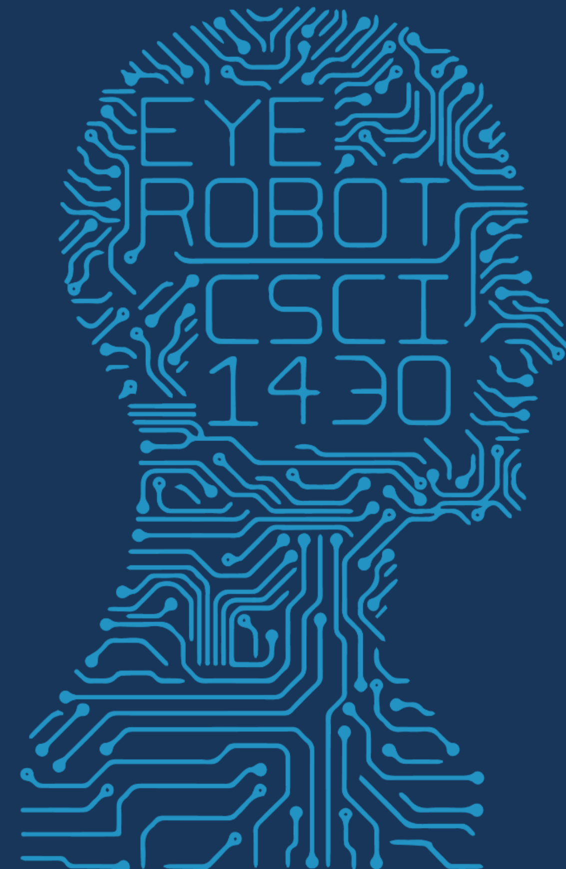
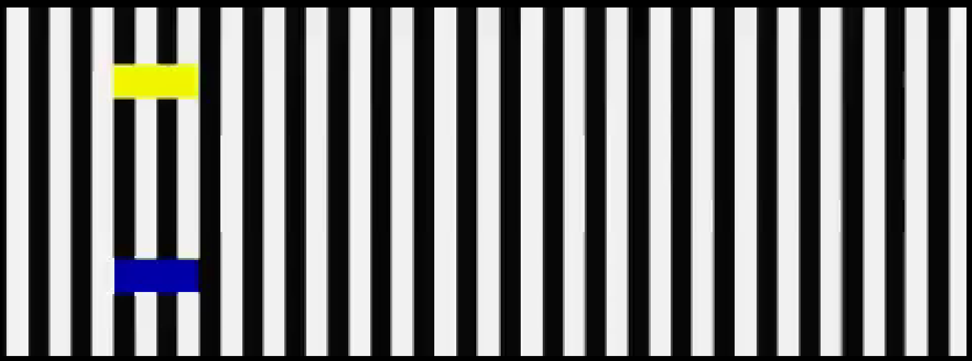
1950

Future Vision

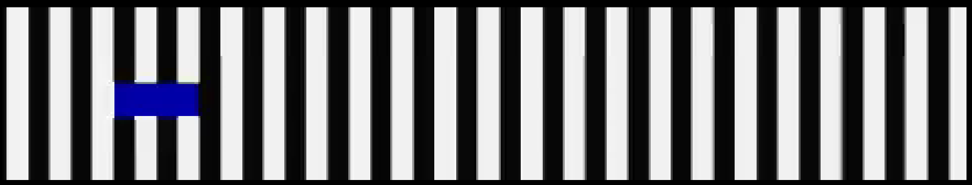


2017 MWF 1pm

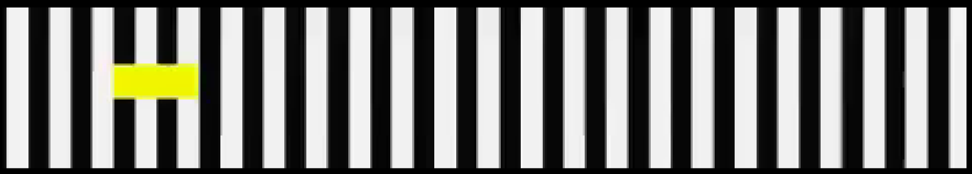
Computer Vision



[Footsteps and Inchworms – Anstis, Perception 2001]



[Footsteps and Inchworms – Anstis, Perception 2001]



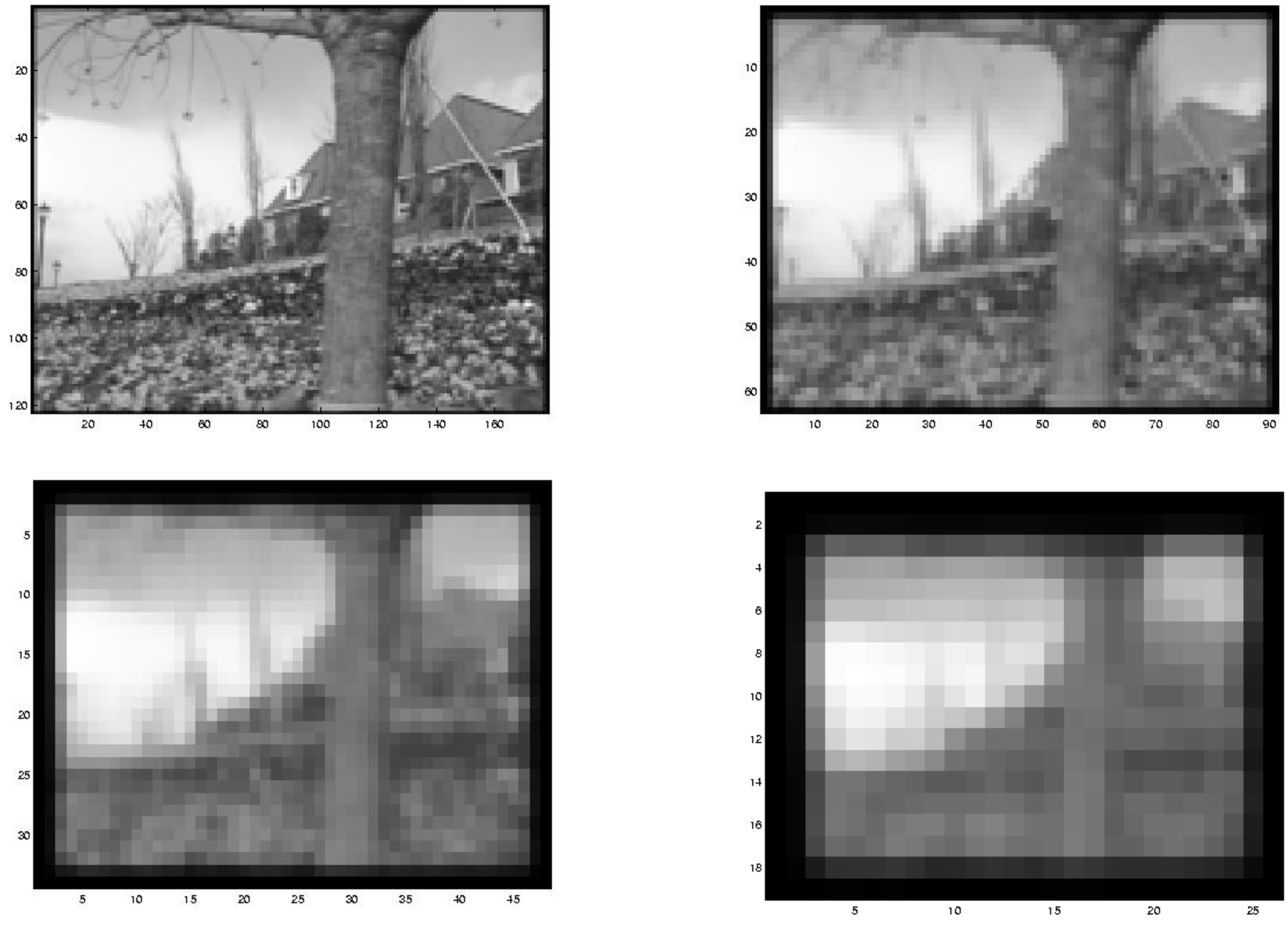
[Footsteps and Inchworms – Anstis, Perception 2001]

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

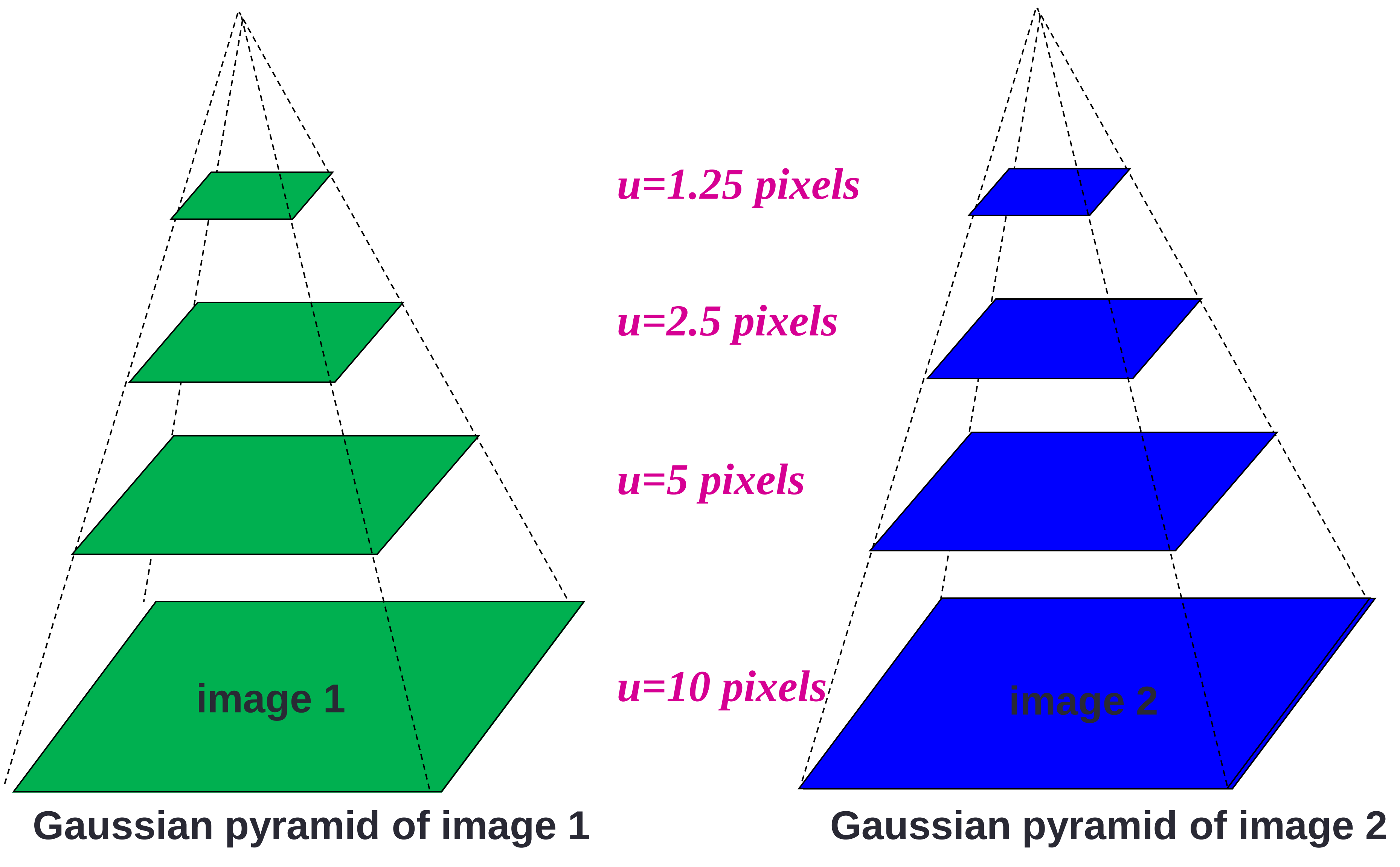
Optical flow

Gradient is informative of direction over only < 1 pixel

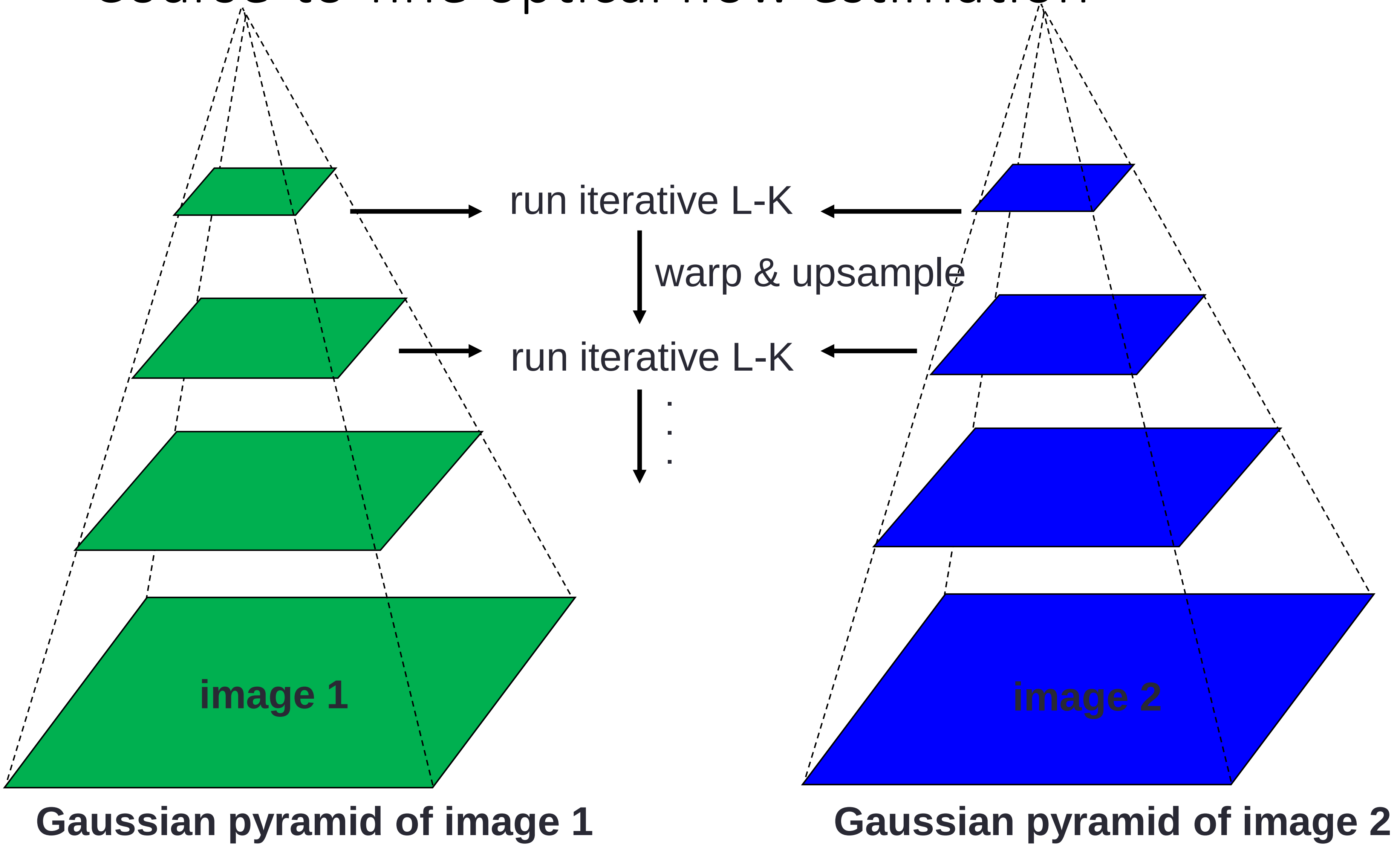
Reduce the resolution!



