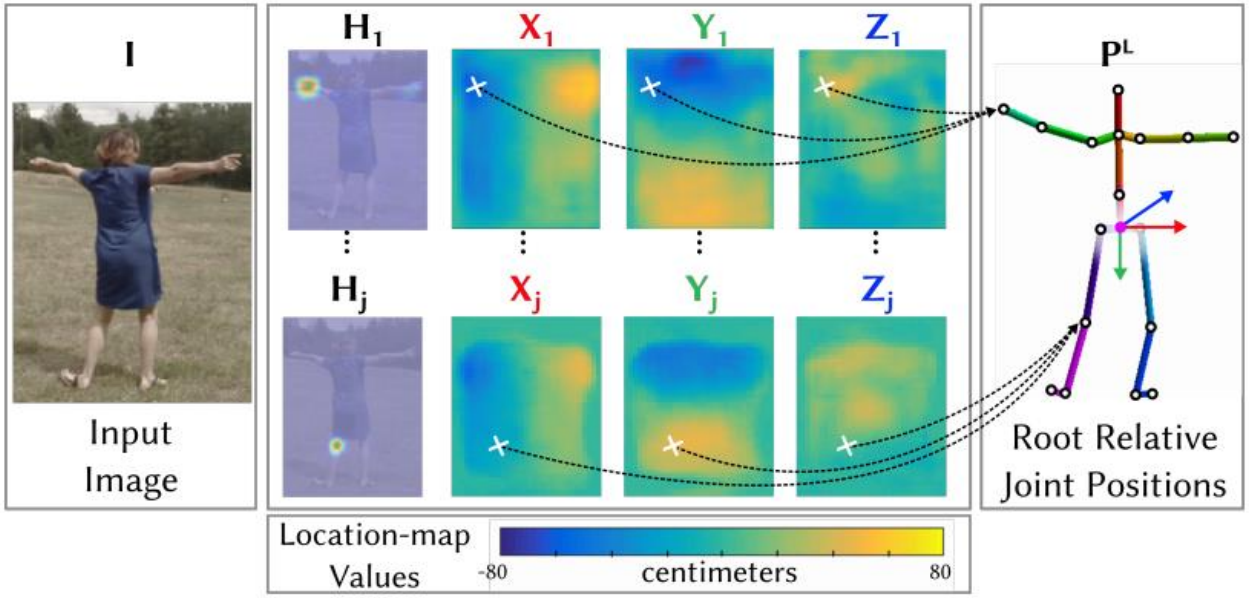


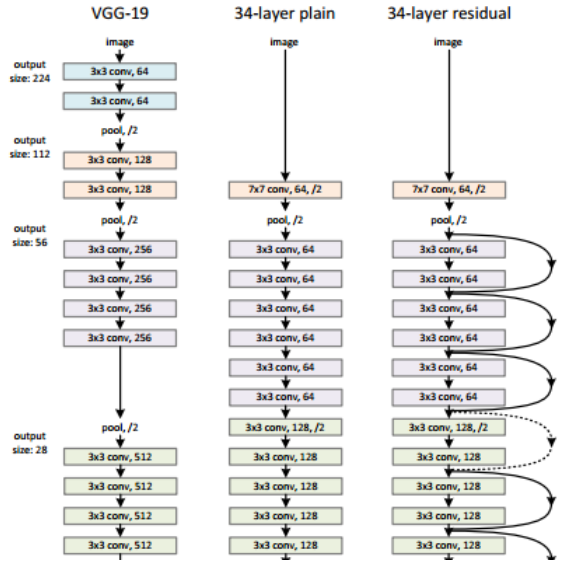
Fig. 3. Schema of the fully-convolutional formulation for predicting root relative joint locations. For each joint j, the 3D coordinates are predicted from their respective *location-maps* $\mathbf{X}_j$, $\mathbf{Y}_j$, $\mathbf{Z}_j$ at the position of the maximum in the corresponding 2D heatmap $\mathbf{H}_j$. The structure observed here in the location-maps emerges due to the spatial loss formulation. See Section 4.1.

Training data

Fig. 4. Representative training frames from Human3.6m and MPI-INF-3DHP 3D pose datasets. Also shown are the background, clothing and occluder augmentations done on MPI-INF-3DHP training data.

