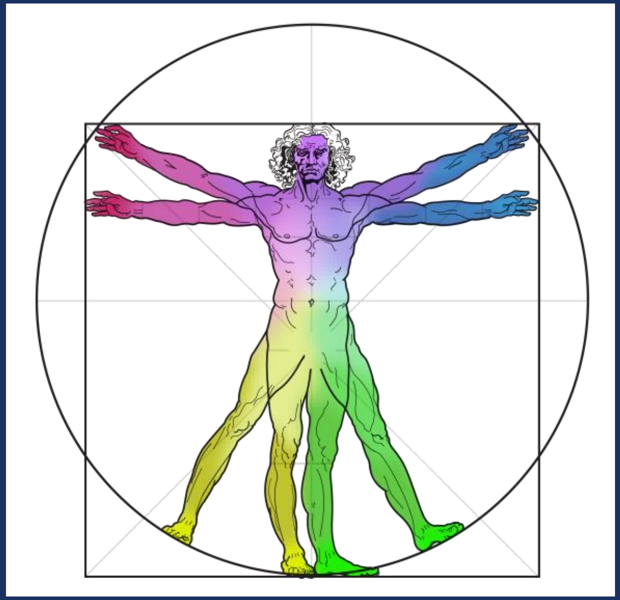
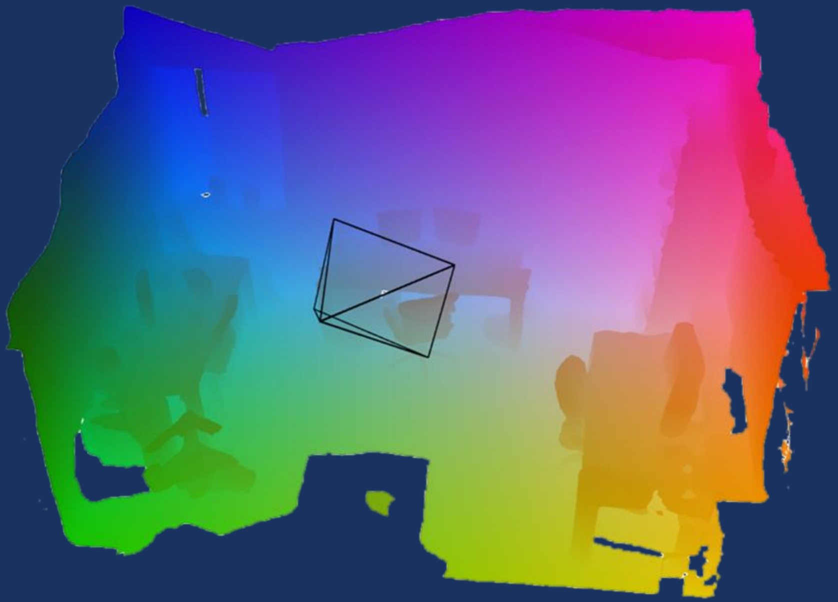
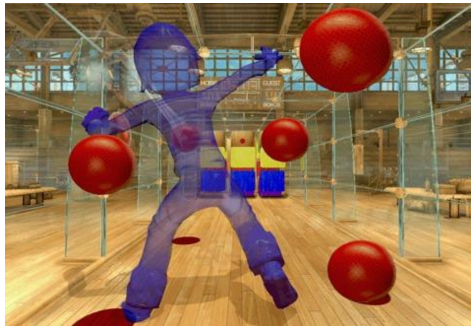
Microsoft Research

# Depth, You, and the World

JAMIE SHOTTON



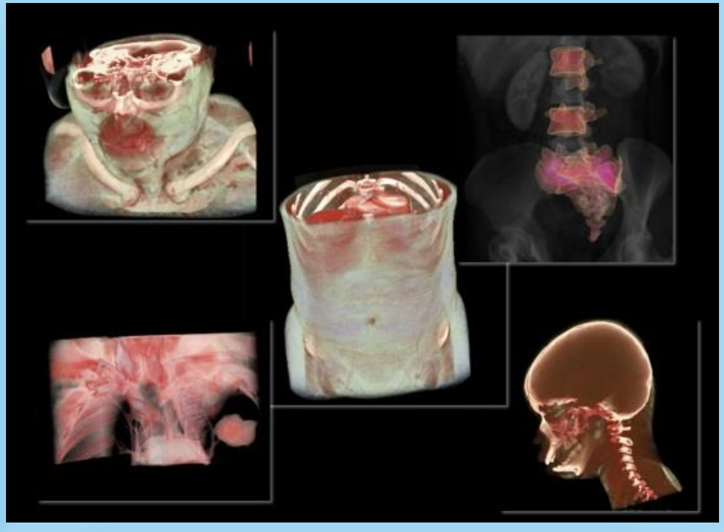



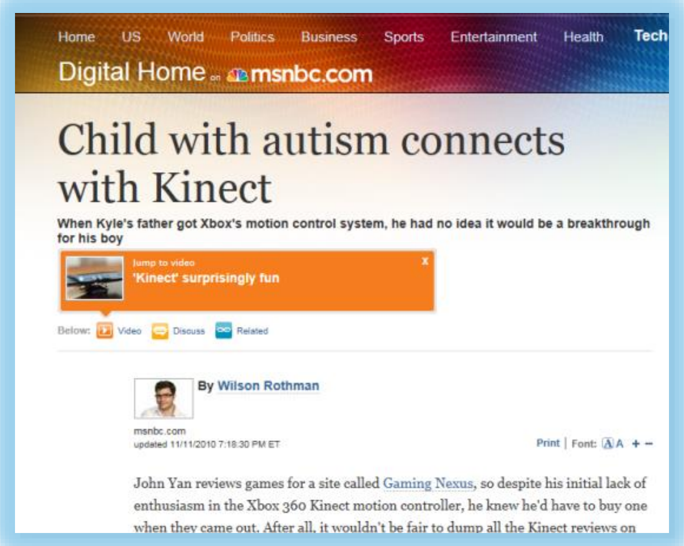
Kinect Adventures

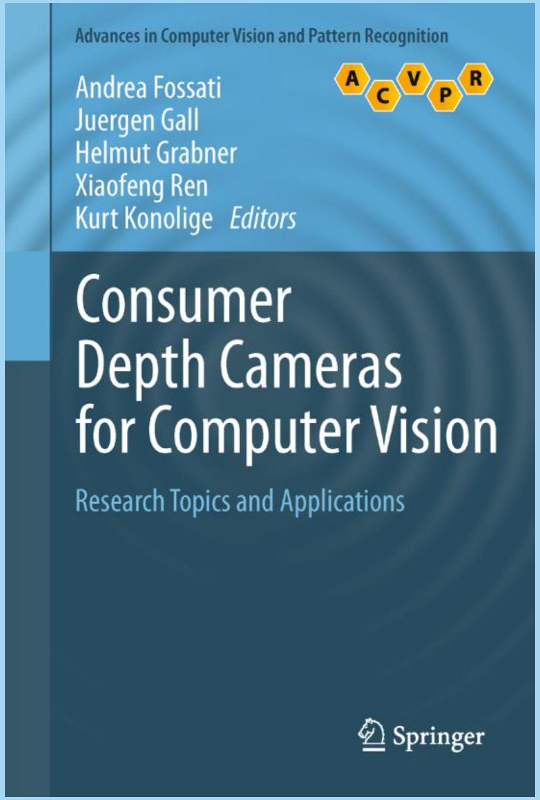
- Depth sensing camera
- Tracks 20 body joints in real time
- Recognises your face and voice


CODING4FUN
Subscribe to this blog
KINECTHACKS.com
KINECTHACKS
HOME
TIP US
Smile for the camera!
It's estimated that 3.5 trillion photos have been taken since the first pinhole camera. Even if you account for population growth the progression of photography is mind boggling. Did you know that we take 4 times as many photos compared to 10 years ago and Facebook currently hosts over 140 billion of them, with an additional [...]
Continue Reading →
Iceland Airwaves Interactive Music Installation
Siminn and the media arts company Wonwei collaborated on an interactive installation at the Reykjavik Art Museum during the 2011 Iceland Airwaves Music Festival. The installation integrates music, sensor technology and generative graphics to create a whole new experience for the festival attendees.
Few days before the start of the festival, Siminn and Wonwei gathered [...]
Continue Reading →

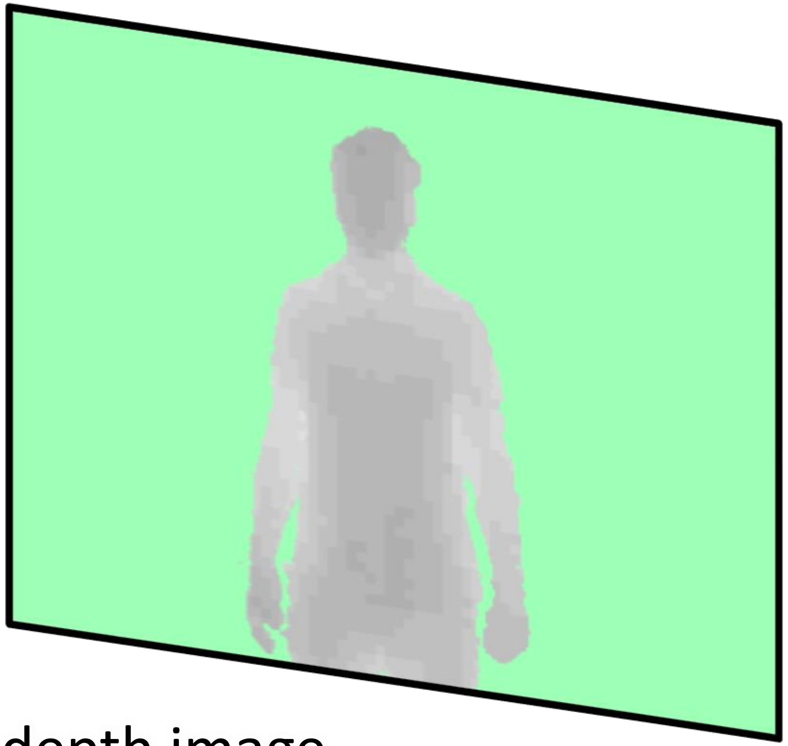

BBC
News
Sport
Weather
Travel
Future
NEWS HEALTH
Home UK Africa Asia Europe Latin America Mid-East US & Canada Business Health Sci/Environment
31 May 2012 Last updated at 06:39 GMT
Share
Trial of "touchless" gaming technology in surgery
By Adam Brimelow
Health Correspondent, BBC News
Doctors in London are trialling "touchless" technology, often used in TV games, to help them carry out delicate keyhole surgery.
The system allows them to manipulate images with their voice and hand-gestures rather than using a keyboard and mouse.
Surgeons say it gives them more control and avoids disruption.
The technology could be a valuable aid to sugery




Home US World Politics Business Sports Entertainment Health Tech
Digital Home on msnbc.com
Child with autism connects with Kinect
When Kyle's father got Xbox's motion control system, he had no idea it would be a breakthrough for his boy
Jump to video
'Kinect' surprisingly fun
By Wilson Rothman
msnbc.com
updated 11/11/2010 7:18:30 PM ET
John Yan reviews games for a site called Gaming Nexus, so despite his initial lack of enthusiasm in the Xbox 360 Kinect motion controller, he knew he'd have to buy one when they came out. After all, it wouldn't be fair to dump all the Kinect reviews on