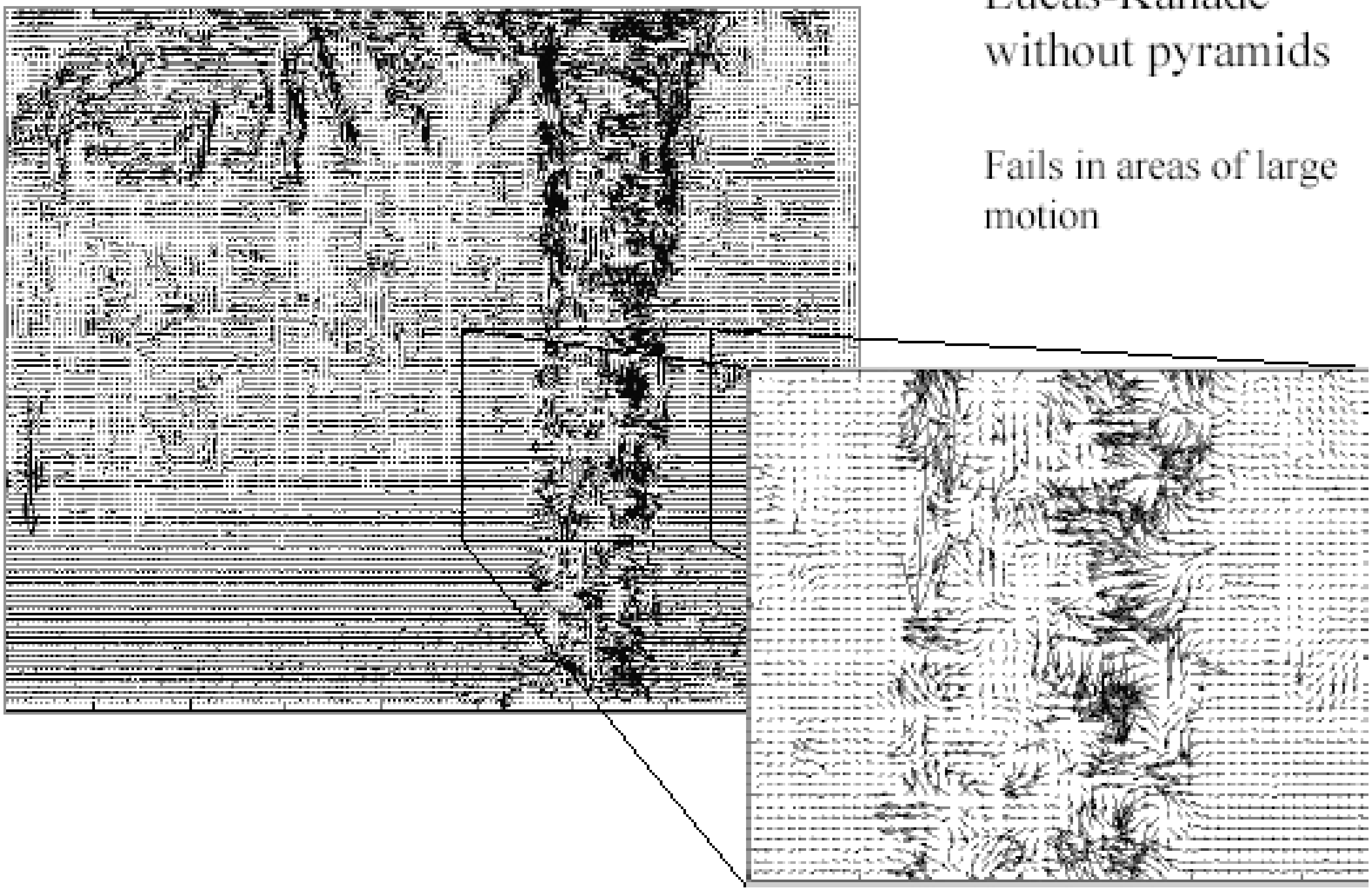
Coarse-to-fine optical flow estimation



Coarse-to-fine optical flow estimation



Optical Flow Results

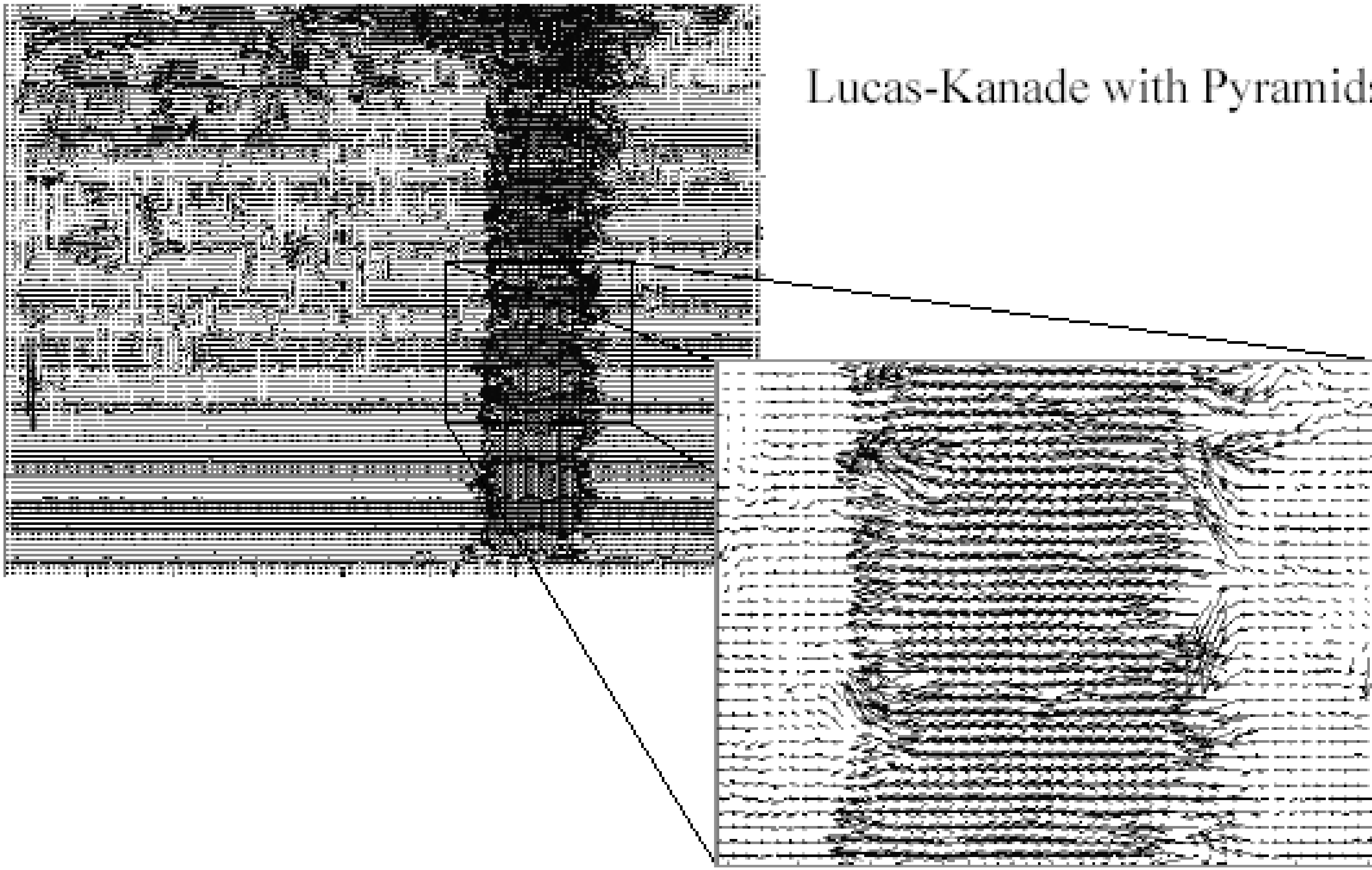


Lucas-Kanade without pyramids

Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



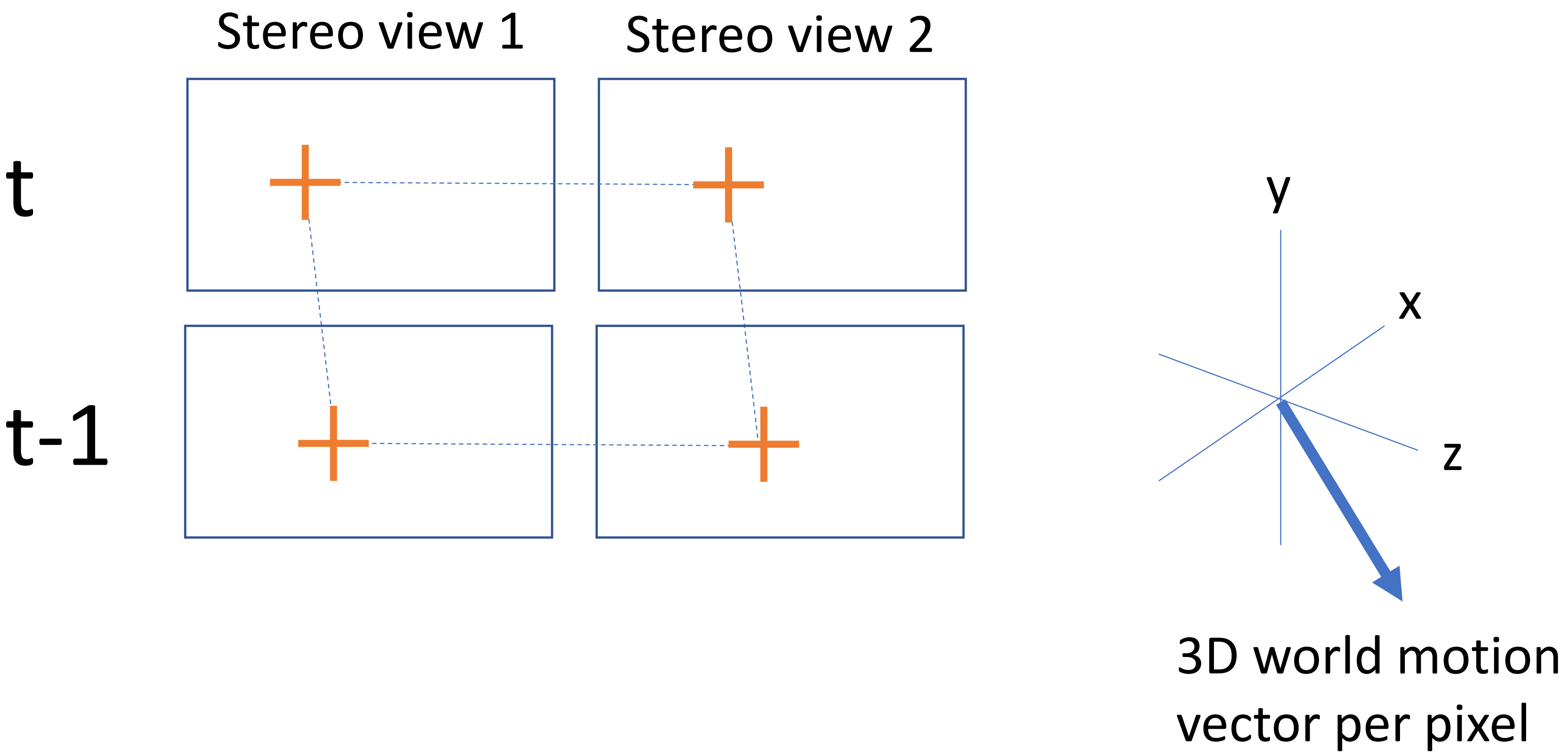
Lucas-Kanade with Pyramids

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

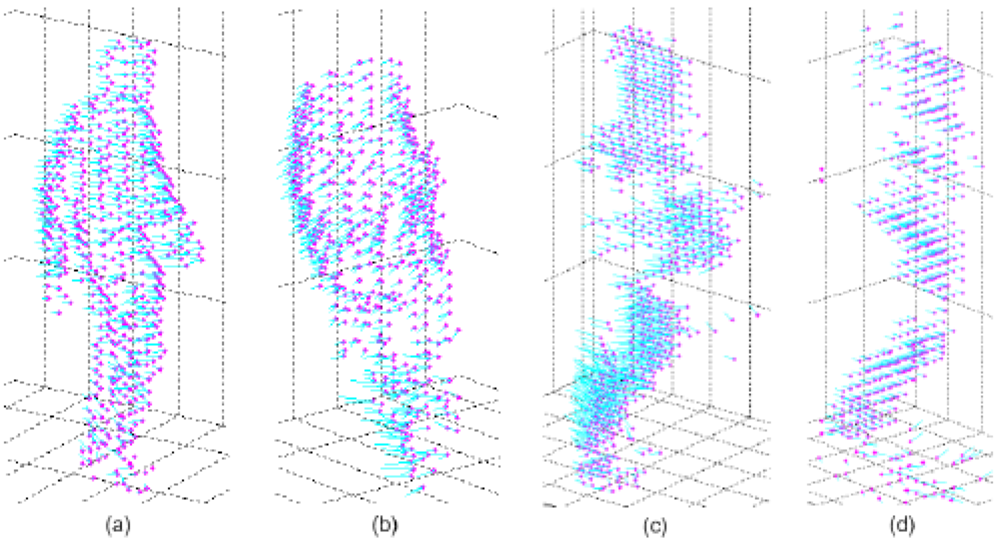
Can we do more? *Scene flow*

Combine spatial stereo & temporal constraints

Recover 3D vectors of world motion



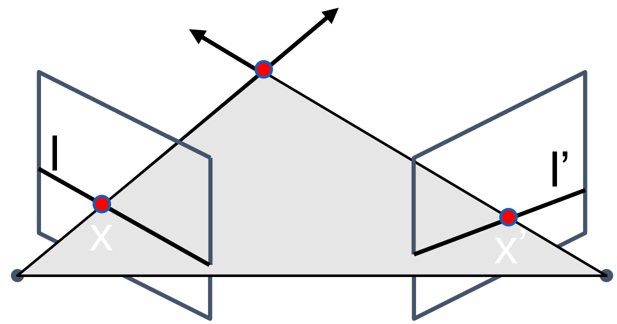
Scene flow example for human motion

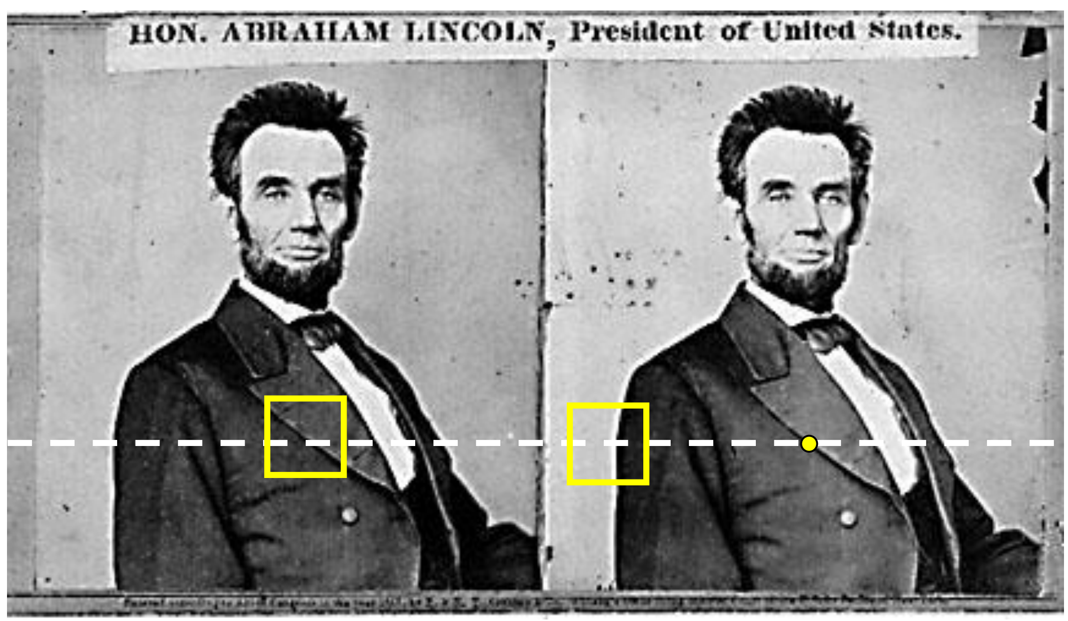


Estimating 3D Scene Flow from Multiple 2D Optical Flows, Ruttle et al., 2009

Stereo correspondence

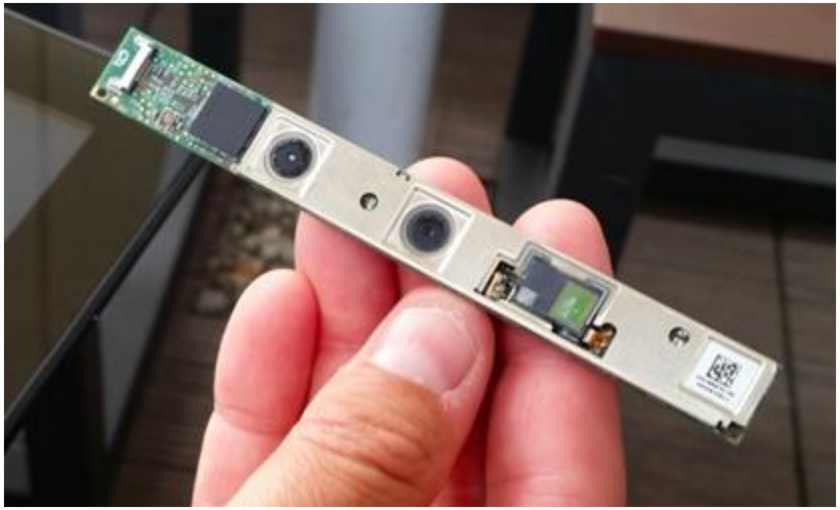
- Let *x* be a point in left image, *x'* in right image
- Epipolar relation
  - *x* maps to epipolar line *l'*
  - *x'* maps to epipolar line *l*