Architecture

ResNet reminder



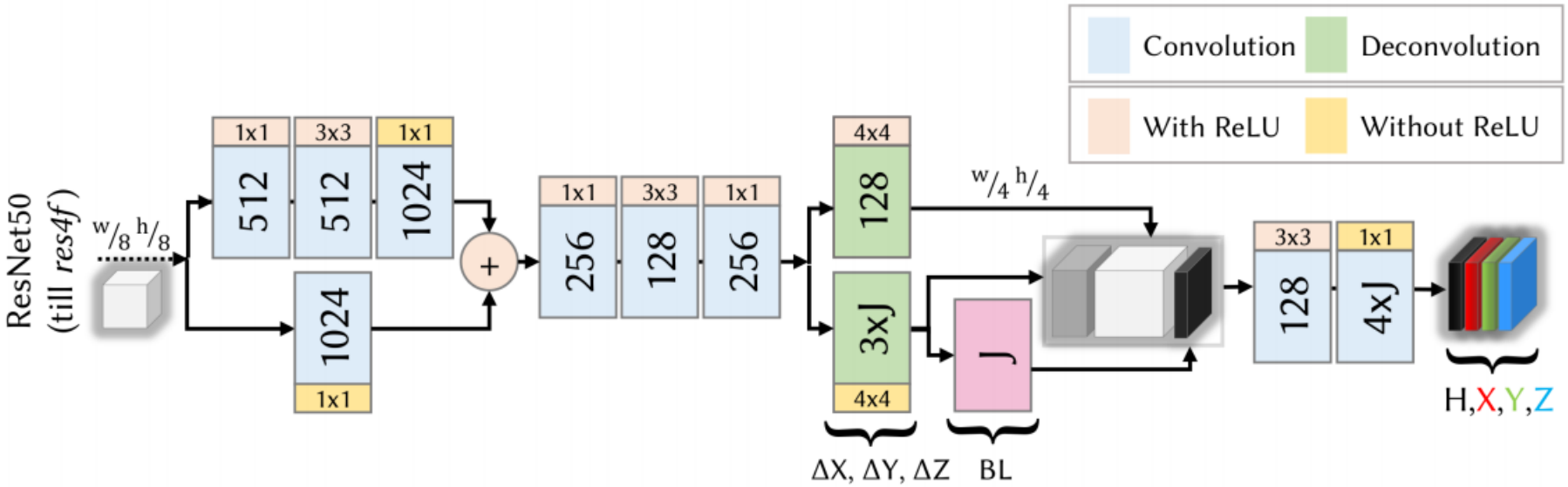
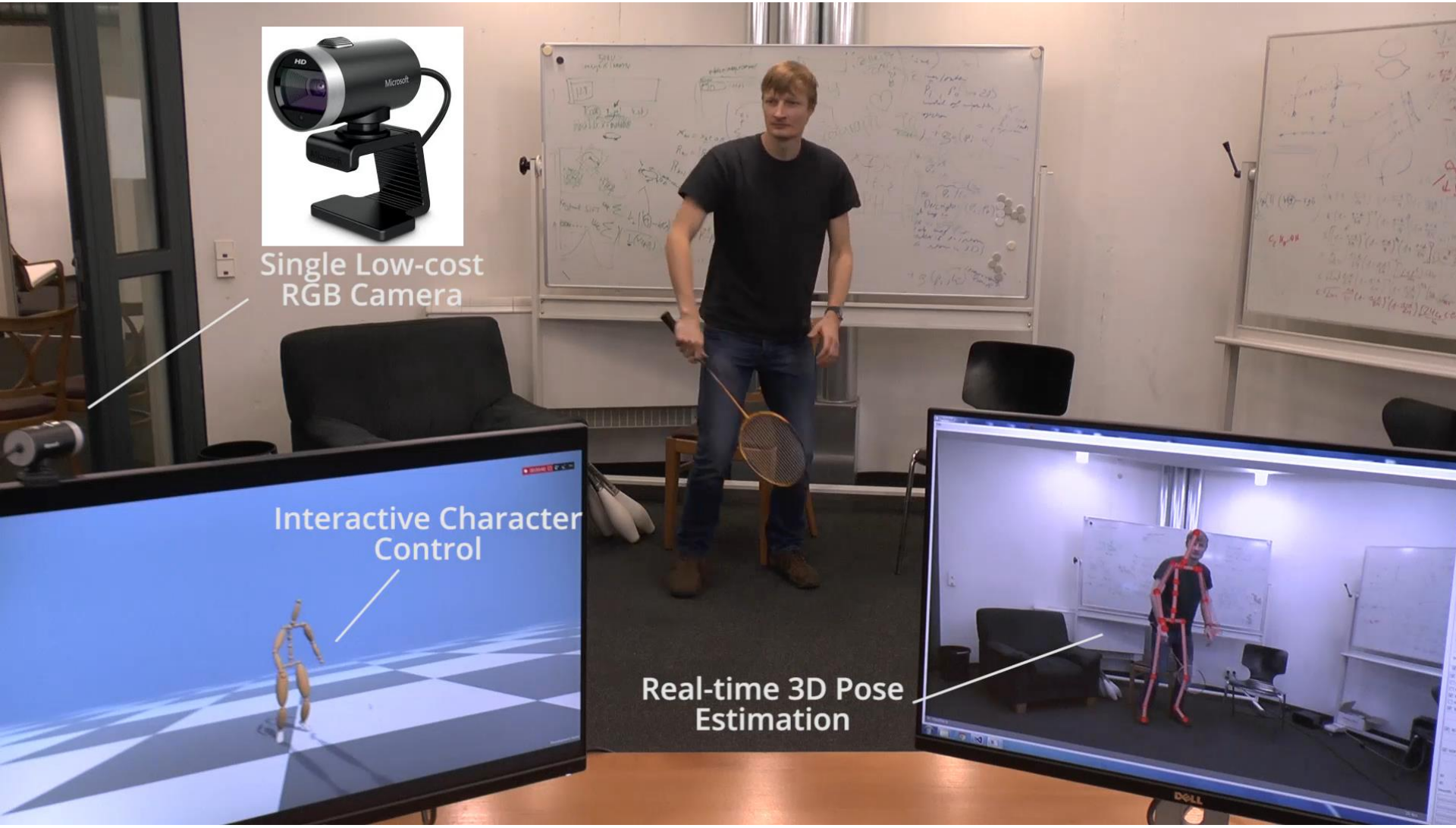
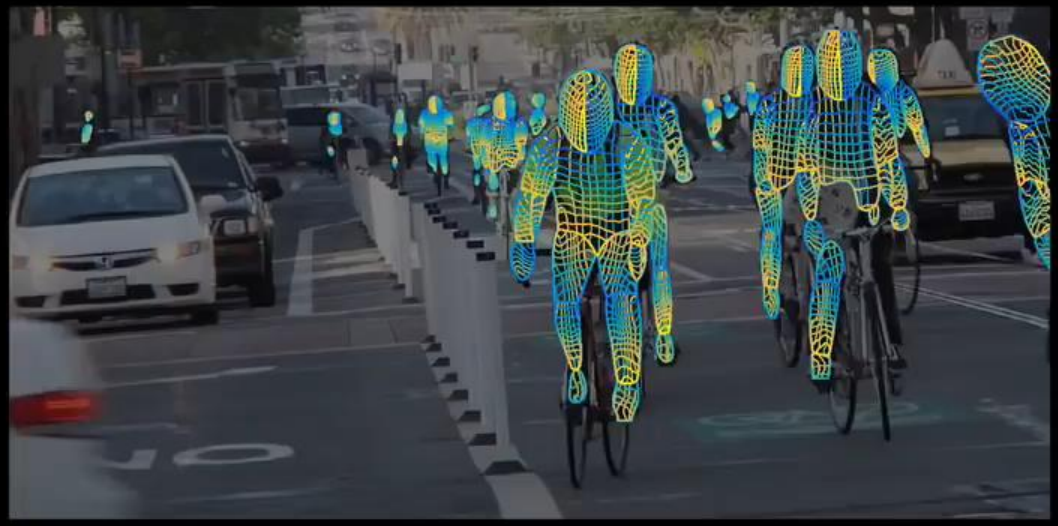


Fig. 5. Network Structure. The structure above is preceded by ResNet50/100 till level 4. We use kinematic parent relative 3D joint location predictions **ΔX**, **ΔY**, **ΔZ** as well as bone length maps **BL** constructed from these as auxiliary tasks.The network predicts 2D location heatmaps **H** and root relative 3D joint locations **X**, **Y**, **Z**. Refer to Section 4.1.


Single Low-cost
RGB Camera
Interactive Character
Control
Real-time 3D Pose
Estimation

http://densepose.org/

DensePose:
Dense Human Pose Estimation In The Wild



Rıza Alp Güler *
INRIA, CentraleSupélec

Natalia Neverova
Facebook AI Research

Iasonas Kokkinos
Facebook AI Research

* Rıza Alp Güler was with Facebook AI Research during this work.

[Güler et al. CVPR 2018]