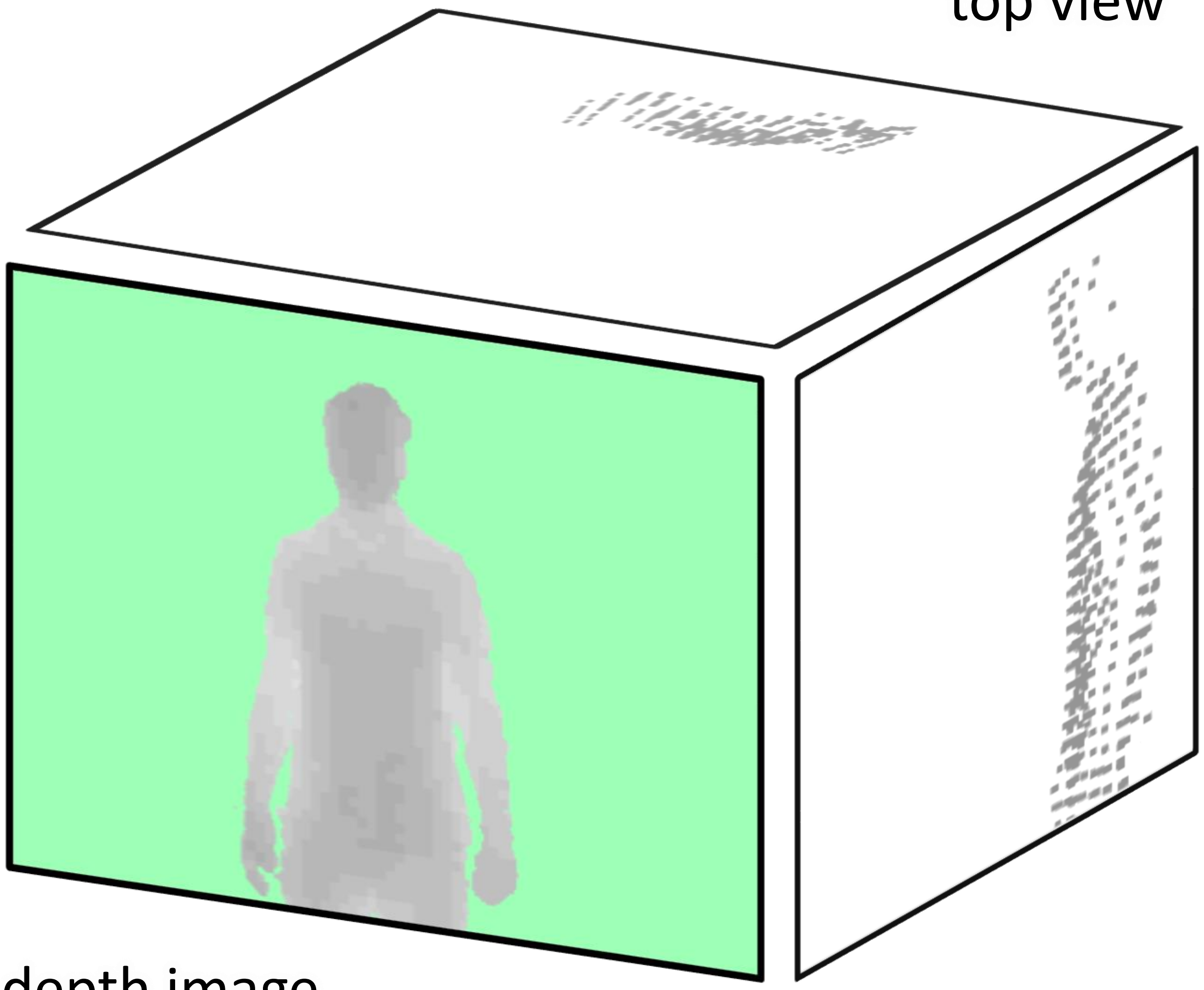

Advances in Computer Vision and Pattern Recognition
Andrea Fossati
Juergen Gall
Helmut Grabner
Xiaofeng Ren
Kurt Konolige Editors
ACVPR
Consumer Depth Cameras for Computer Vision
Research Topics and Applications
Springer

# What the Kinect Sees

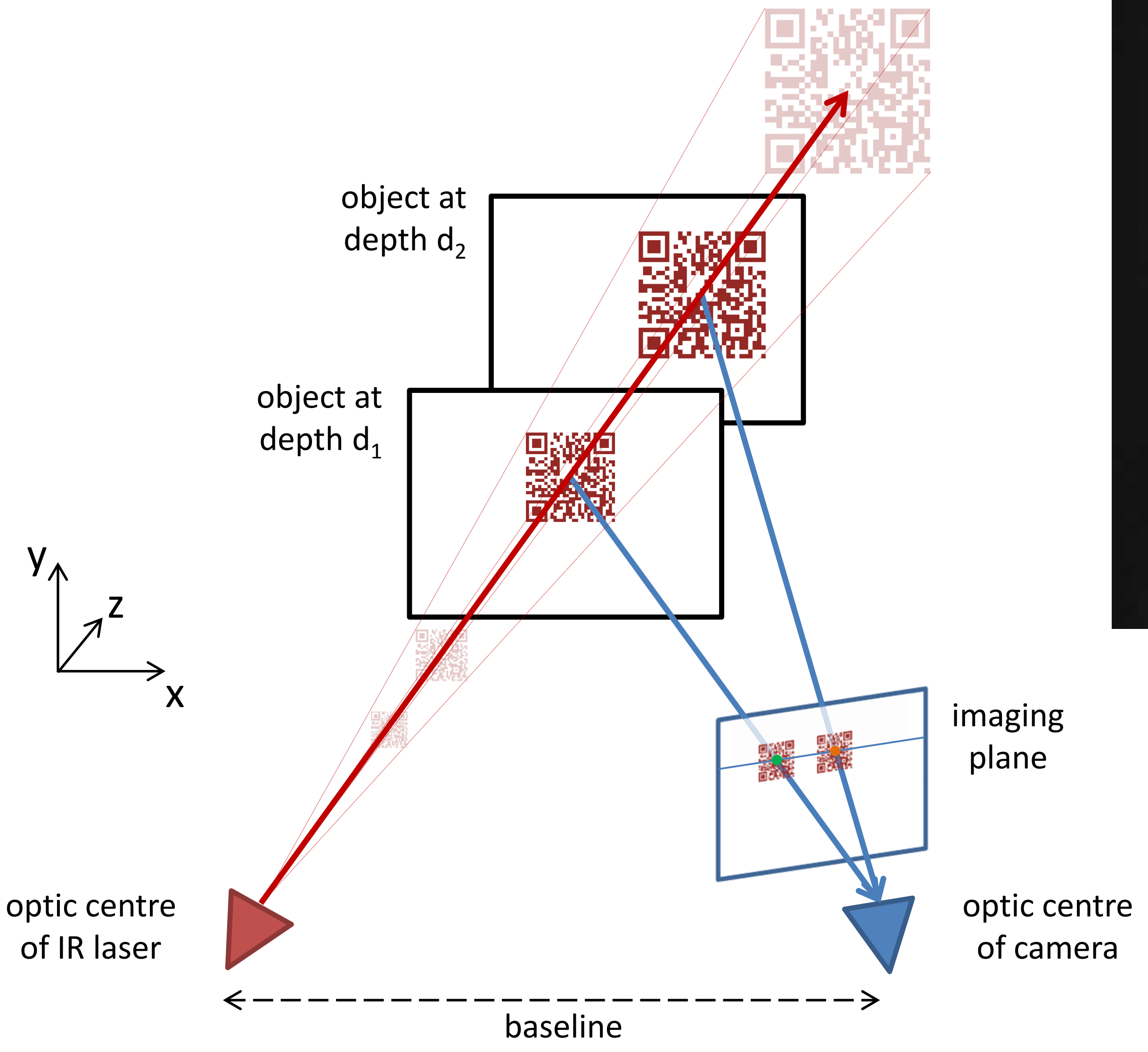


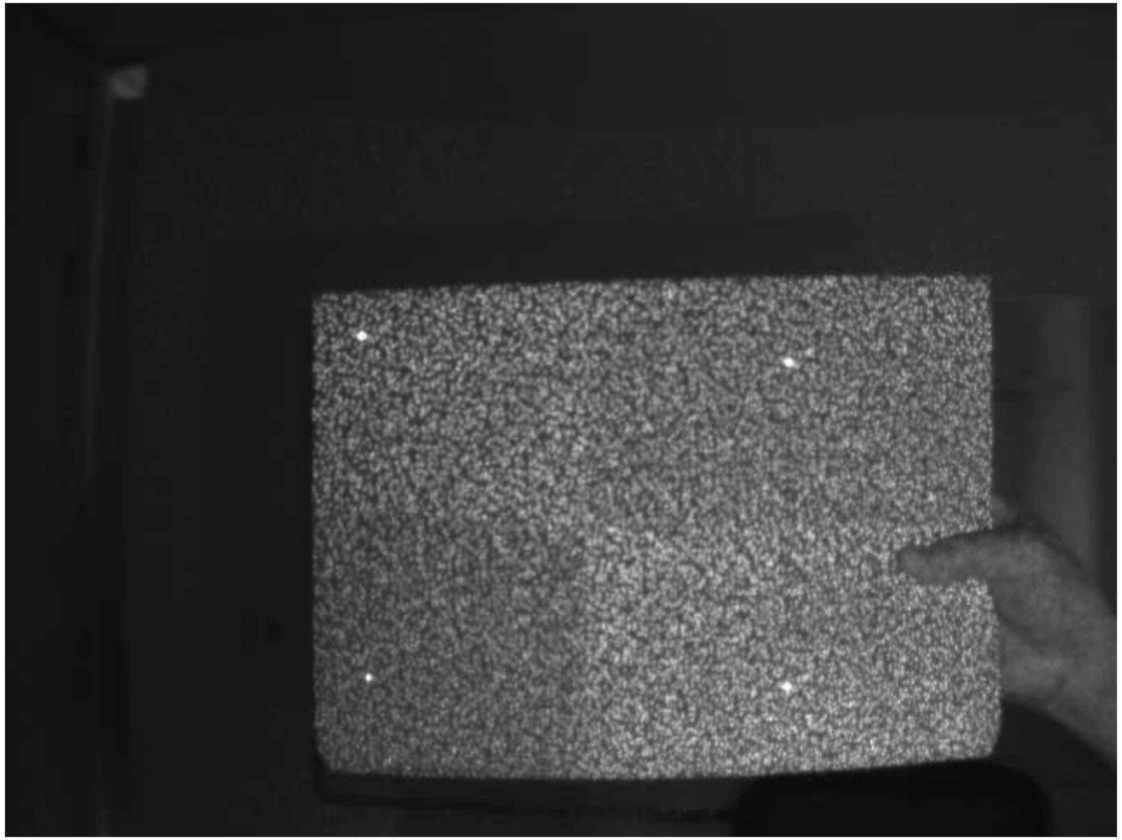
depth image
(camera view)

# What the Kinect Sees



# Structured light



# Depth Makes Vision That Little Bit Easier


## RGB

☒ Only works well lit

☒ Background clutter

☒ Scale unknown

☒ Color and texture variation

## Depth

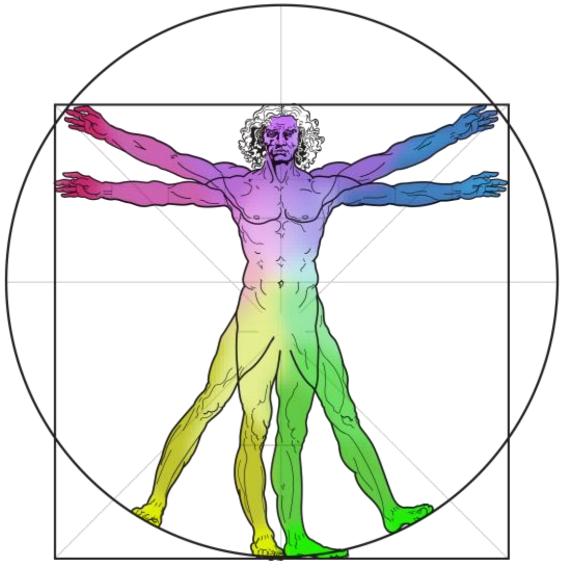
☑ Works in low light

☑ Background removal easier

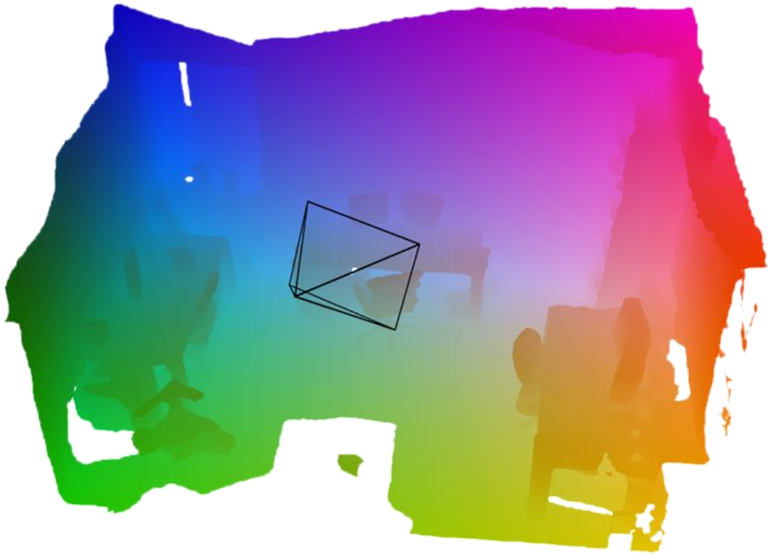
☑ Scale known

☑ Uniform texture

# ROADMAP



THE VITRUVIAN MANIFOLD
[CVPR 2012]

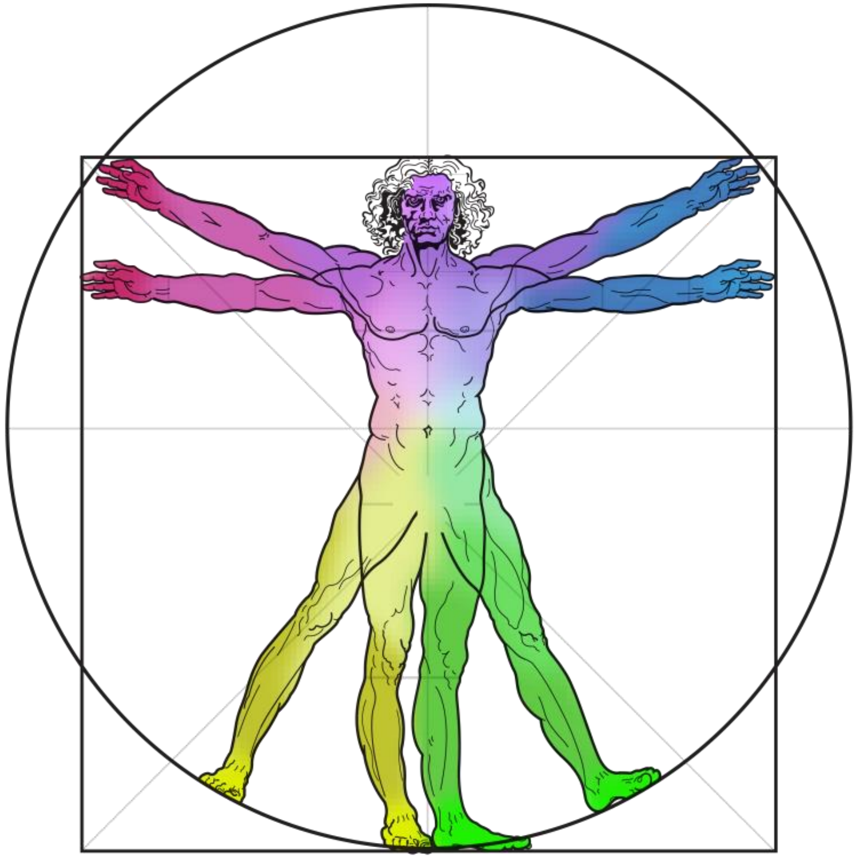


SCENE COORDINATE REGRESSION
[CVPR 2013]