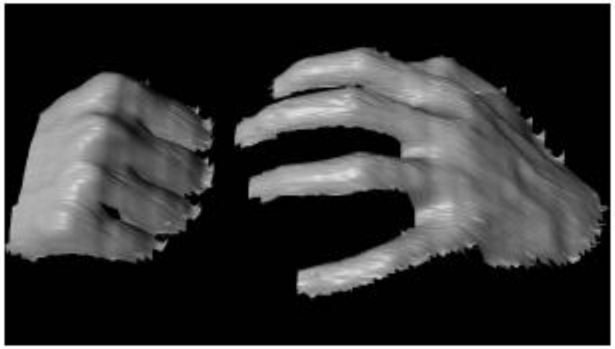
How does a depth camera work?

Microsoft Kinect v1



Intel laptop depth camera

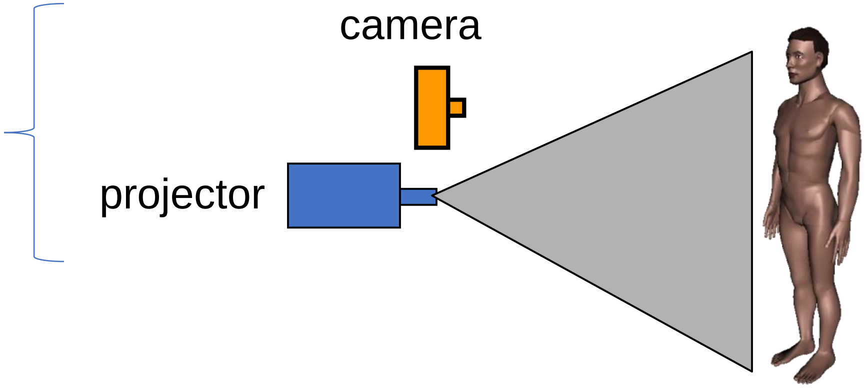
Active stereo with structured light



Project "structured" light patterns onto the object

- Simplifies the correspondence problem
- Allows us to use only one camera

Stereo system!



L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming.](#) *3DPVT* 2002

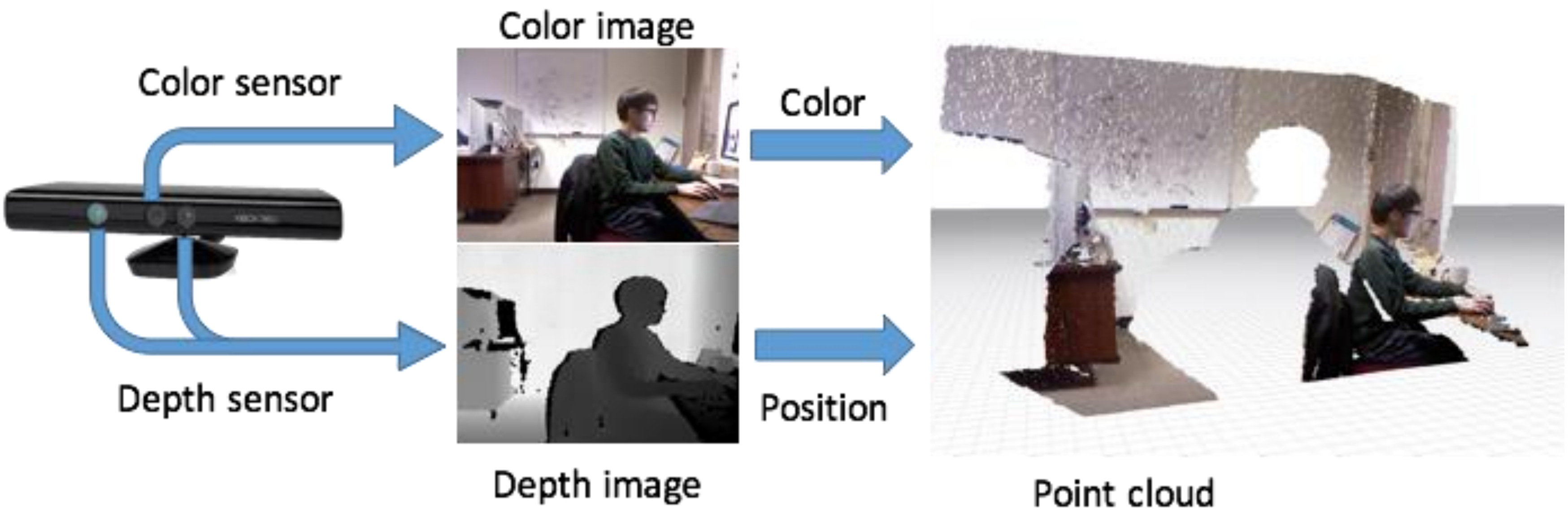
Kinect: Structured infrared light

http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/

With either technique…

…I gain depth maps over time.

Optex Depth Camera Based on Canesta Solution



Yisheng Zhou

Demo

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

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

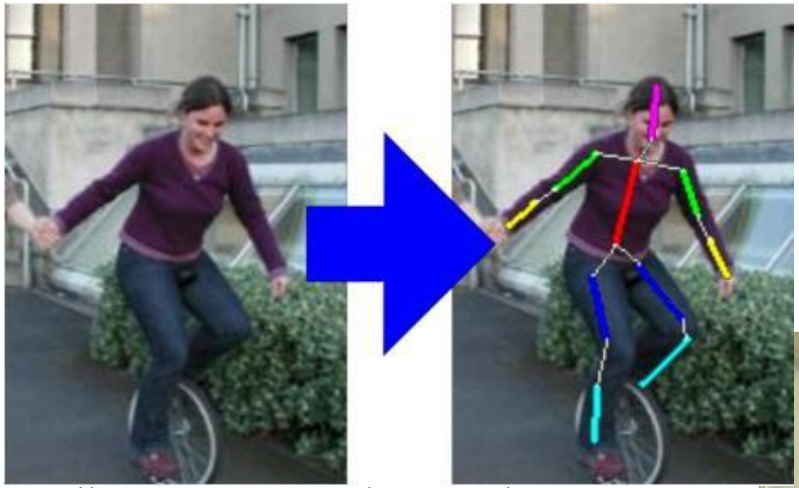
Background

- **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

Background

- **Depth Image**
  - **Each pixel has distance information, instead of RGB**

RGB Image

RGB Camera

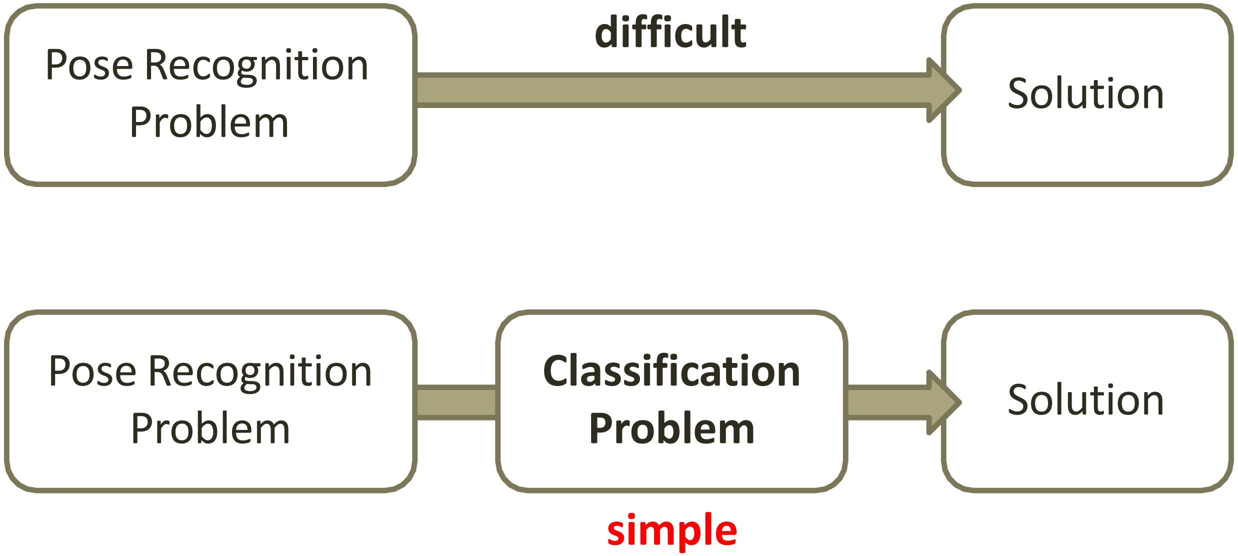
Depth Image

Depth Camera

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

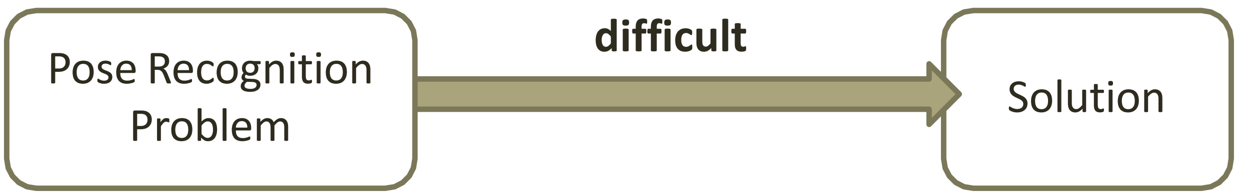
Why this paper?

- **Main Contribution**
  - **Convert pose recognition problem to classification problem**
    - One of application for image retrieval technique

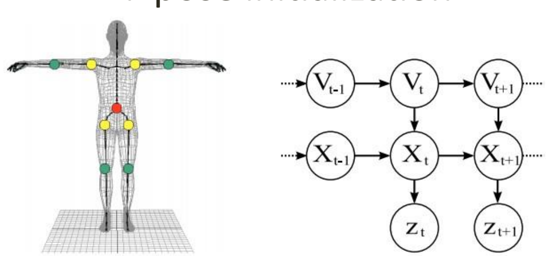


Why this paper?

- **Main Contribution**
  - **Convert pose recognition problem to classification problem**
    - One of application for image retrieval technique

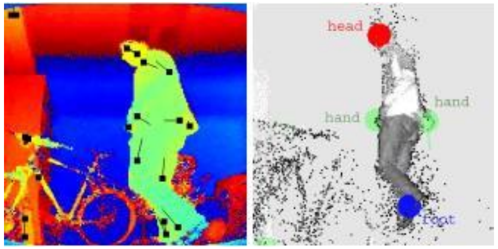


Kinematic constraint
T-pose initialization



[1] V. Ganapathi et al.,
Real-Time motion Capture
using a Single Time-of-Flight camera,
CVPR, 2010

Limited patches
Only 3 parts



[2] C. Palgemann et al.,
Real-Time Identification and Localization
of Body Parts from Depth Images,
ICRA, 2010

Why this paper?

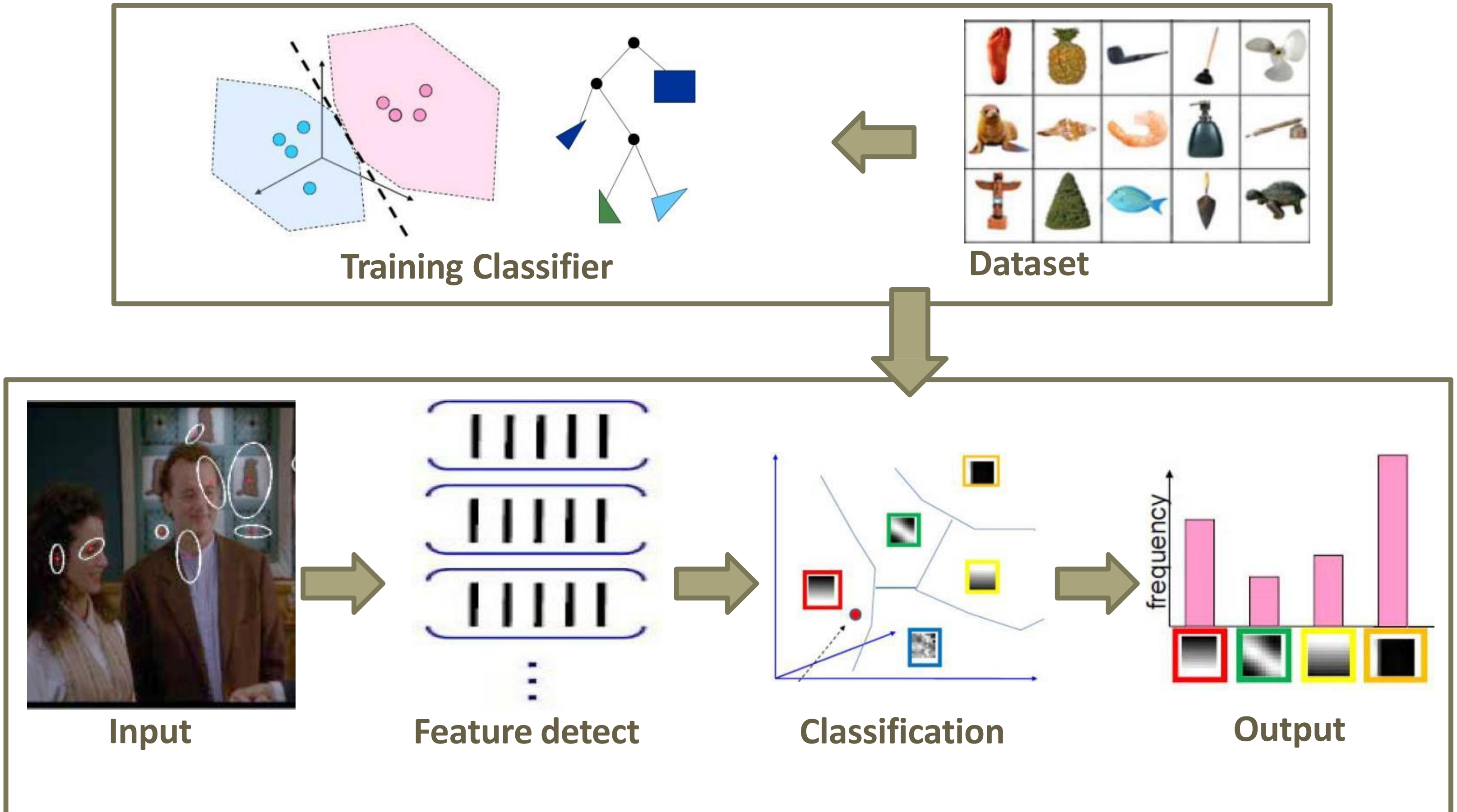
- **Main Contribution**
  - **Convert pose recognition problem to classification problem**
    - One of application for image retrieval technique

Pose Recognition Problem → **Classification Problem** → Solution

No constraint, More General

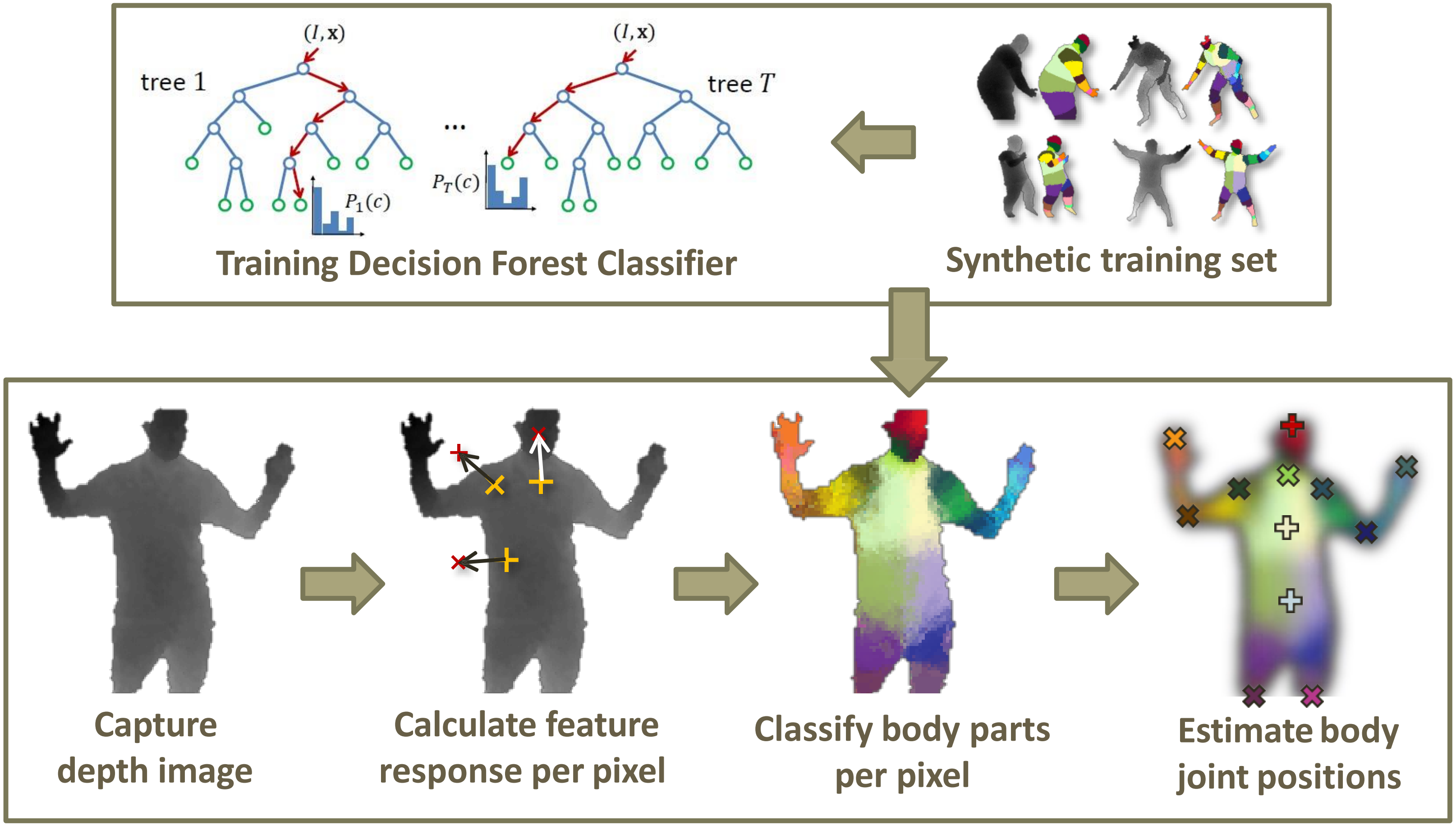
Overview

- **Overview**



Overview

- **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

Supplementary Material

Depth Image Feature Comparison

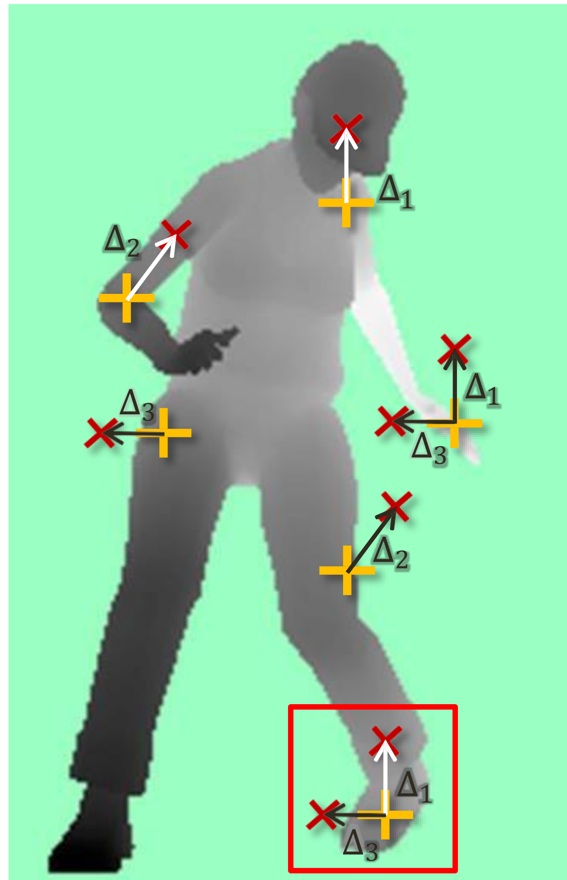
Calculate feature response for each pixel

Feature Response Function

$$f(\overbrace{I}^{\text{image}},\underbrace{\mathbf{x}}_{\text{pixel}}) = \overbrace{d_I(\mathbf{x})}^{\text{depth}} - \overbrace{d_I(\mathbf{x}+\Delta)}^{\text{offset depth}}$$

- Δ is chosen in training step randomly
- For example
  - $\Delta_1 = (0\,,1)$ $\qquad \Delta_3 = (-1\,,0)$
  - $f(I, \boldsymbol{x}|\Delta_1)$ has small value
  - $f(I, \boldsymbol{x}|\Delta_3)$ has large value
- Can be trained in parallel on GPUs

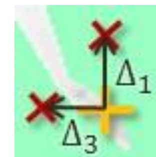
Input depth image

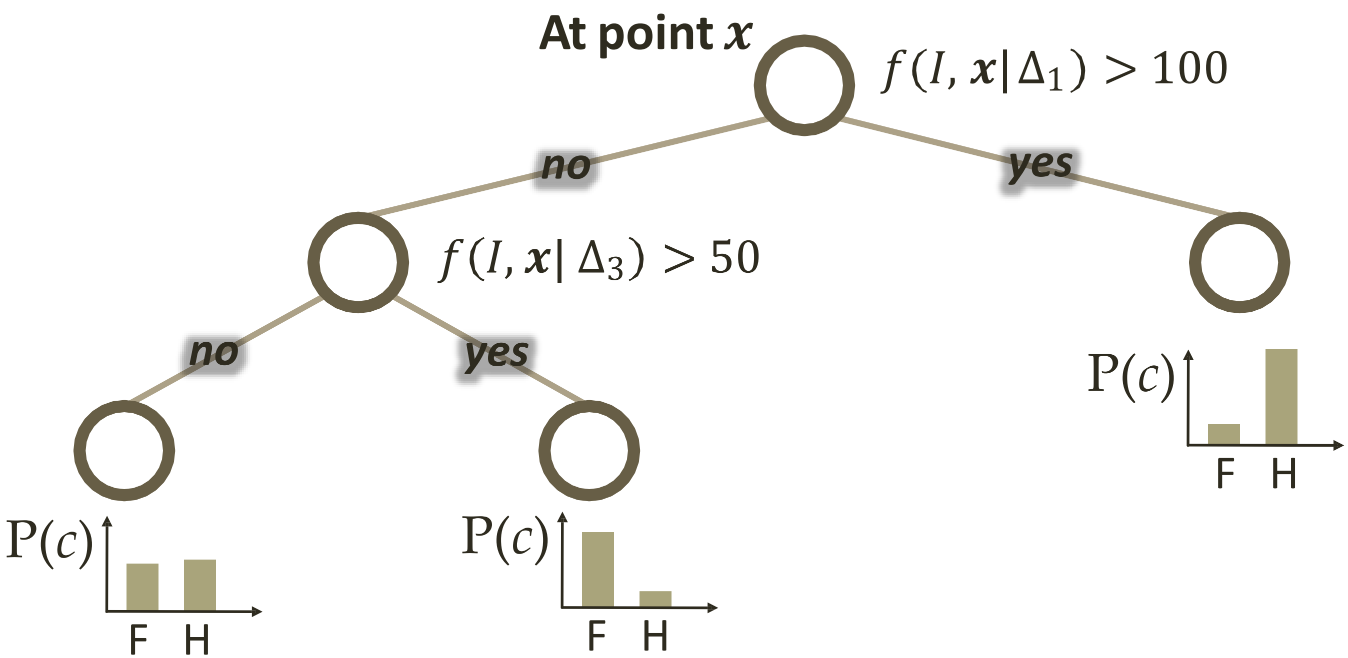


[3] V. Lepetit et al., Randomized trees for real-time keypoint recognition, CVPR, 2005

Decision tree classifier

- **Remember Viola-Jones face detector?**
- **Example of classification for hand(H) or foot(F)**



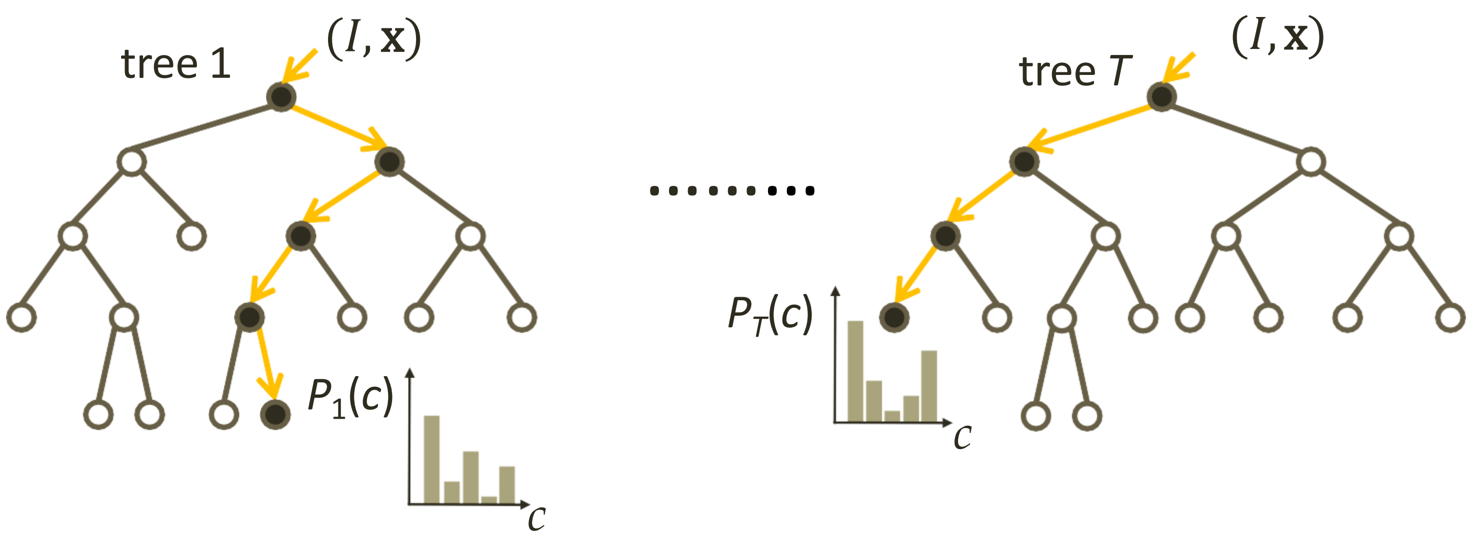


4 T. Amit et al., Shape quantization and recognition with randomized trees, Neural Computation, 1997
5 L. Breiman, Random forests, Mach. Learning, 2001
6 F. Moosmann et al., Fast discriminative visual codebooks using randomized clustering forests, NIPS, 2006

Decision Forest Classifier

- In training step, Δ is chosen randomly
- Generate *many trees* to build a decision forest

- In testing step, check all trees and compute average probability

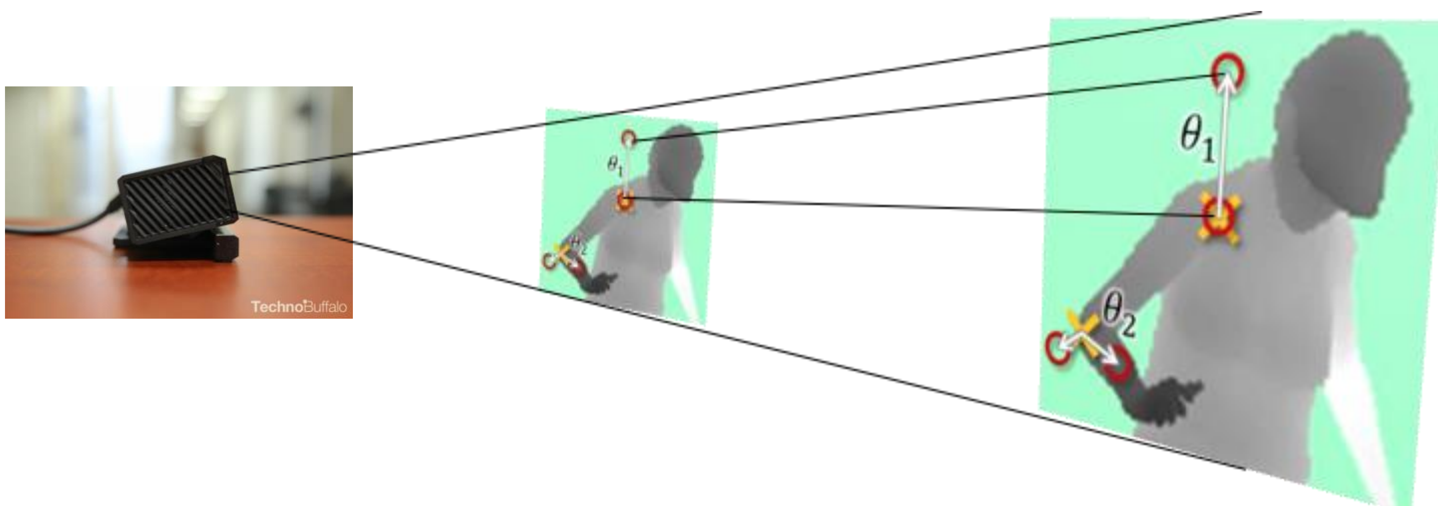


$$P(c|I,\mathbf{x}) = \frac{1}{T}\sum_{t}^{T} P_t(x, I|c)$$

But…normalized in depth

$\frac{1}{d_I(\mathbf{x})}$

- 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

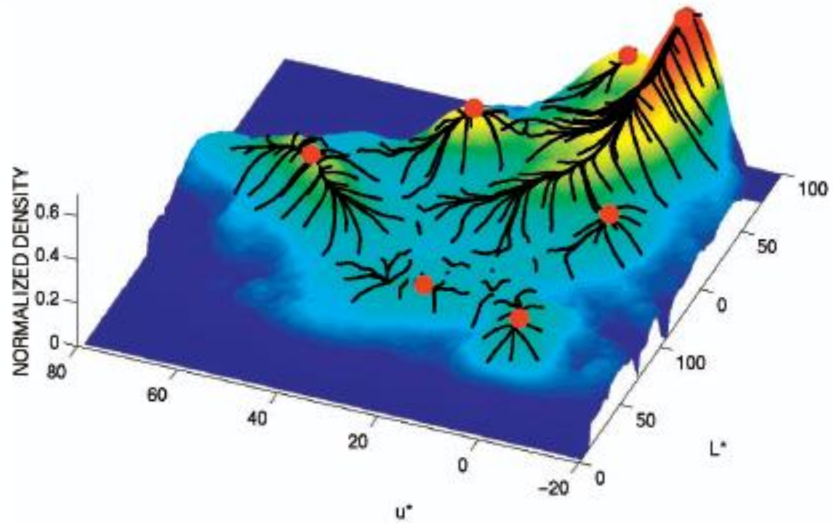
- **Find mode using mean shift algorithm**
  - With weighted Gaussian kernel
  - Using class probabilities for each pixel, find **representative positions** of classes