Unifying principal:

**Per-pixel regression drives per-image model fitting**

# The Vitruvian Manifold





Jonathan Taylor


Jamie Shotton


Toby Sharp


Andrew Fitzgibbon


CVPR 2012

# Human Pose Estimation

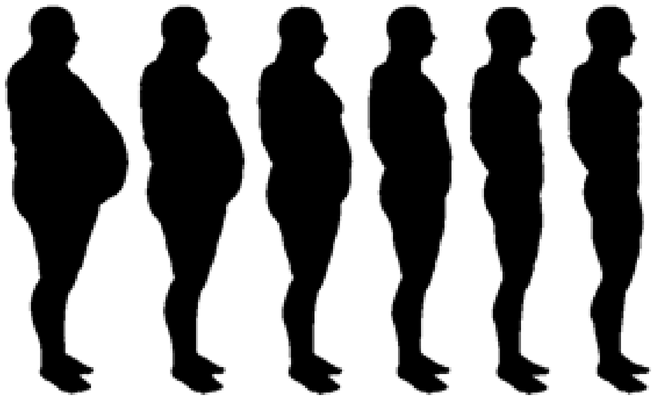
Given some image input, recover the 3D human pose:

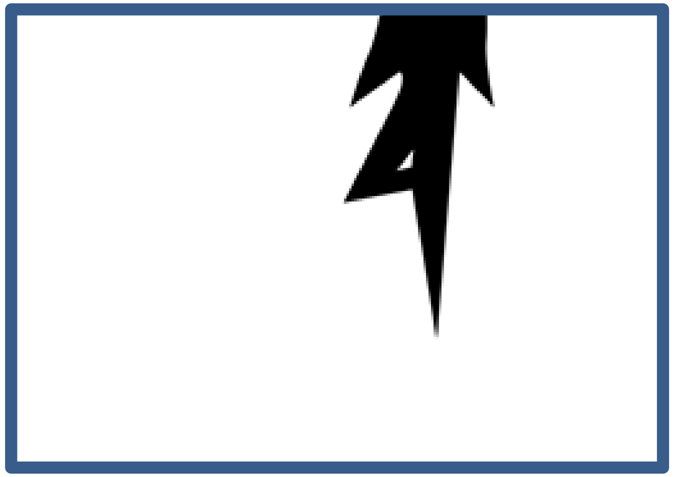
Joint positions and angles

In this work:

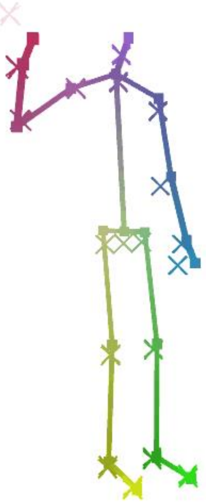
- Single frame at a time (no tracking)
- Kinect depth image as input (background removed)

# Why is Pose Estimation Hard?





# A Few Approaches



Regress directly to pose?

e.g. [Gavrila ’00] [Agarwal & Triggs ’04]

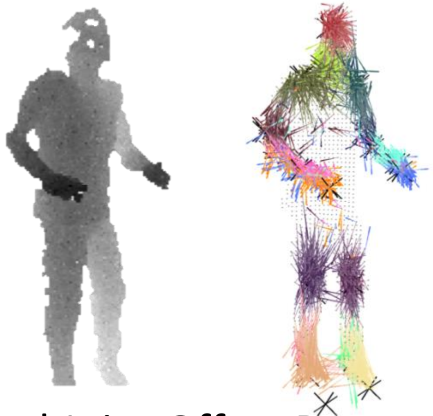
Detect and assemble parts?

e.g. [Felzenszwalb & Huttenlocher ’00] [Ramanan & Forsyth ’03] [Sigal *et al.* ’04]

Detect parts?

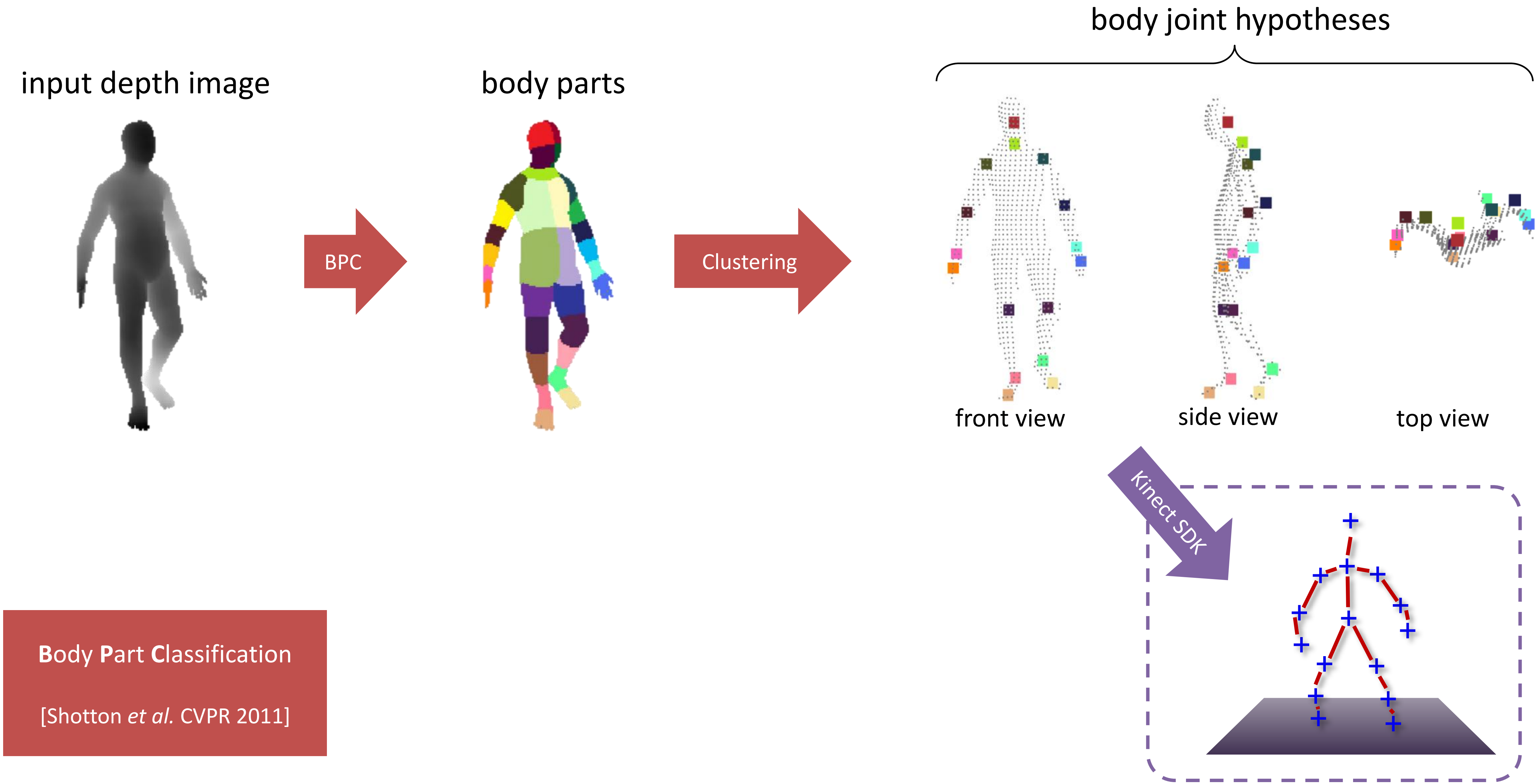
e.g. [Bourdev & Malik ‘09] [Plagemann *et al.* ‘10] [Kalogerakis *et al.* ‘10]

Per-Pixel Body Part Classification
[Shotton *et al*. ‘11]



Per-Pixel Joint Offset Regression
[Girshick *et al*. ‘11]

# Background: Learning Body Parts for Kinect




**Body Part Classification**

[Shotton *et al.* CVPR 2011]

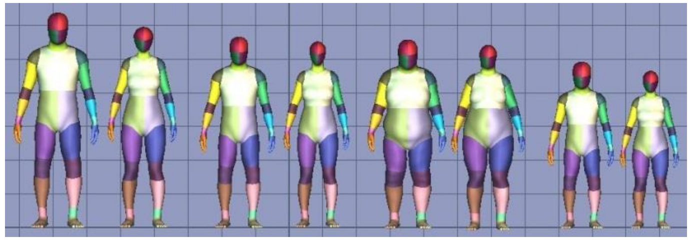
# Synthetic Training Data

[Vicon]

Record mocap

100,000s of poses

Retarget to varied body shapes



Render (depth, body parts) pairs

Train invariance to:

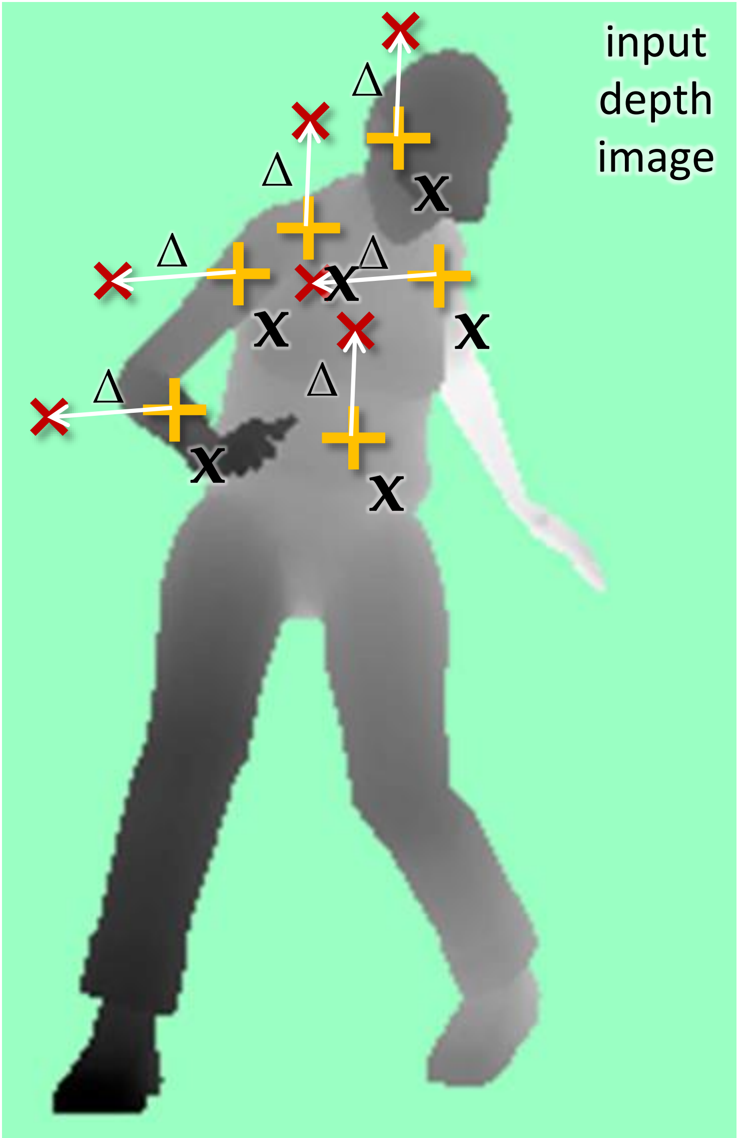
# Depth Image Features

- Depth comparisons
  - very fast to compute

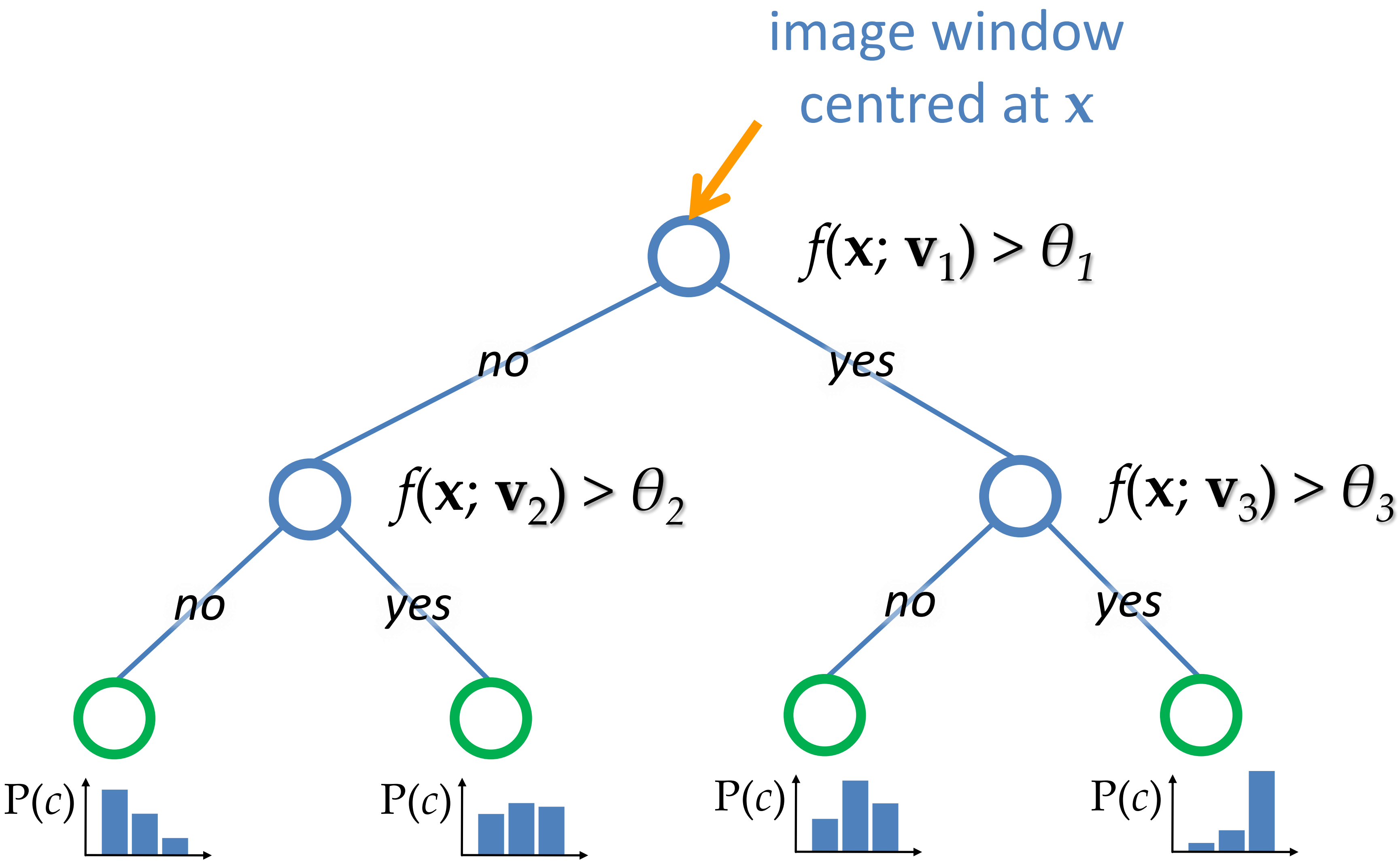
$$\underbrace{f}_{\text{feature response}}(\underbrace{\mathbf{x}}_{\text{image coordinate}};\overbrace{\mathbf{v}}^{\text{offset}}) = \overbrace{d(\mathbf{x})}^{\text{depth}} - \overbrace{d(\mathbf{x}+\Delta)}^{\text{offset depth}}$$

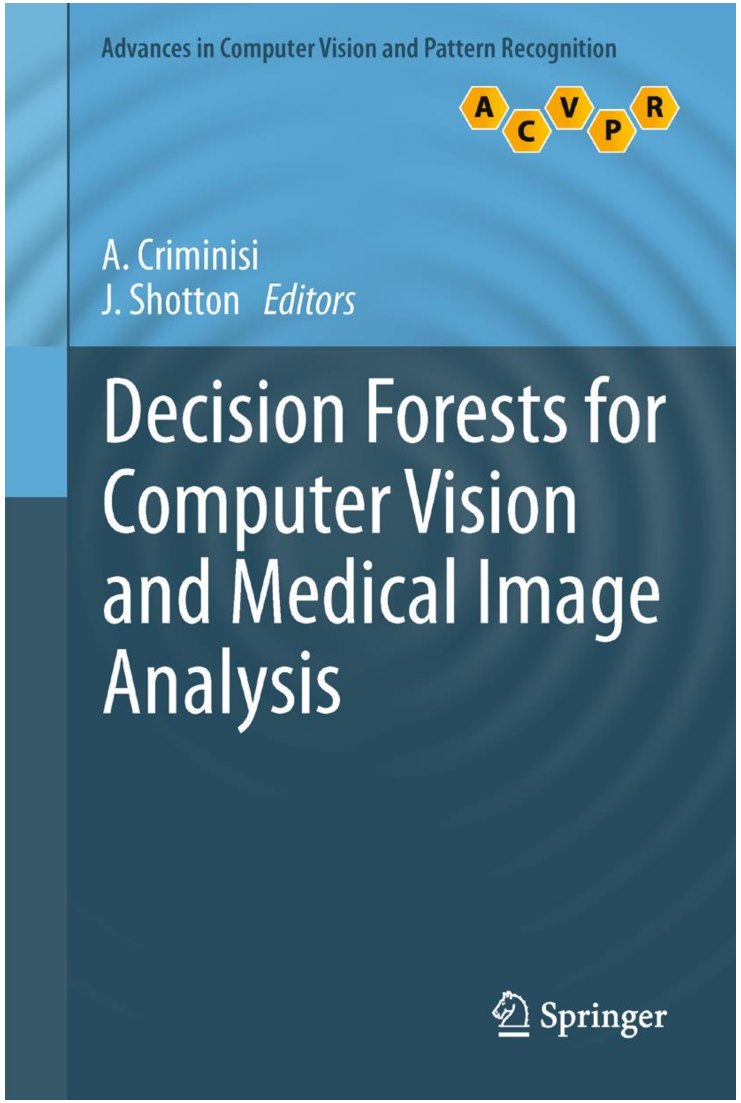
$$\Delta = \frac{\mathbf{v}}{\underbrace{d(\mathbf{x})}_{\text{scales inversely with depth}}}$$

Background pixels
$d$ = large constant



# Decision tree classification

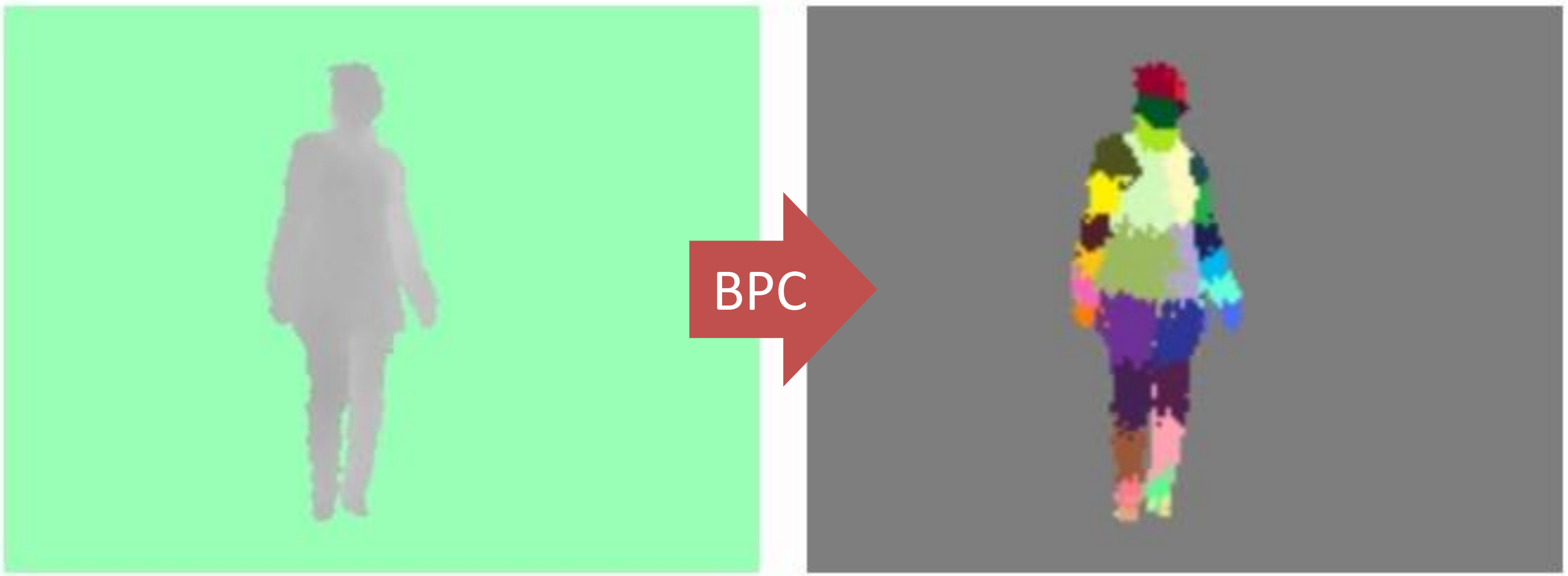





**Decision Forests Book**

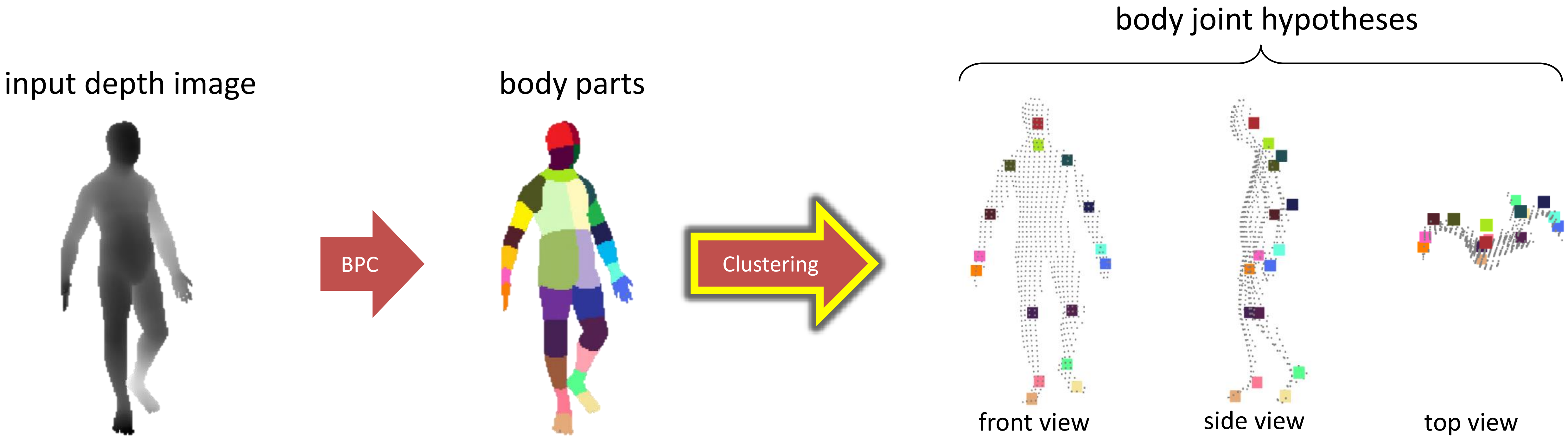
- Theory – Tutorial & Reference
- Practice – Invited Chapters
- Software and Exercises
- Tricks of the Trade

input depth inferred body parts




[Shotton *et al.* CVPR 2011]

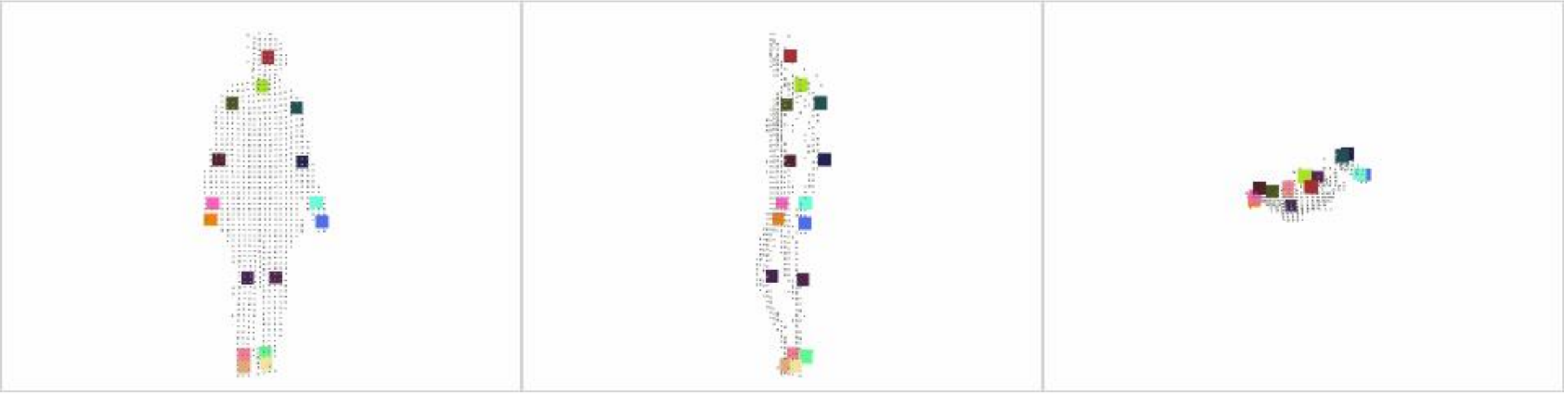
no tracking or smoothing




- Mean shift mode detection on density

input depth

inferred body parts



front view

side view

top view

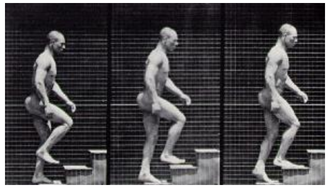
inferred joint position hypotheses

[Shotton *et al.* CVPR 2011]