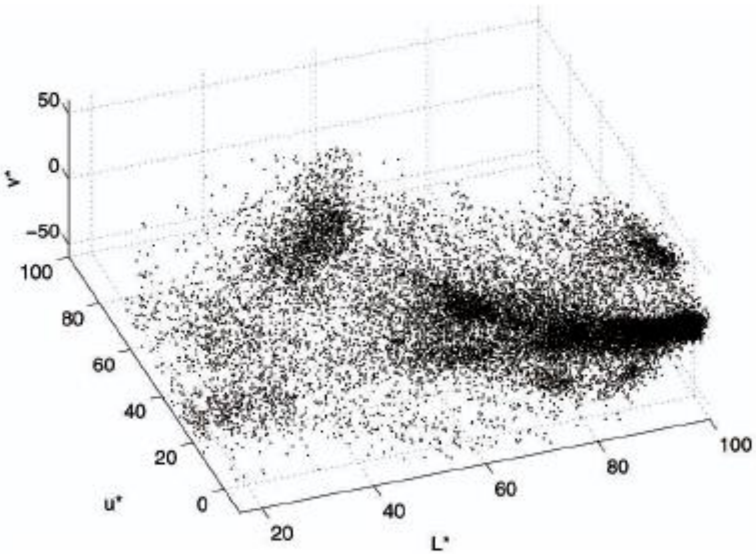
Estimate body joint positions

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

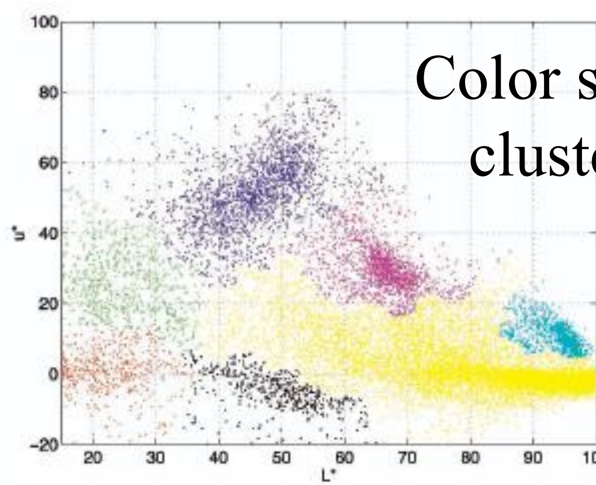
Mean shift algorithm

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



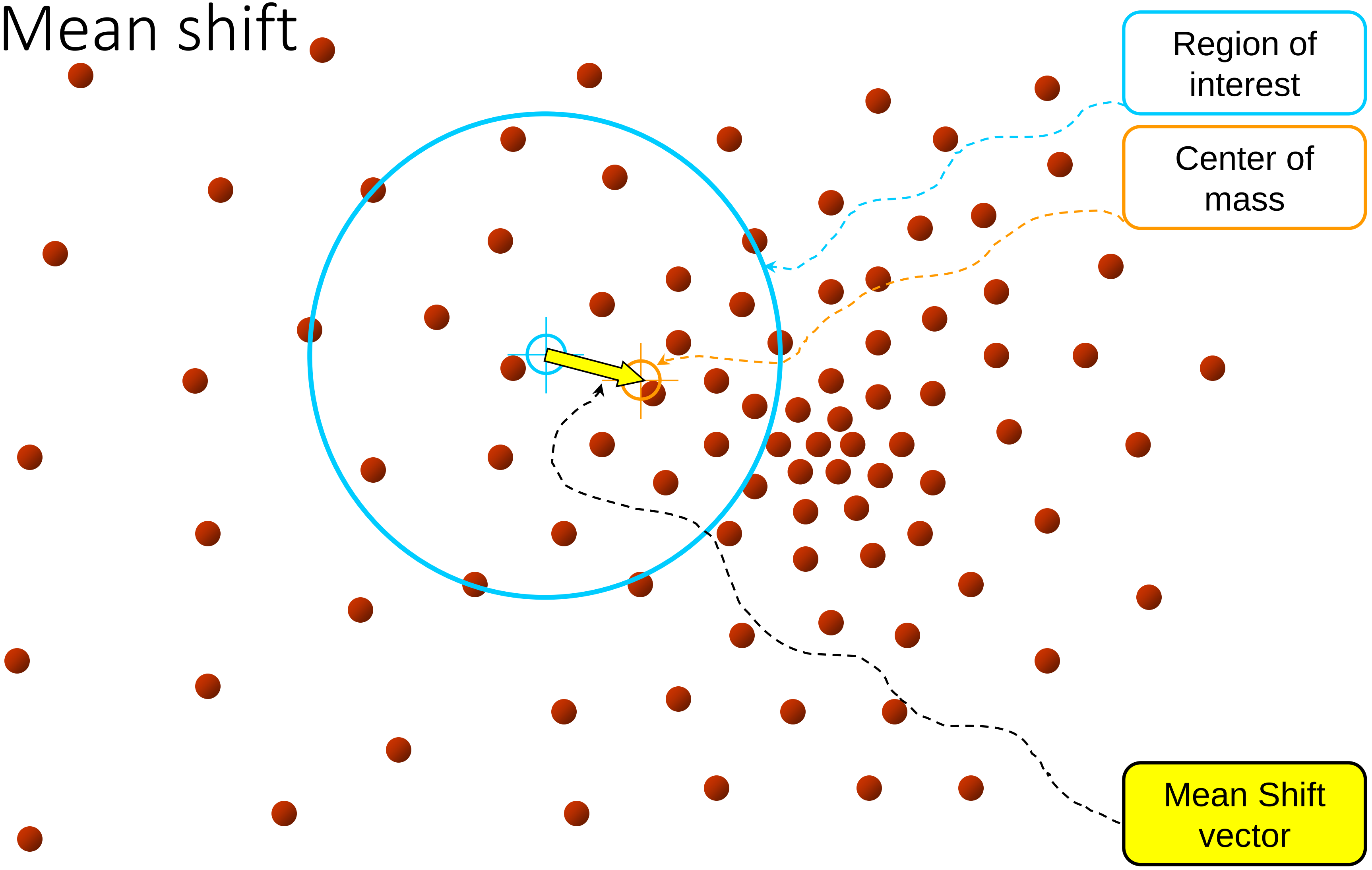


Color space



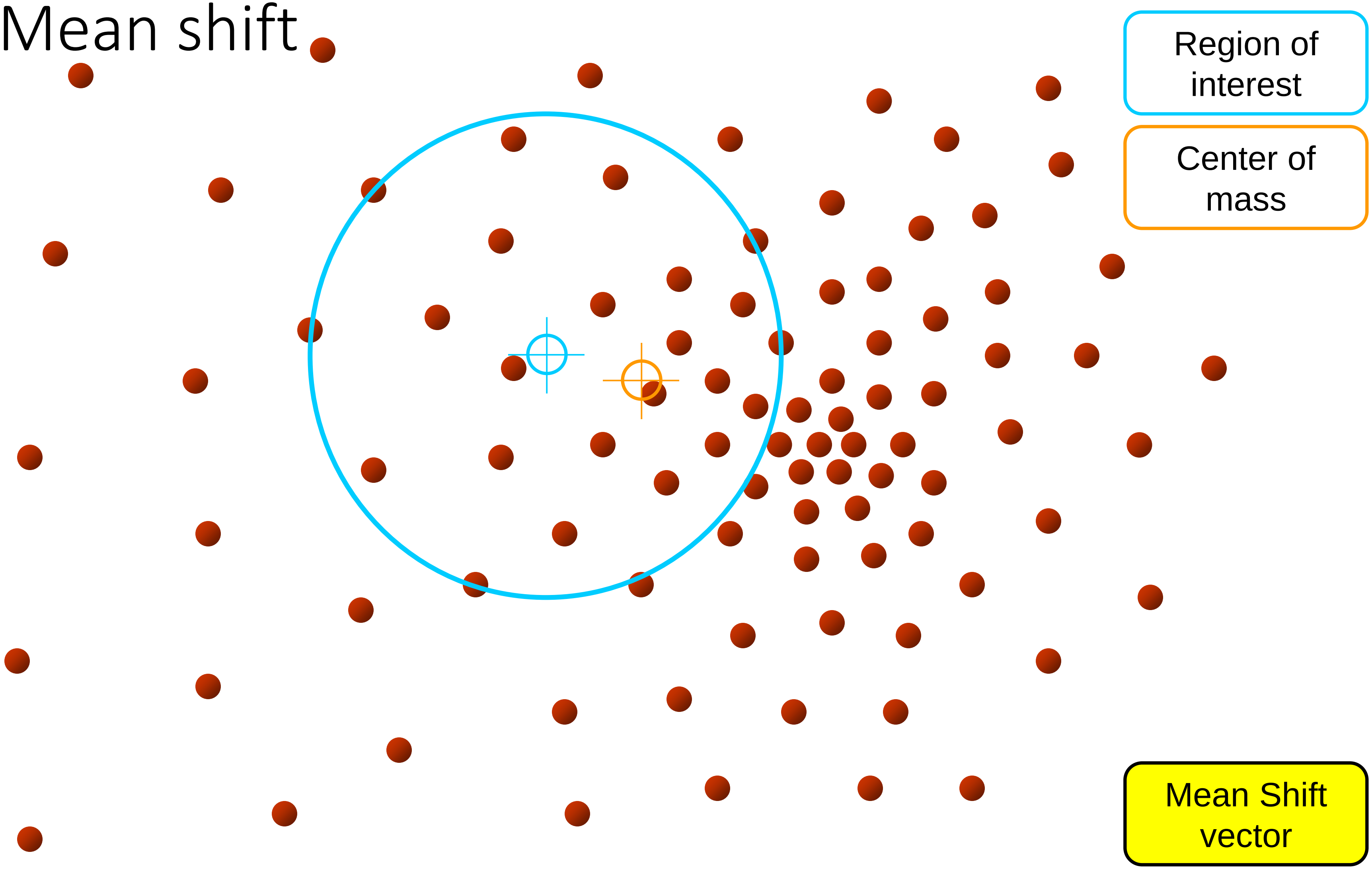
Color space clusters

Mean shift



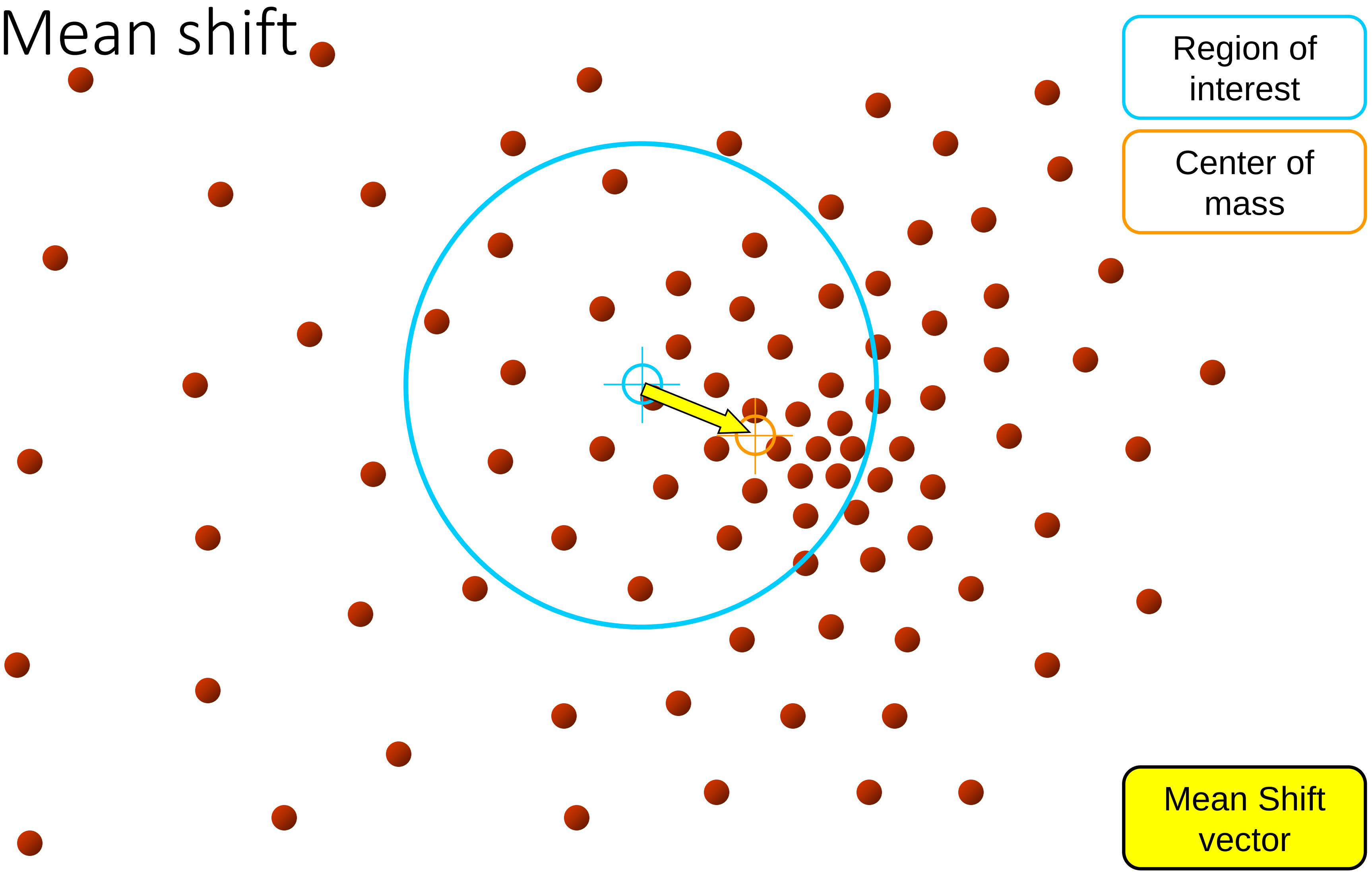
Slide by Y. Ukrainitz & B. Sarel

Mean shift



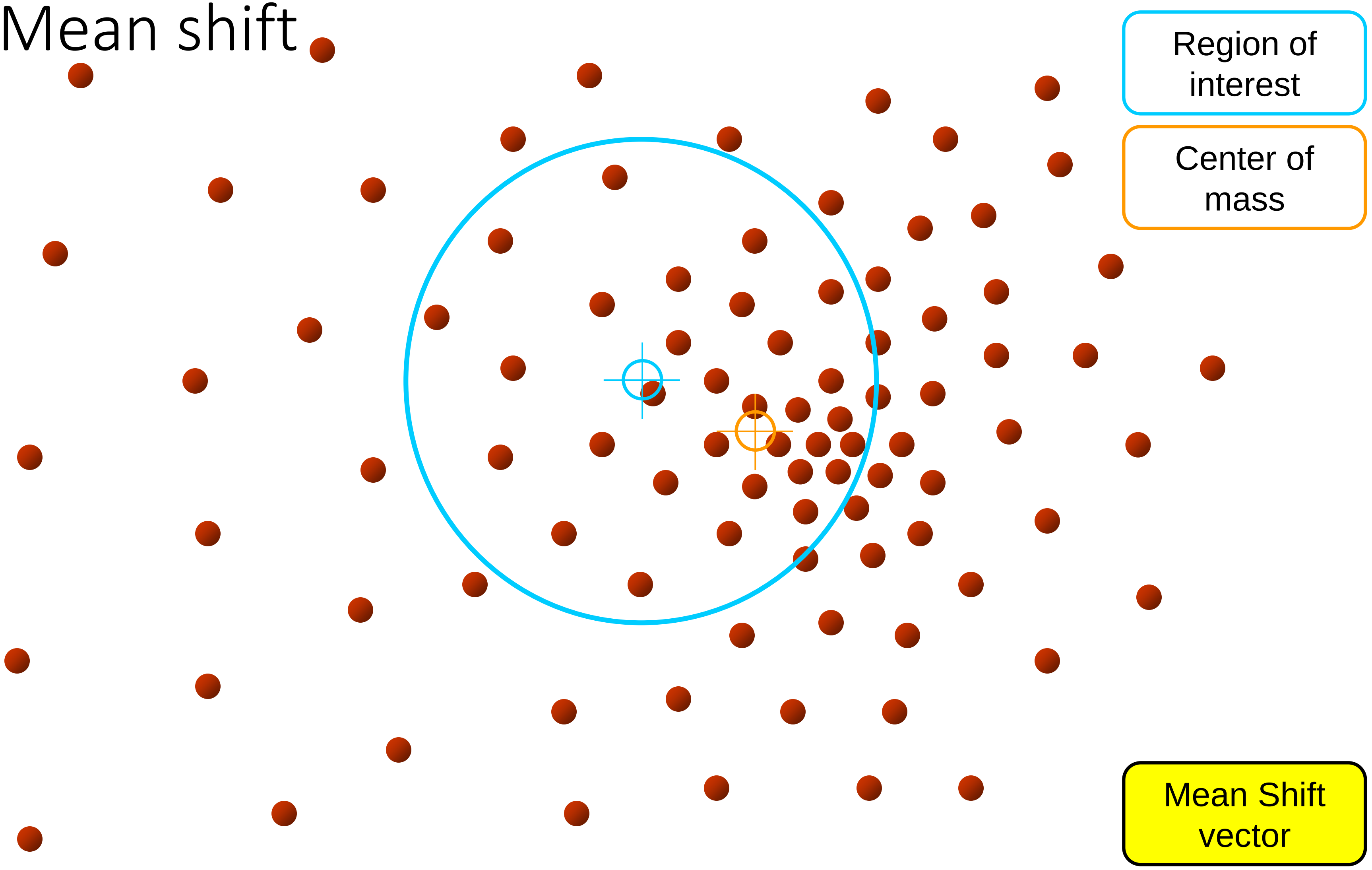
Slide by Y. Ukrainitz & B. Sarel

Mean shift



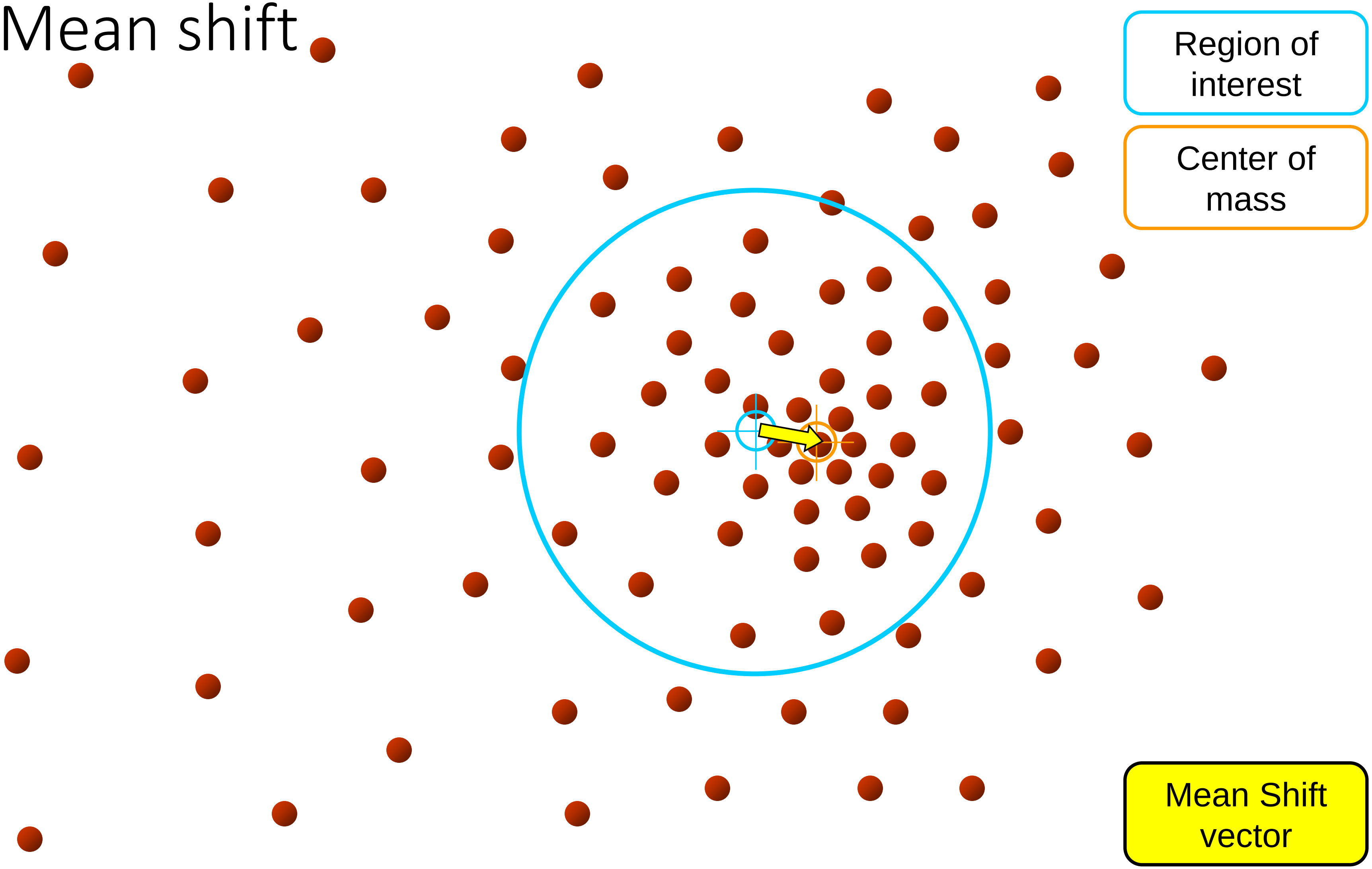
Slide by Y. Ukrainitz & B. Sarel

Mean shift



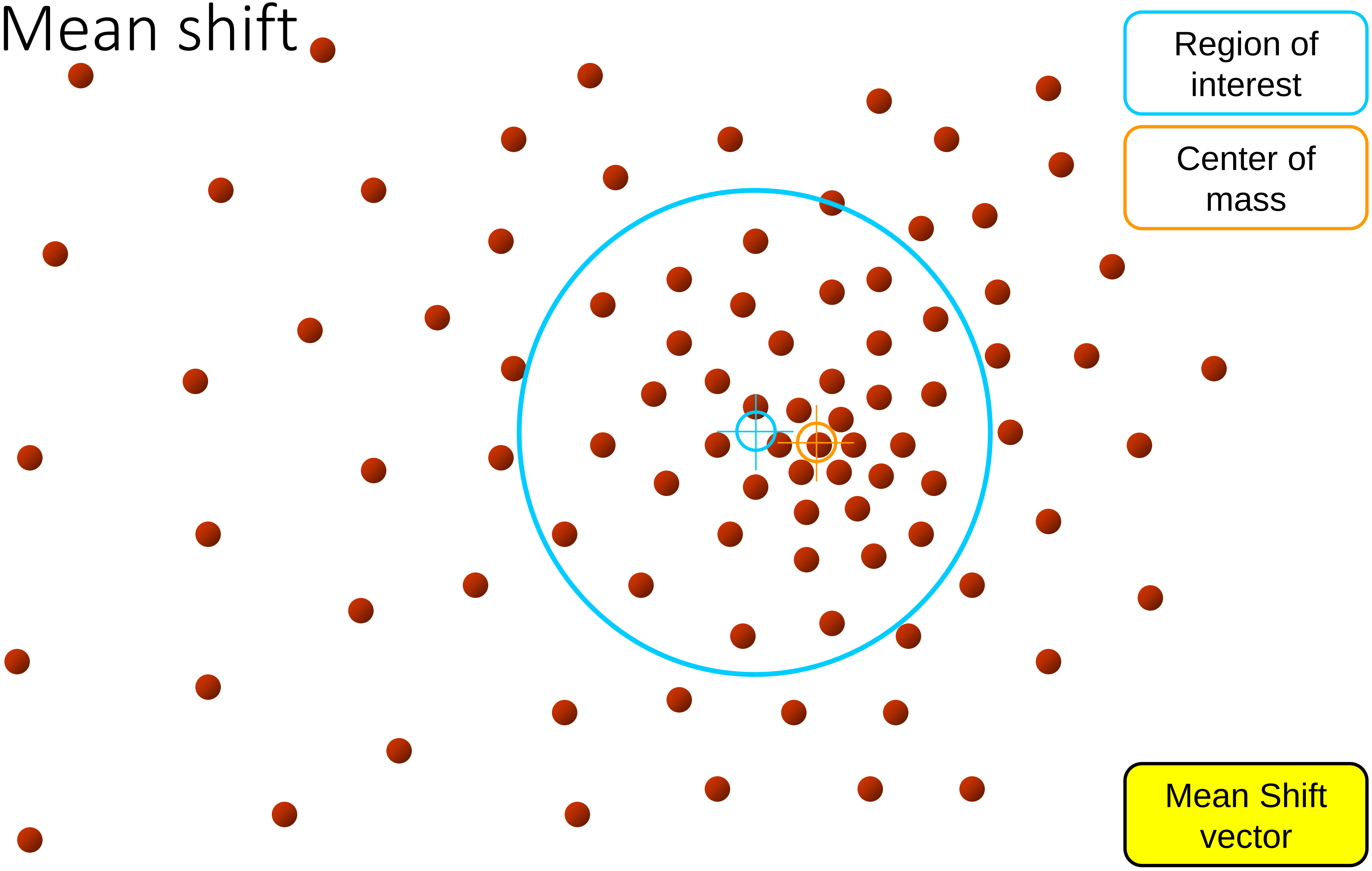
Slide by Y. Ukrainitz & B. Sarel

Mean shift



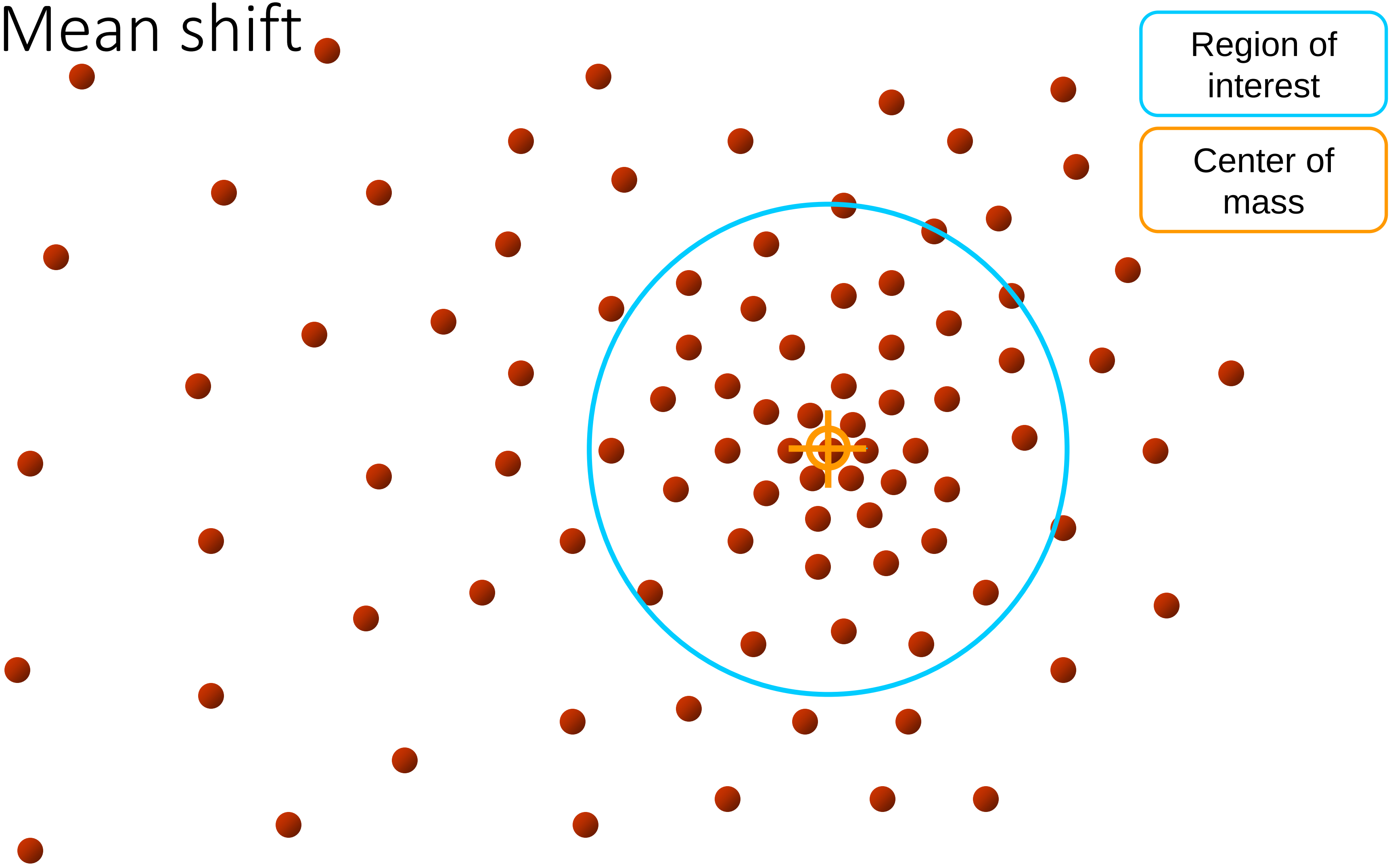
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Kernel density estimation

Kernel density estimation function

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

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 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

$$f_c(\overbrace{\hat{\mathbf{x}}}^{\text{3D position of class}}) \propto \sum_{\underbrace{i=1}_{\text{pixel index i}}}^{N} \overbrace{w_{ic}}^{\text{pixel weight}} \exp\left(-\left\|\frac{\hat{\mathbf{x}} - \overbrace{\hat{\mathbf{x}}_i}^{\text{3D position of i pixel}}}{\underbrace{b_c}_{\text{bandwidth}}}\right\|^2\right)$$