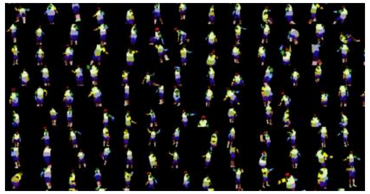
**no tracking or smoothing**

# Body Part Recognition in Kinect

KINECT

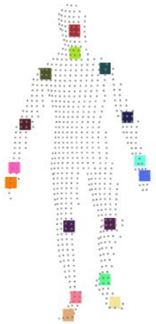


Single frame at a time –> robust

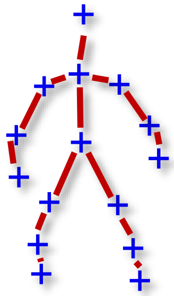


Large training corpus -> invariant

Fast, parallel implementation



✓

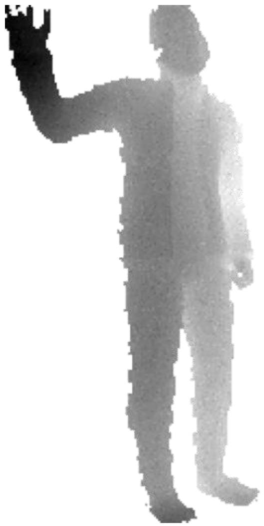


✗

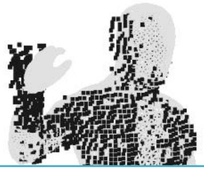
No kinematic skeleton

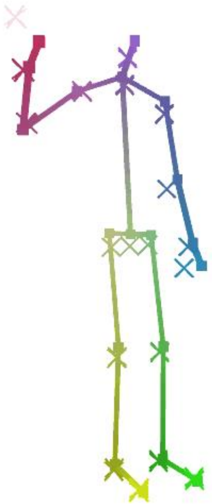
Limited handling of occlusion

# A few approaches



Explain the data directly with a mesh model
[Ballan *et al*. ‘08] [Baak *et al.* ‘11]





- GOOD: Full skeleton
- GOOD: Kinematic constraints enforced from the outset
- GOOD: Able to cope with occlusion and cropping
- BAD: Many local minima
- BAD: Highly sensitive to initial guess
- BAD: Potentially slow

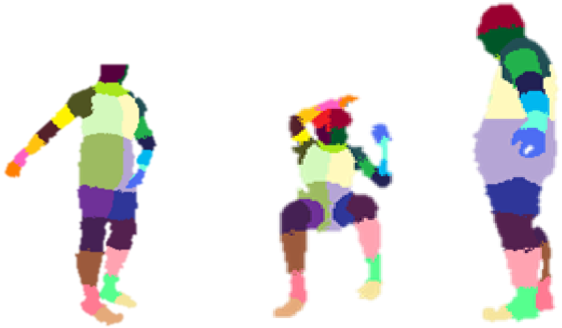
# From Body Parts to Dense Correspondences

Body Parts

increasing number of parts

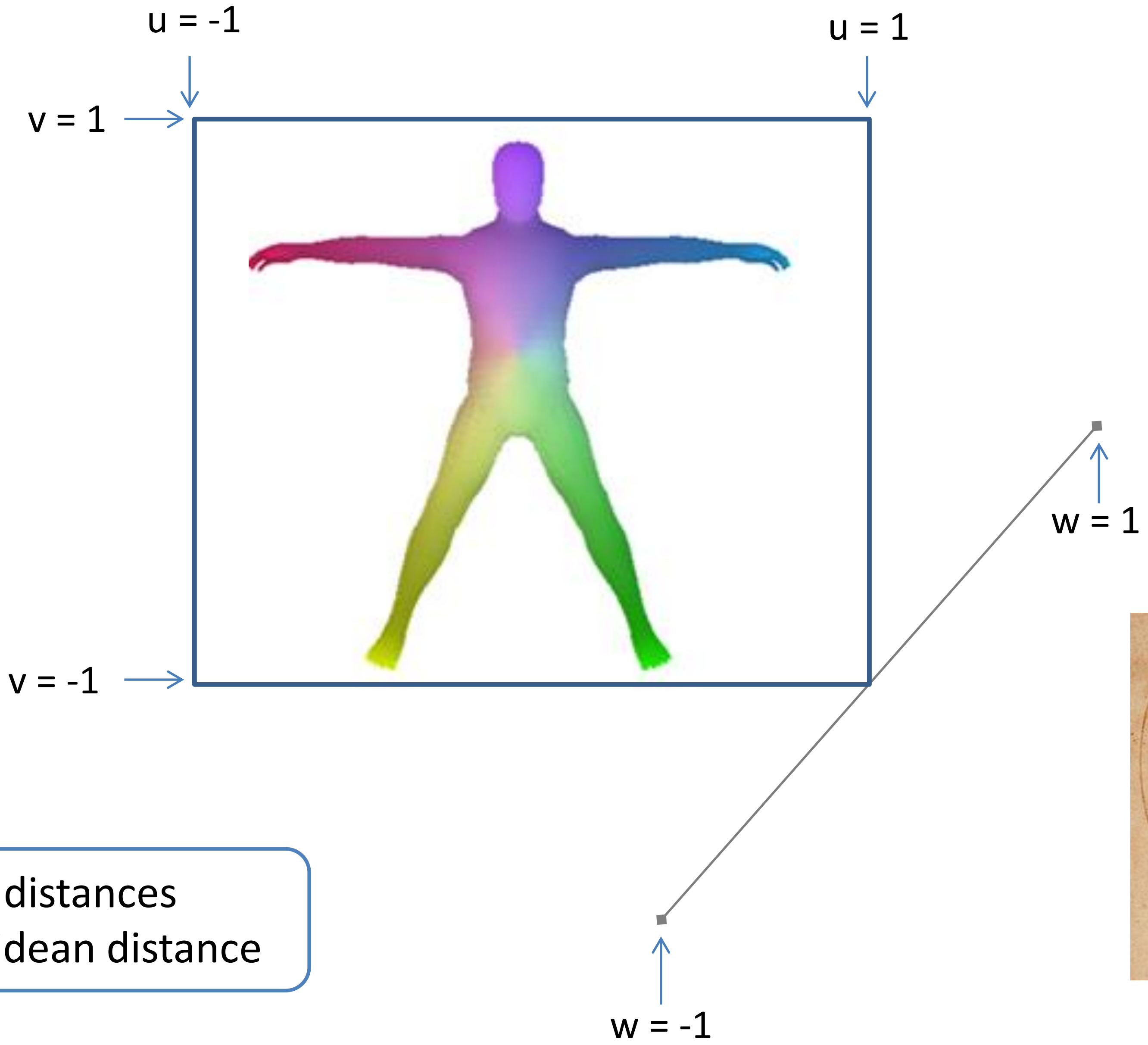
classification → regression

The “Vitruvian Manifold”

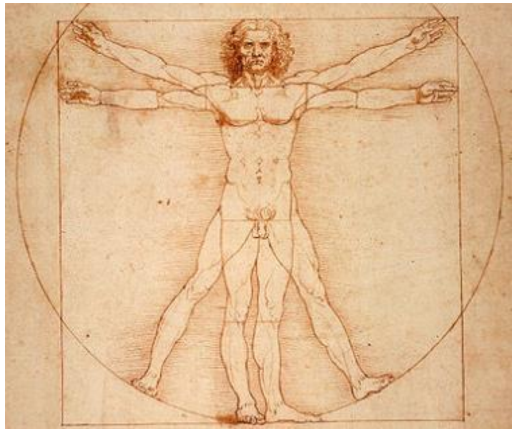


Texture is mapped across body shapes and poses

# The “Vitruvian Manifold” Embedding in 3D

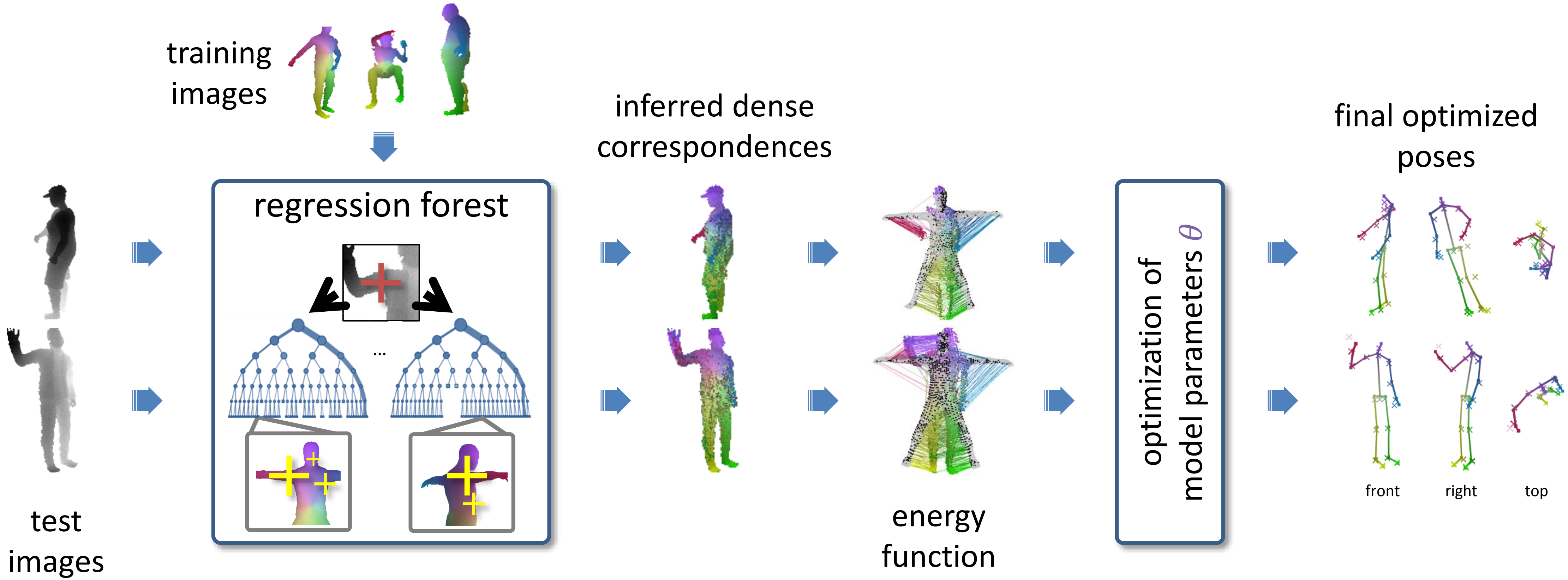



Geodesic surface distances
approximated by Euclidean distance

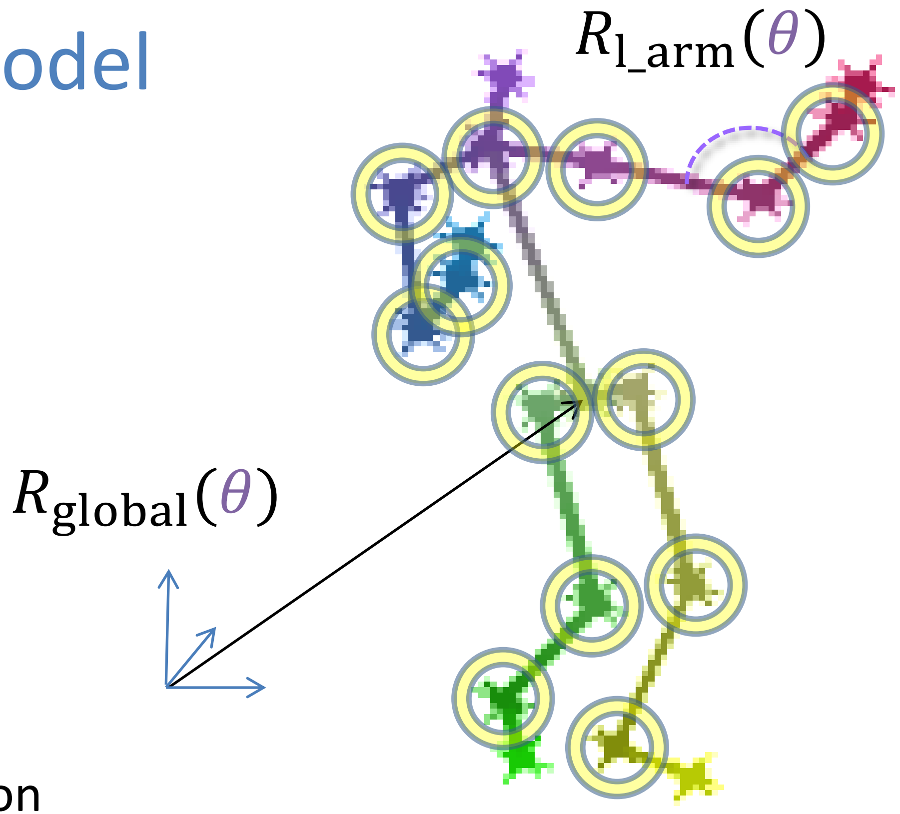


[L. Da Vinci, 1487]

# Overview


training images
regression forest
...
test images
inferred dense correspondences
energy function
optimization of model parameters $\theta$
final optimized poses
front
right
top

# Human Skeleton Model




- Mesh is attached to a hierarchical skeleton

- Each limb $l$ has a transformation matrix $T_l(\theta)$ relating its local coordinate system to the world:

$$
\boxed{\begin{aligned}
T_{\text{root}}(\theta) &= R_{\text{global}}(\theta) \\
T_l(\theta) &= T_{\text{parent}(l)}(\theta) R_l(\theta)
\end{aligned}}
$$

- $R_{\text{global}}(\theta)$ encodes a global scaling, translation and rotation

- $R_l(\theta)$ encodes a rotation and fixed translation relative to its parent

- 13 parameterized joints using quaternions to represent unconstrained rotations

- This gives $\theta$ a total of $1 + 3 + 4 + 4 * 13 = 60$ degrees of freedom

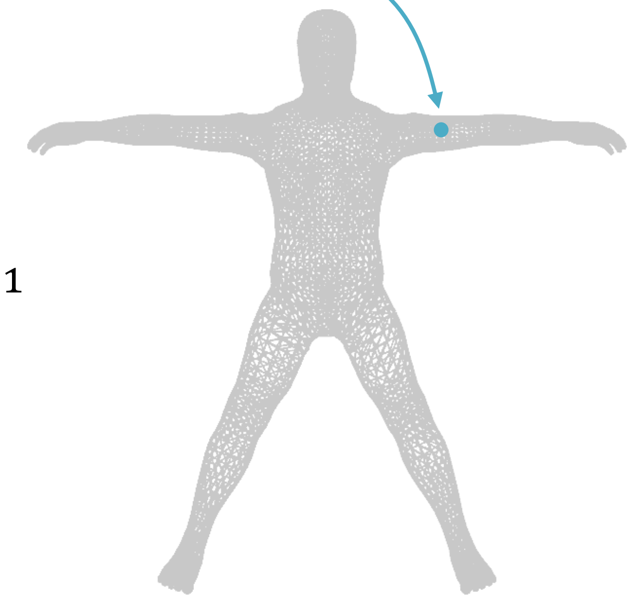
# Linear Blend Skinning

Each vertex $u$

- has position $p$ in base pose $\theta_0$
- is attached to $K$ limbs $\{l_k\}_{k=1}^K$ with weights $\{\alpha_k\}_{k=1}^K$

In a new pose $\theta$, the skinned position $u$ of is:

$$M(u;\theta) = \sum_{k=1}^{K} \underbrace{\alpha_k T_{l_k}(\theta) \underbrace{T_{l_k}^{-1}(\theta_0) p}_{\text{position in limb } l_k\text{'s coordinate system}}}_{\text{position in world coordinate system}}$$



Mesh in base pose $\theta_0$

# Test Time Model Fitting

## Optimization Strategies

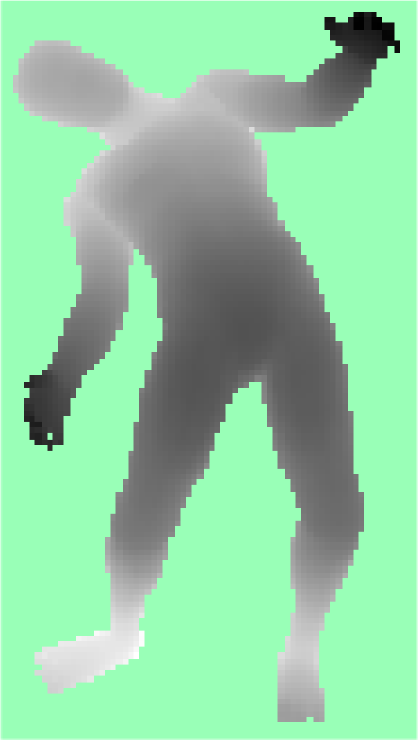
- Alternating between pose $\theta$ and correspondences $u_1, \ldots u_n$
  - ➢ Iterative Closest Point (ICP)
- Traditionally, start from initial $\theta$
  - from tracking or manual initialization
- Instead, we start from initial $u_1, \ldots u_n$
  - inferred discriminatively
- “One-shot” pose estimation
  - *can we achieve a good result without iterating?*

$\theta \quad u_1 \ldots u_n \quad \sum_i$

Note: simplified energy - more details to come

# One-Shot Pose Estimation: An Early Result

*Can we achieve a good result without iterating?*

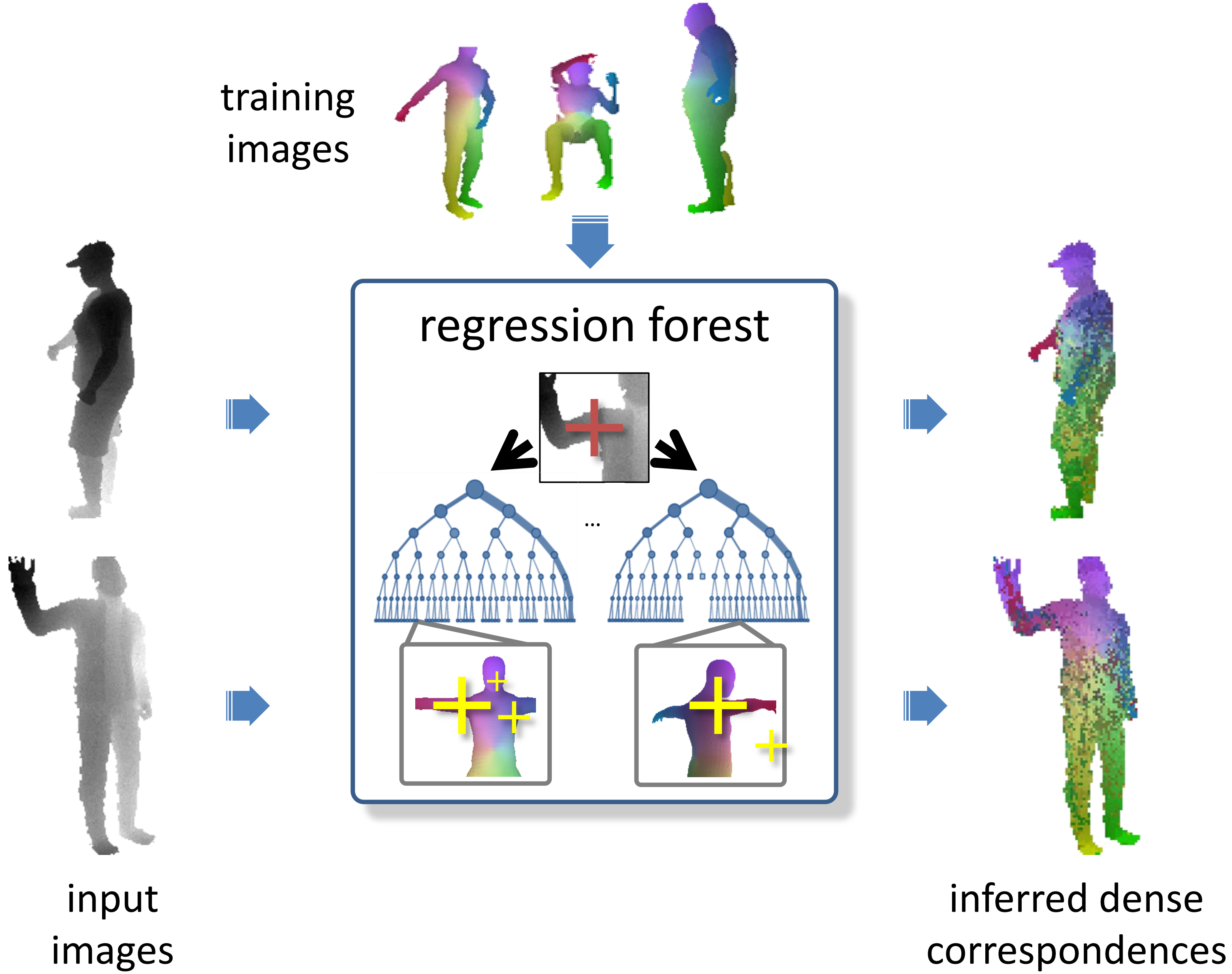


test
depth image

ground truth
correspondences
(legacy coloring scheme)

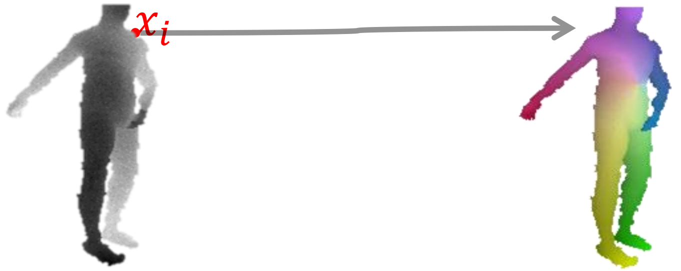
convergence
visualization

# Discriminative Model: Predicting Correspondences

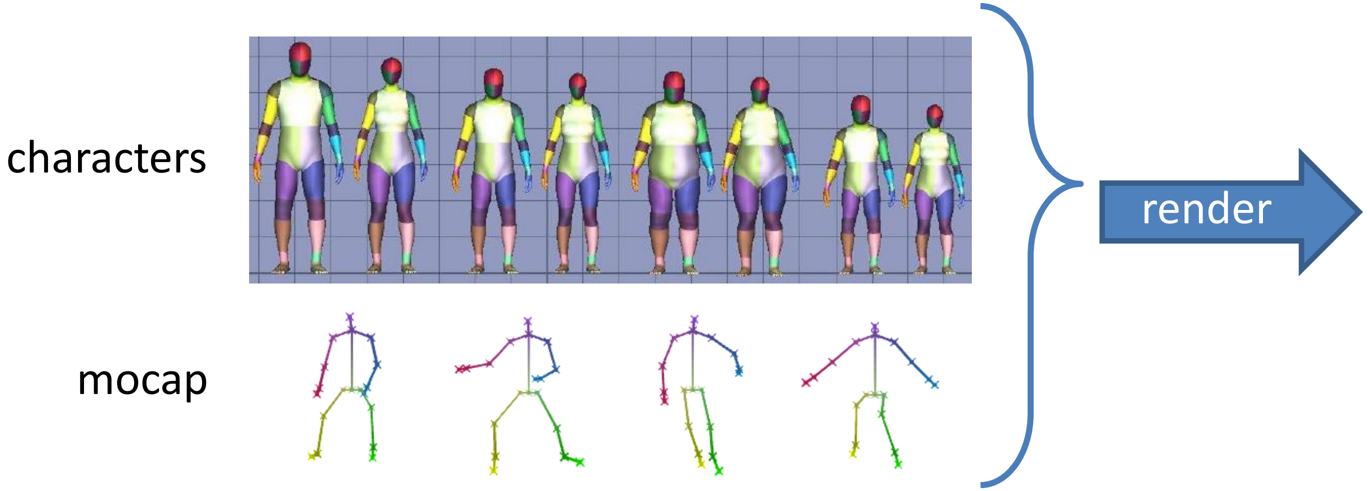


# Learning the Correspondences

- How to learn the mapping from depth pixels to correspondences?