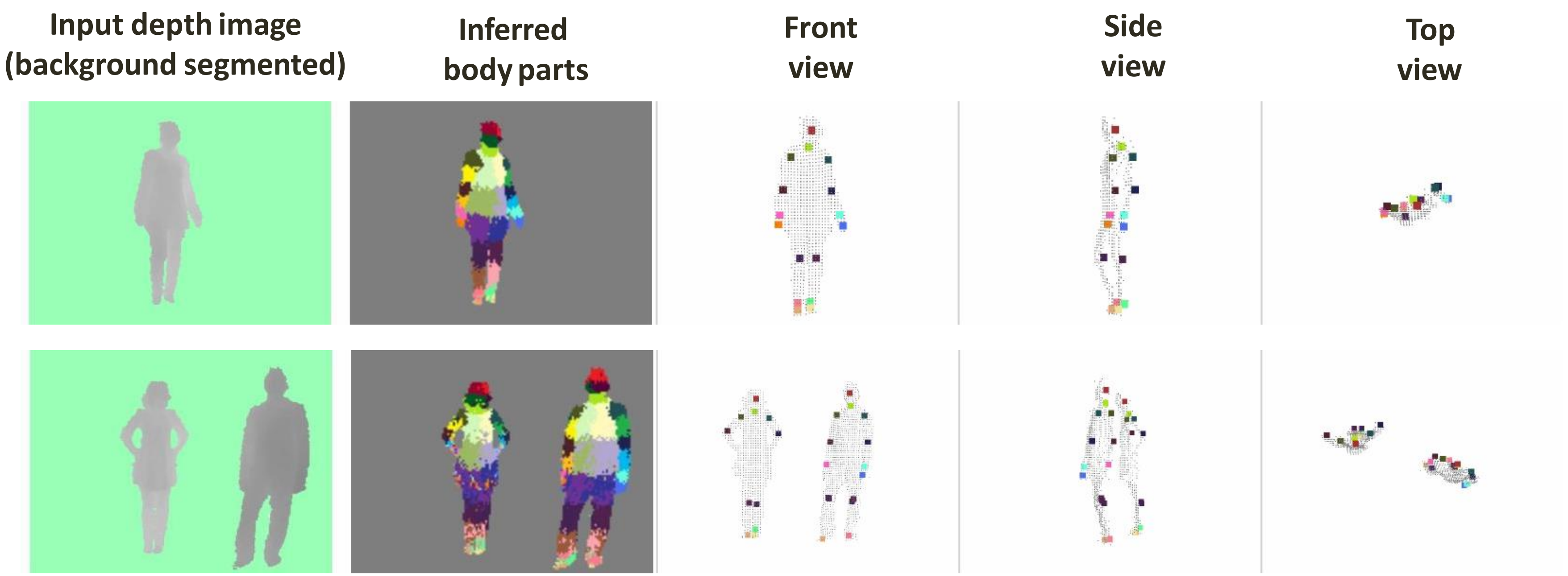
$$w_{ic} = \underbrace{P(c|I, \mathbf{x}_i)}_{\text{class probability}} \cdot \underbrace{d_I(\mathbf{x}_i)^2}_{\text{depth at i pixel}}$$

Estimate body joint positions

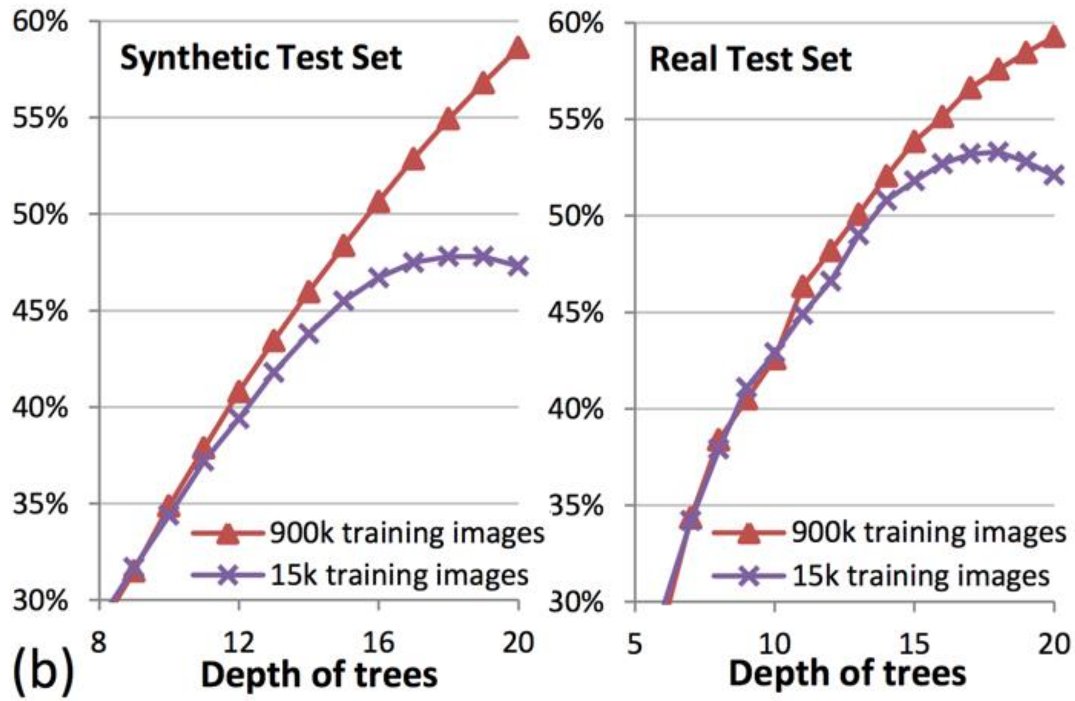
[7] S. Belongie et al., Mean shift: A robust approach toward feature space analysis, PAMI, 2002

Results

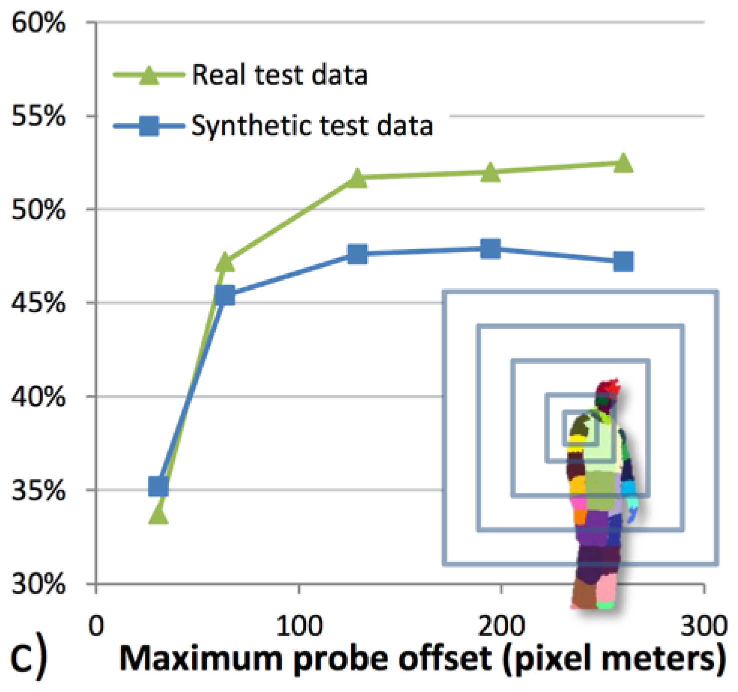
- **Fast Joint Proposals**
  - **Max. 200 FPS on Xbox 360 GPU, 50 FPS on 8 core CPU**
    - Previous work was 4 ~ 16FPS