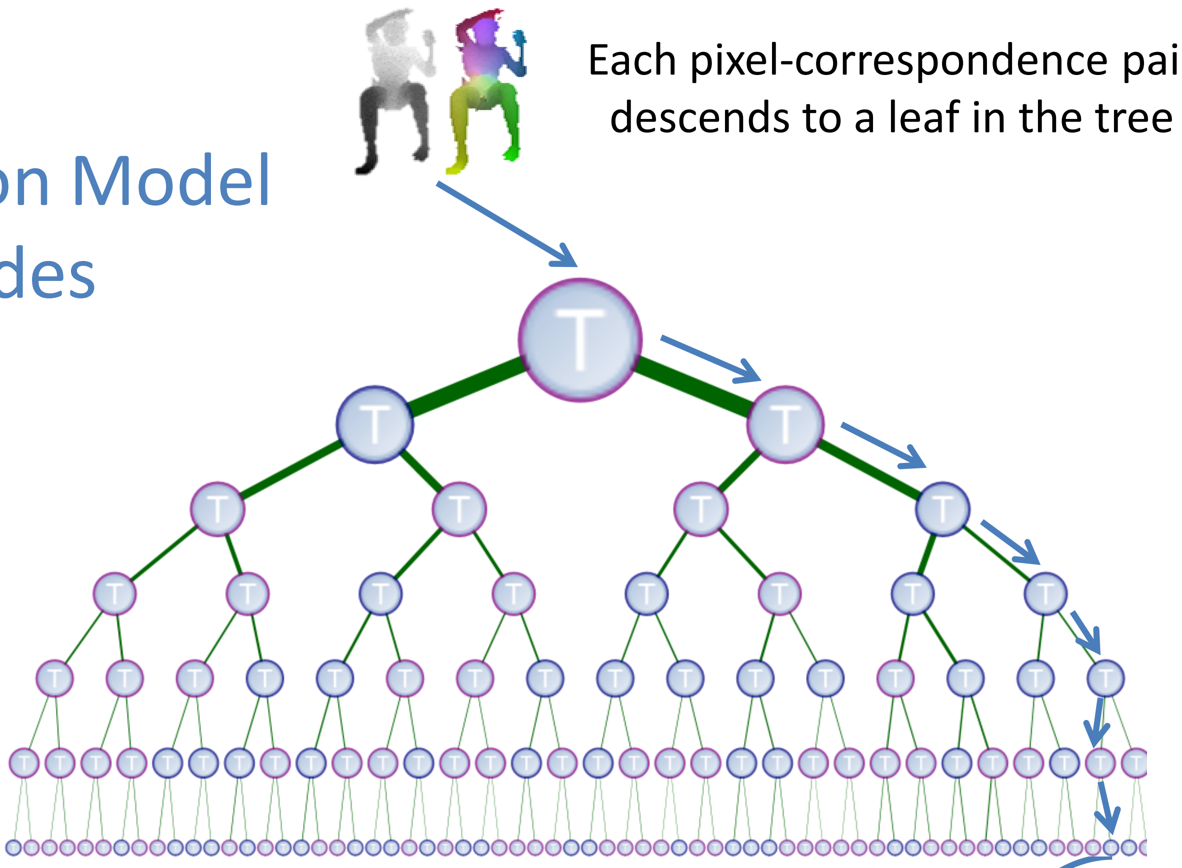


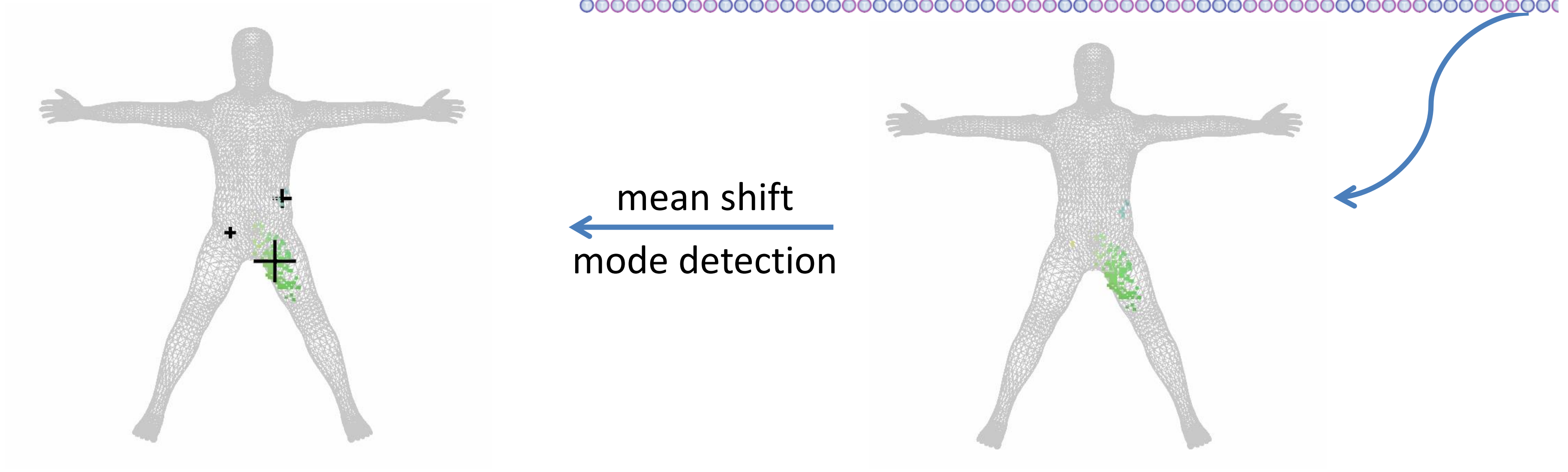

- Render synthetic training set:



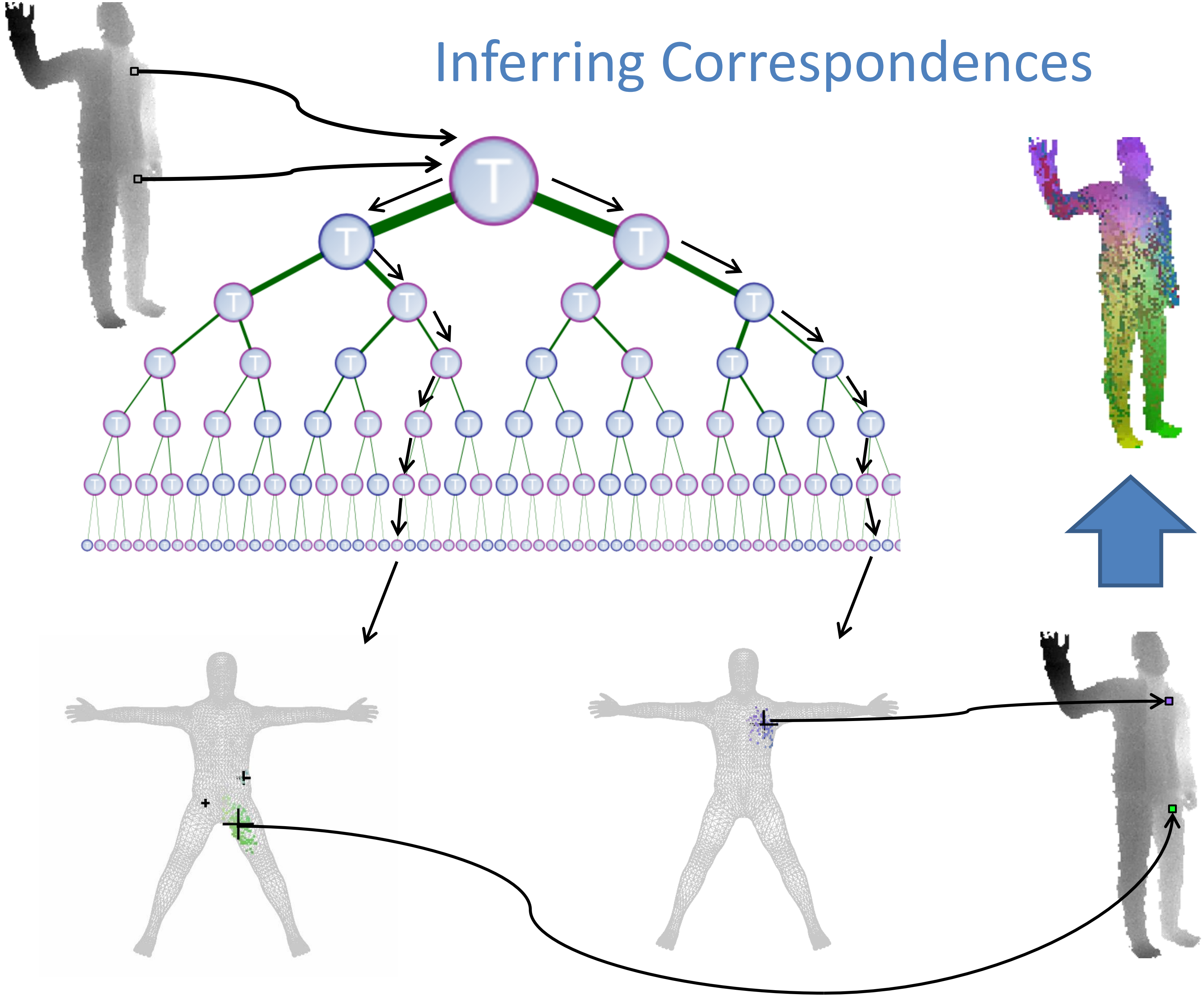

- Train regression forest

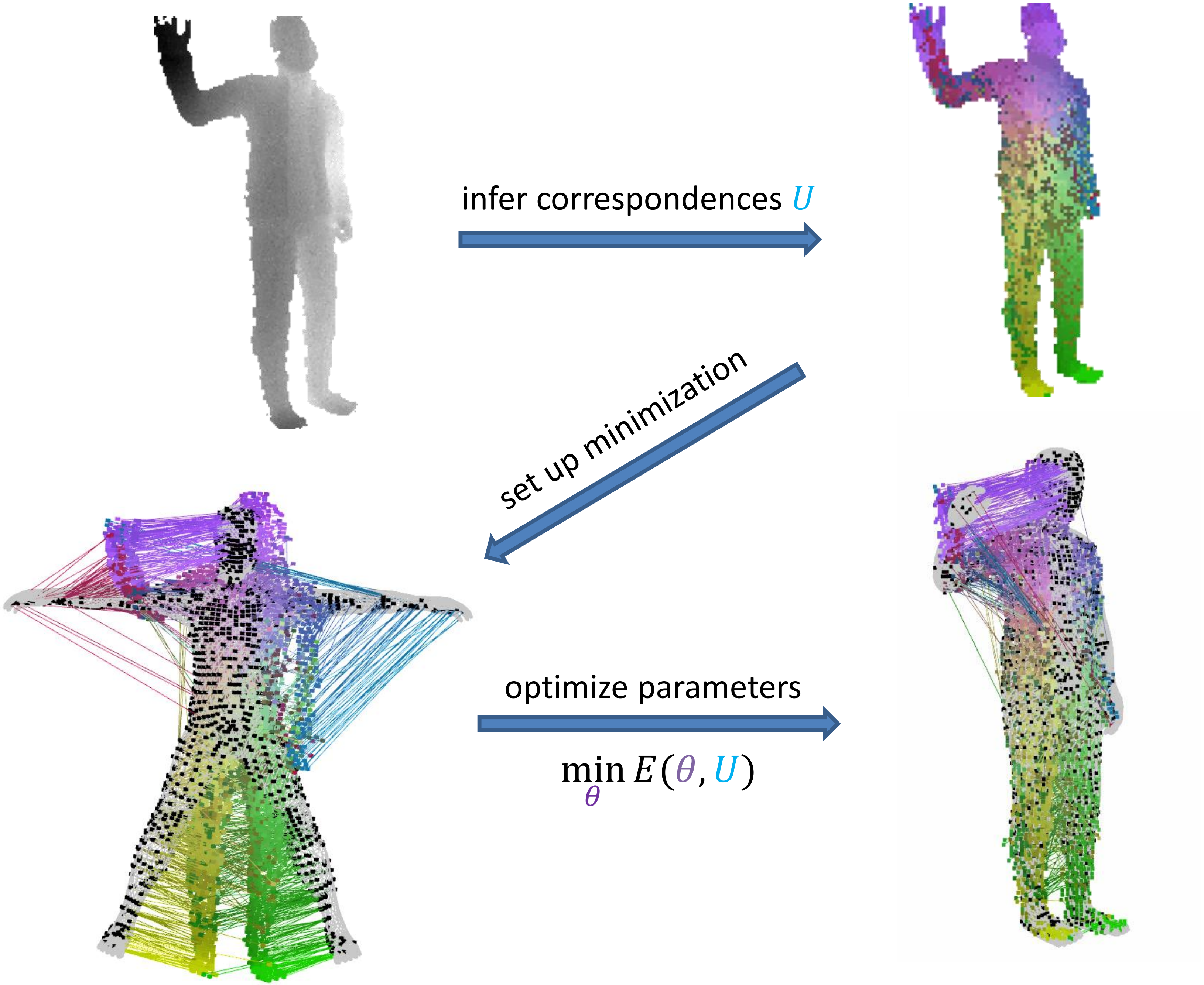
# Learning a Regression Model at the Leaf Nodes



# Inferring Correspondences

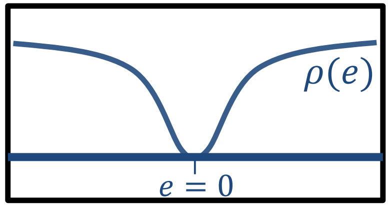




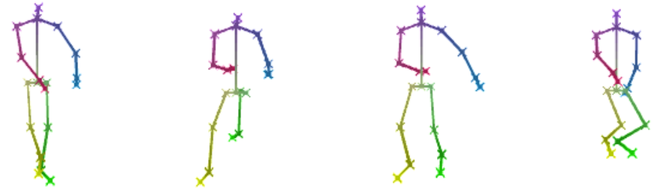
infer correspondences $U$
set up minimization
optimize parameters
$\min_{\theta} E(\theta, U)$

# Full Energy

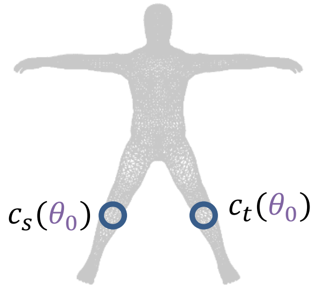
$$\boxed{E(\theta, U) = \lambda_{\text{vis}} E_{\text{vis}}(\theta, U) + \lambda_{\text{prior}} E_{\text{prior}}(\theta) + \lambda_{\text{int}} E_{\text{int}}(\theta)}$$




- Term $E_{\text{vis}}$ approximates hidden surface removal and uses robust error
- Gaussian prior term $E_{\text{prior}}$