Depth of trees



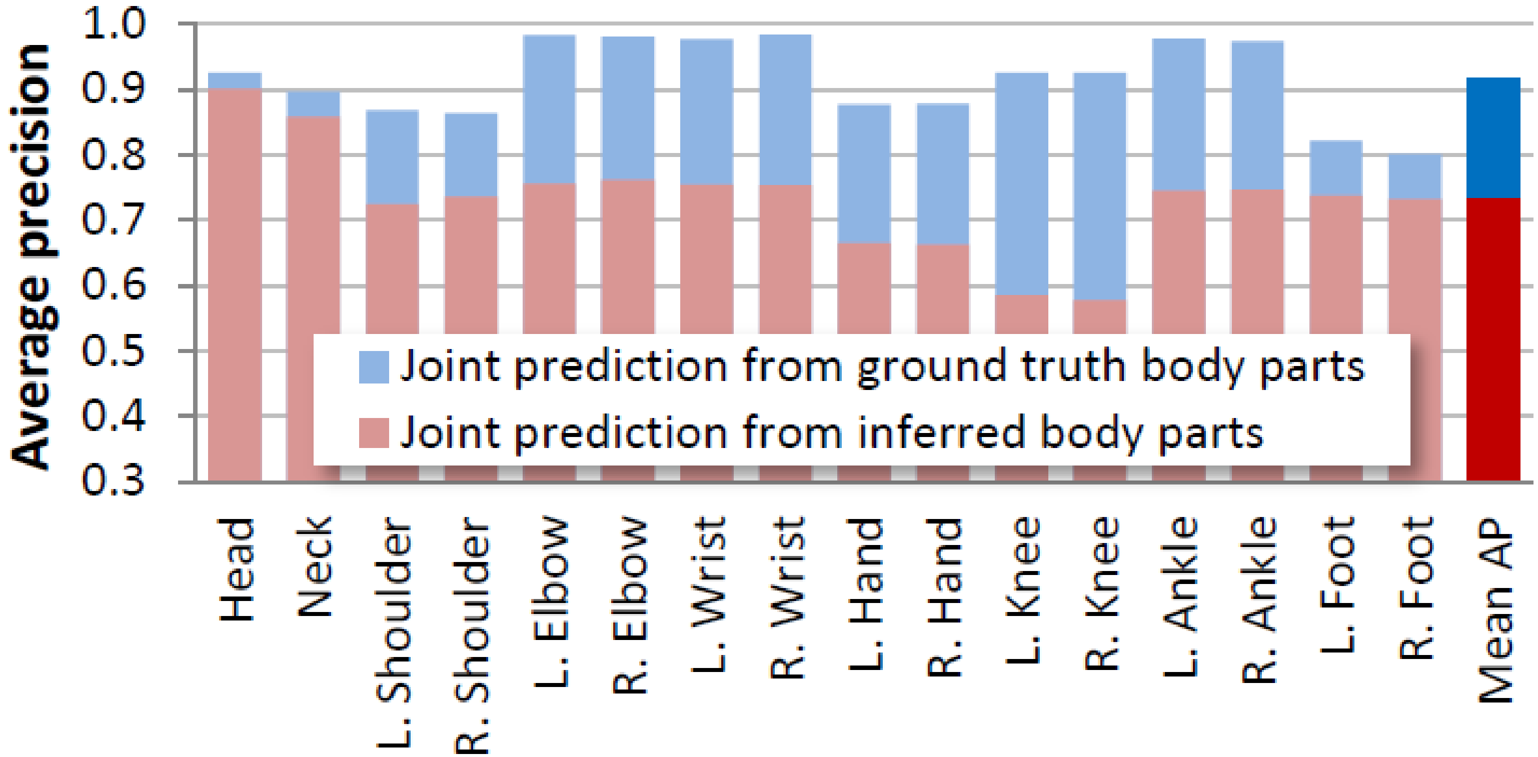
Offset Size



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

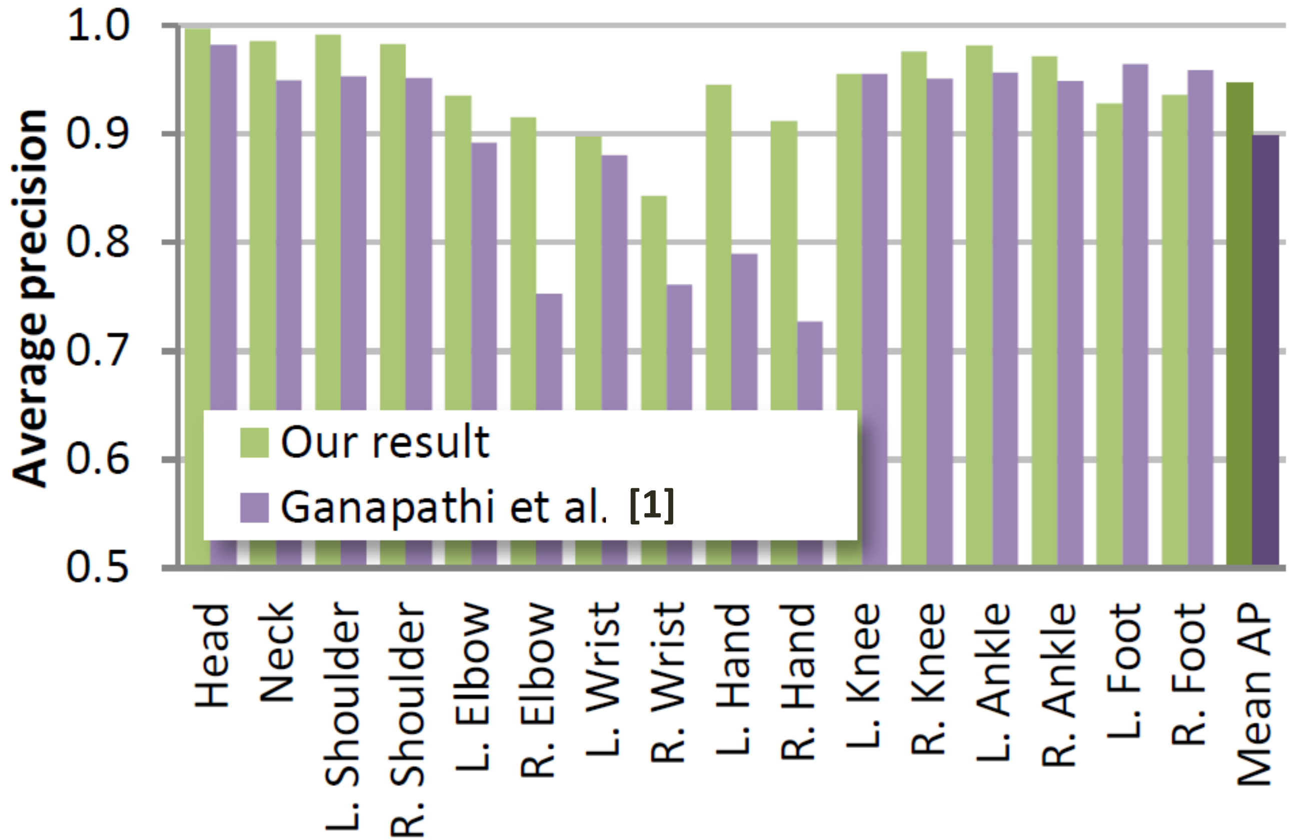
Results

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



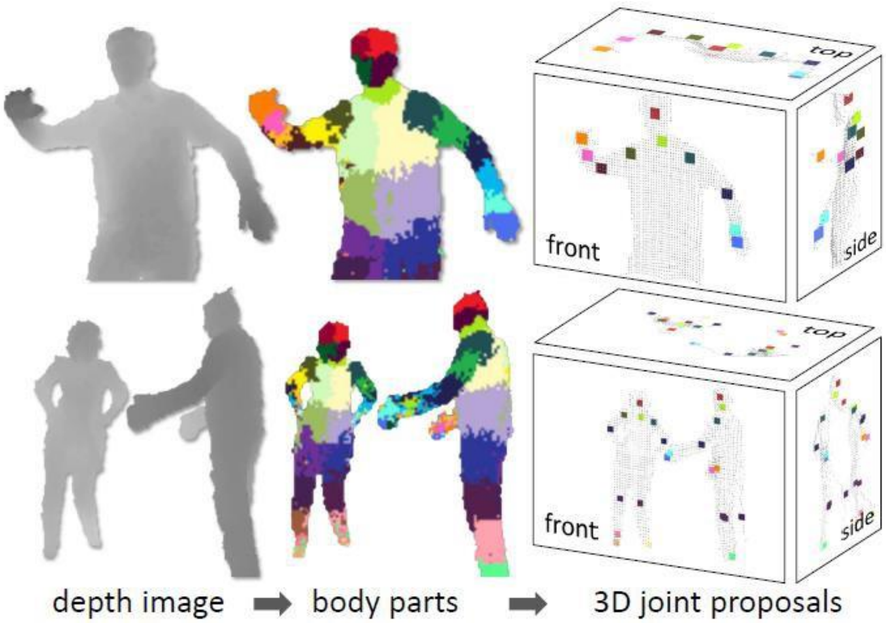
Results

- **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**

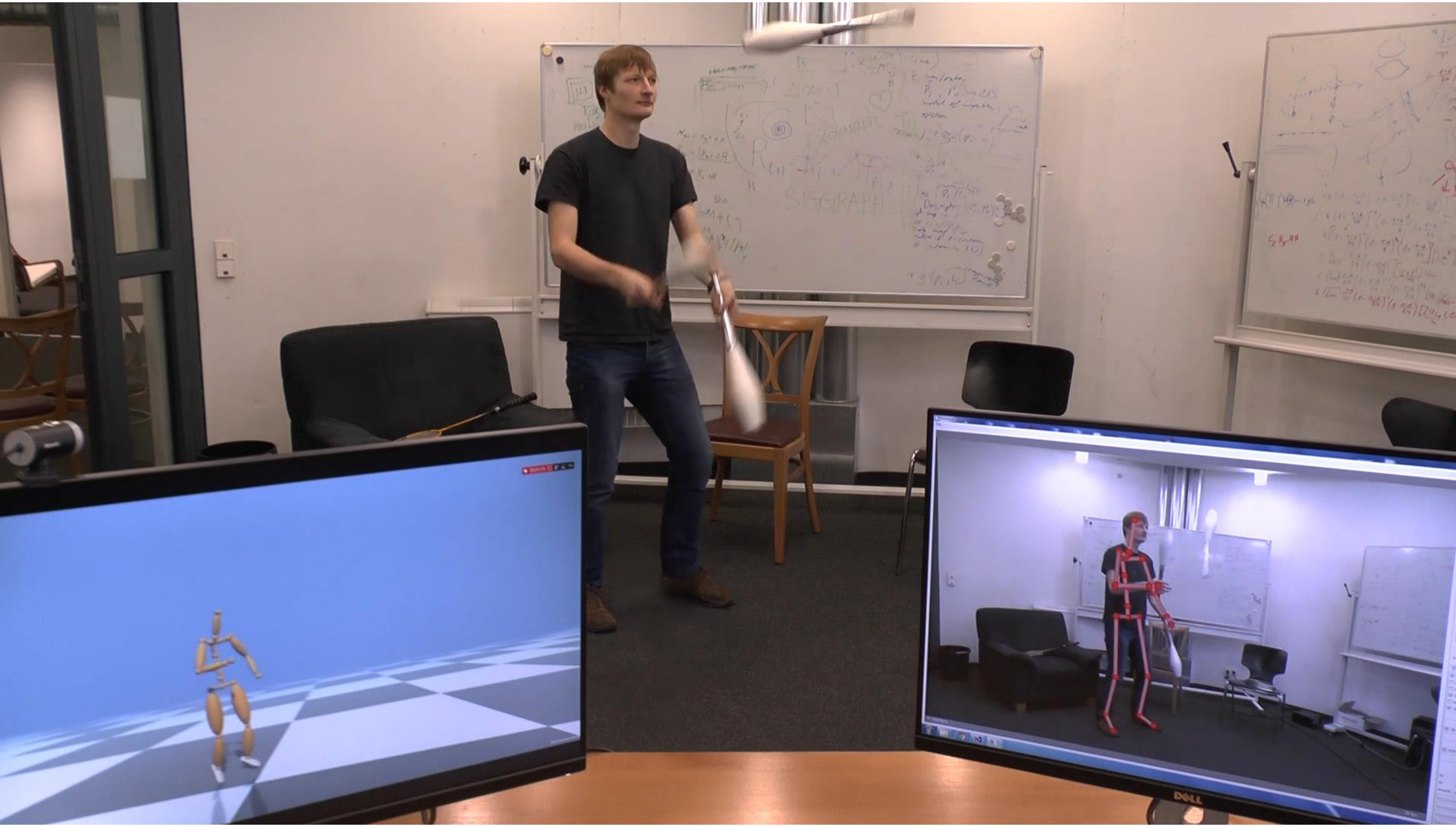


VNect – Mehta et al.

Depth information is rich…

…but do we always need it?

Can we learn to predict joint locations from RGB data?


SIGGRAPH

Pipeline

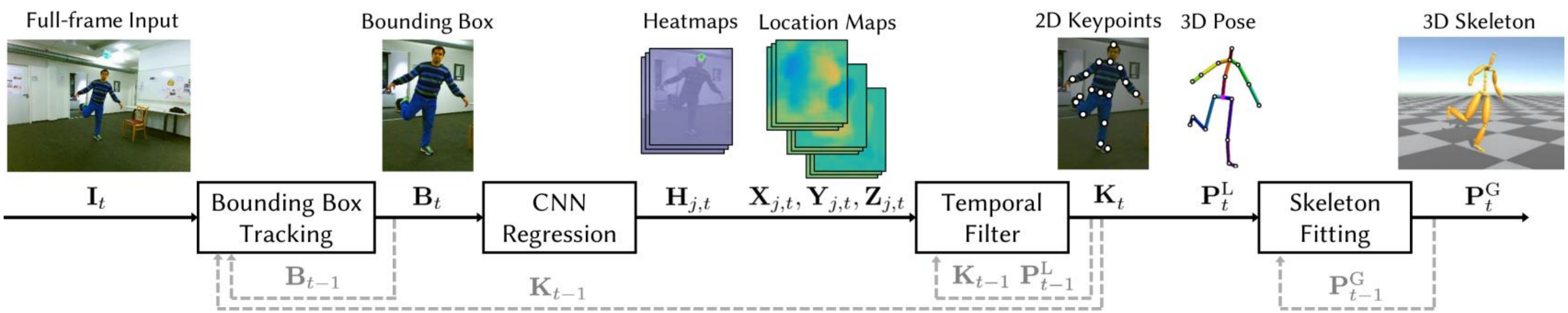


Fig. 2. Overview. Given a full-size image $\mathbf{I}_t$ at frame t, the person-centered crop $\mathbf{B}_t$ is efficiently extracted by bounding box tracking, using the previous frame's keypoints $\mathbf{K}_{t-1}$. From the crop, the CNN jointly predicts 2D heatmaps $\mathbf{H}_{j,t}$ and our novel 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

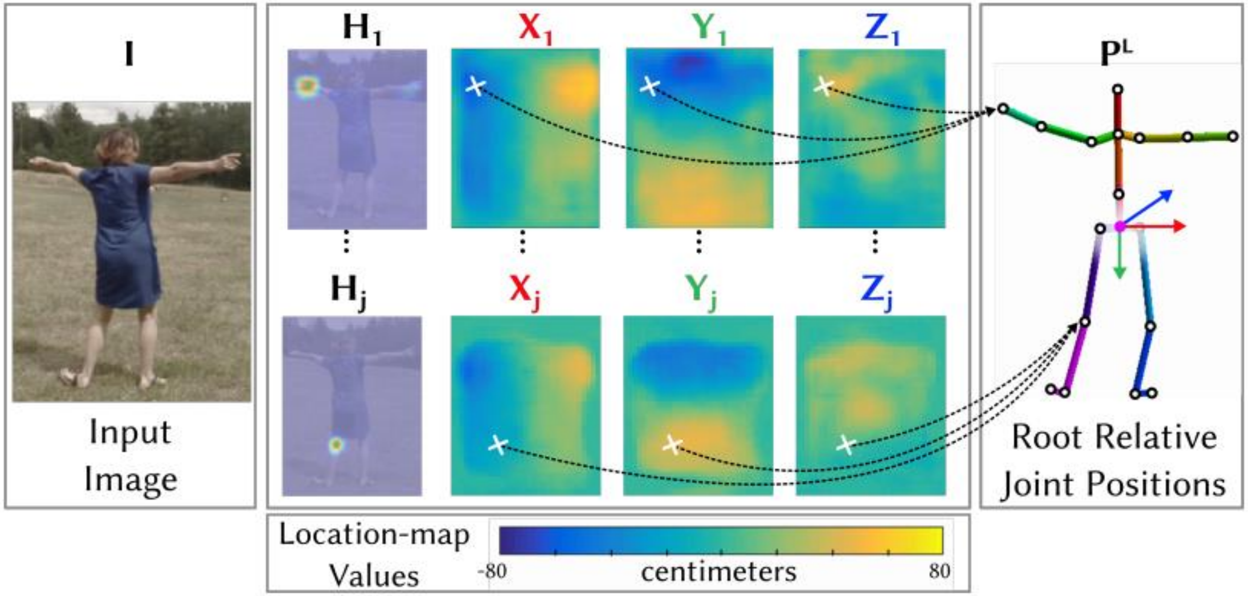


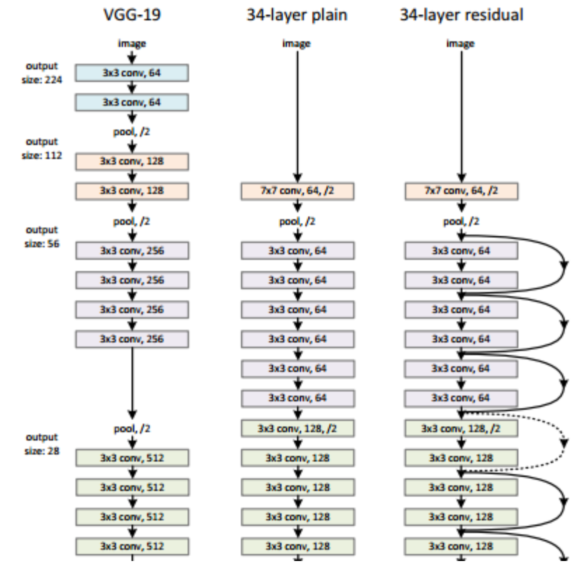
Fig. 3. Schema of the fully-convolutional formulation for predicting root relative joint locations. For each joint j, the 3D coordinates are predicted from their respective *location-maps* X_j, Y_j, Z_j at the position of the maximum in the corresponding 2D heatmap $\mathbf{H}_j$. The structure observed 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



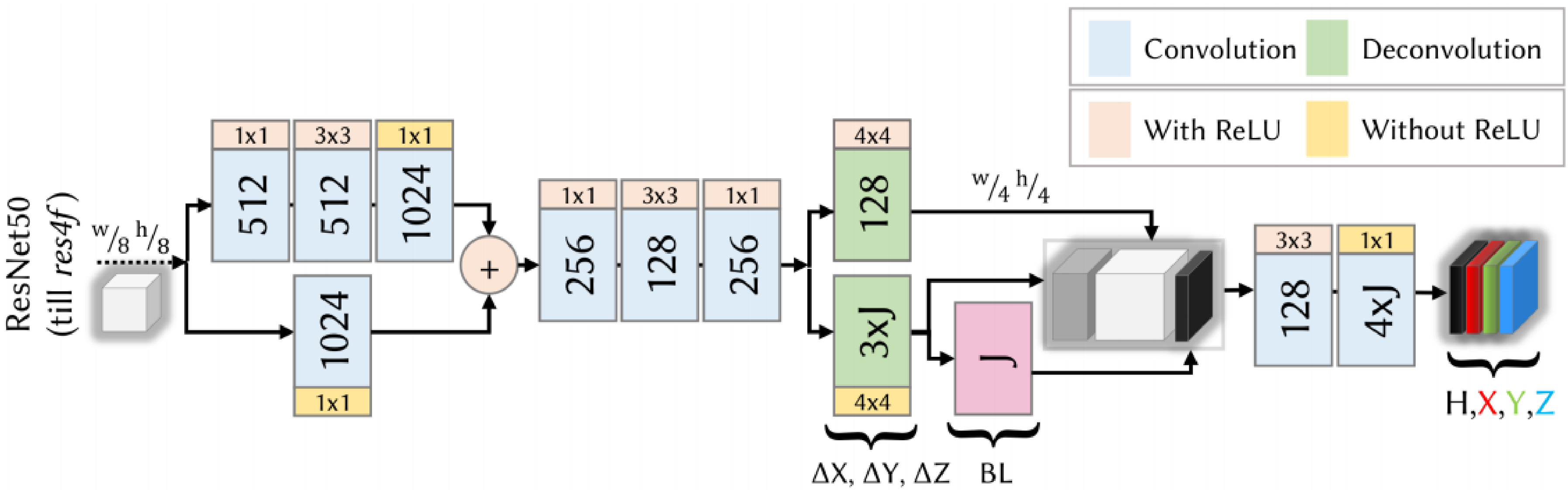
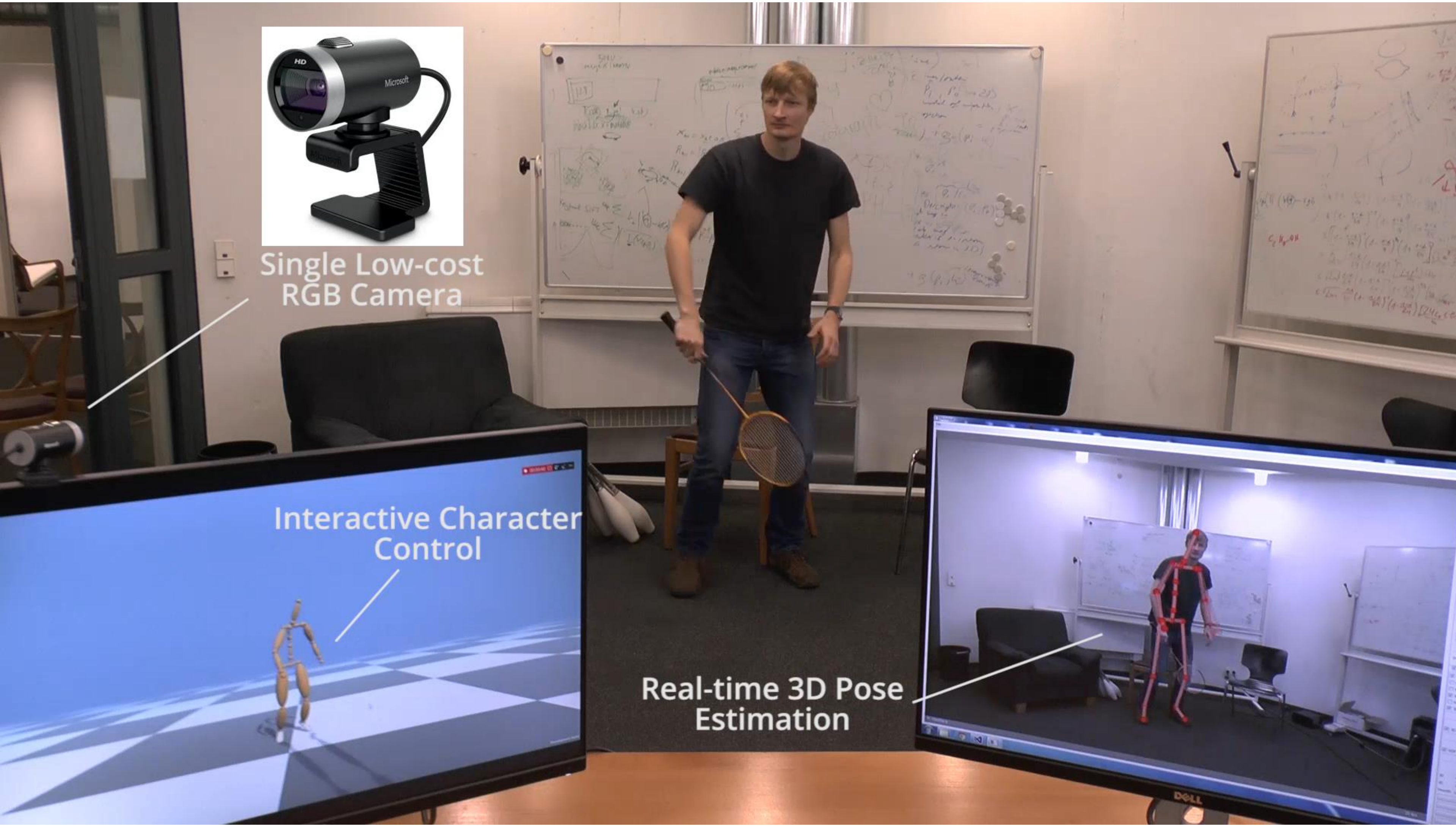


Fig. 5. Network Structure. The structure above is preceded by ResNet50/100 till level 4. We use kinematic parent relative 3D joint location predictions ΔX, ΔY, ΔZ as well as bone length maps **BL** constructed from these as auxiliary tasks.The network predicts 2D location heatmaps **H** and root relative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



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