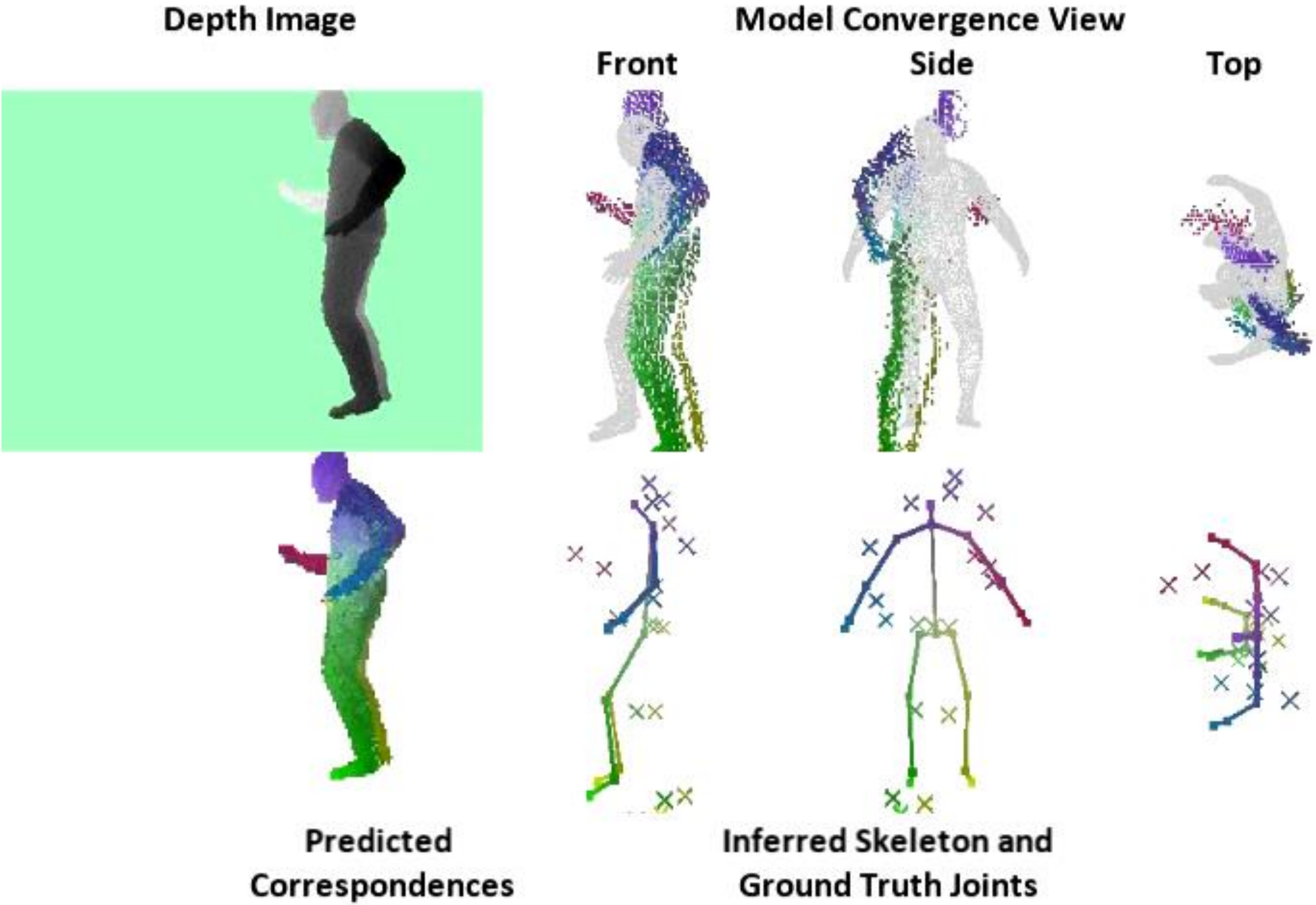


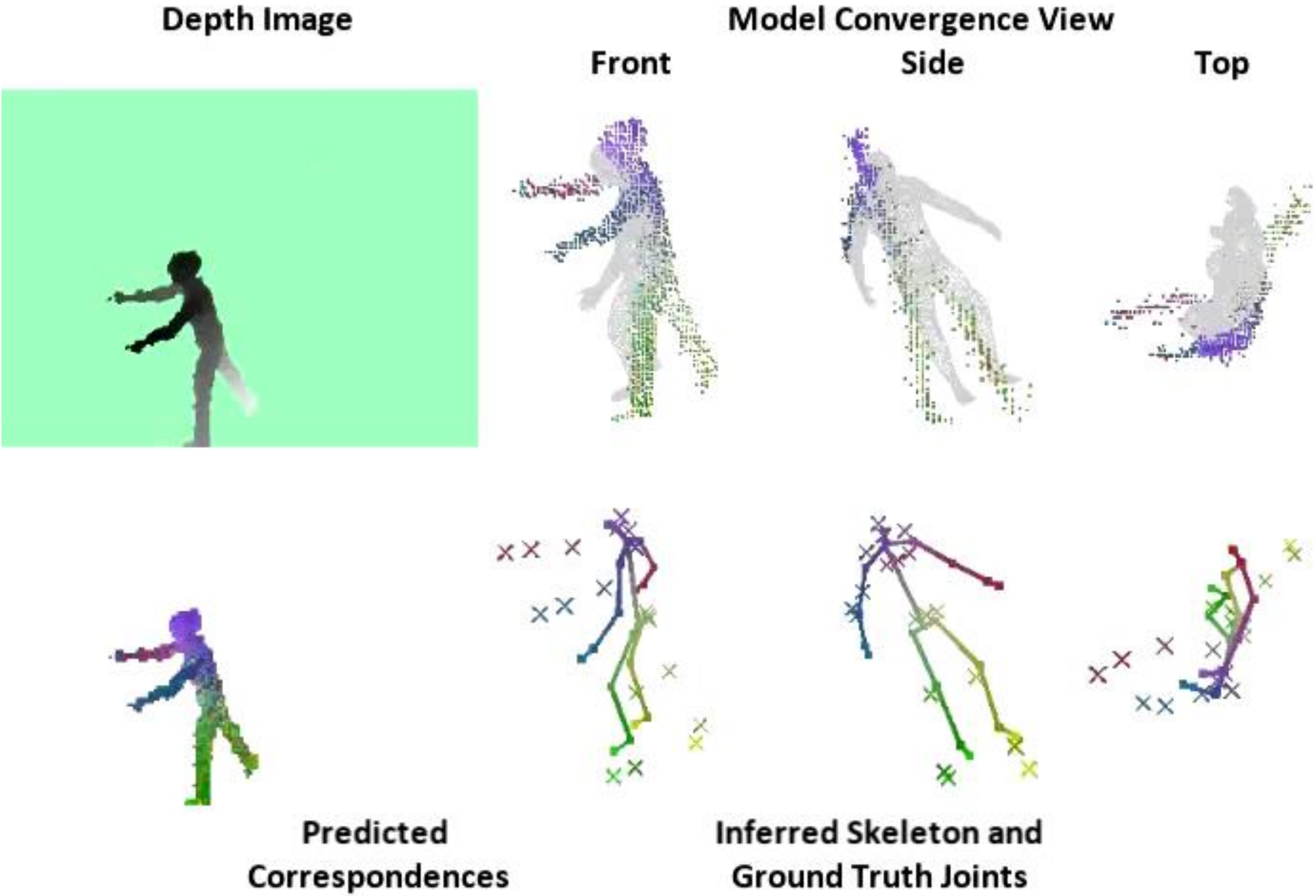
- Self-intersection prior term $E_{\text{int}}$ approximates interior volume



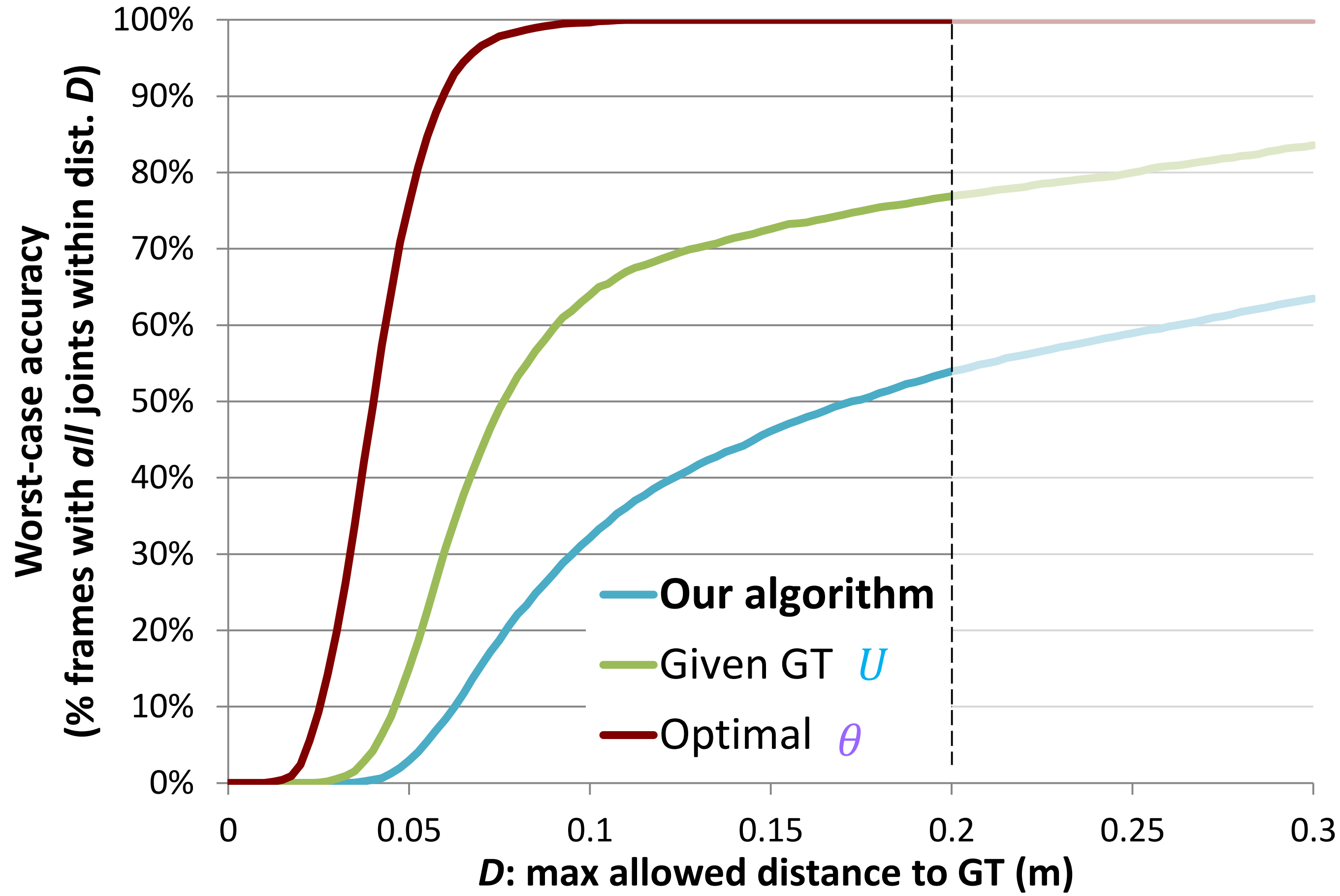

**Energy is robust to noisy correspondences**

- Correspondences far from their image points are "ignored"
- Correspondences facing away from the camera are "ignored"
  - avoids model getting stuck in front of the image pixels

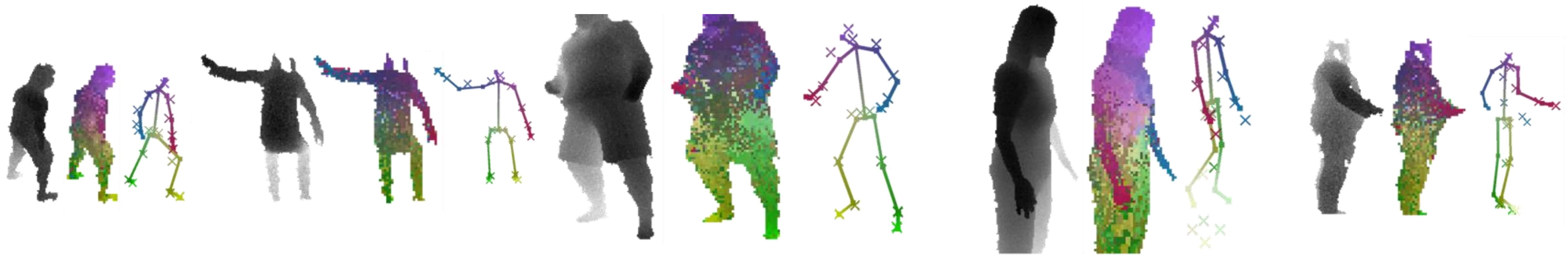

Depth Image
Model Convergence View
Front
Side
Top
Predicted Correspondences
Inferred Skeleton and Ground Truth Joints


Depth Image
Model Convergence View
Front
Side
Top
Predicted Correspondences
Inferred Skeleton and Ground Truth Joints

# Hard Metric:
# "Perfect" Frame Accuracy

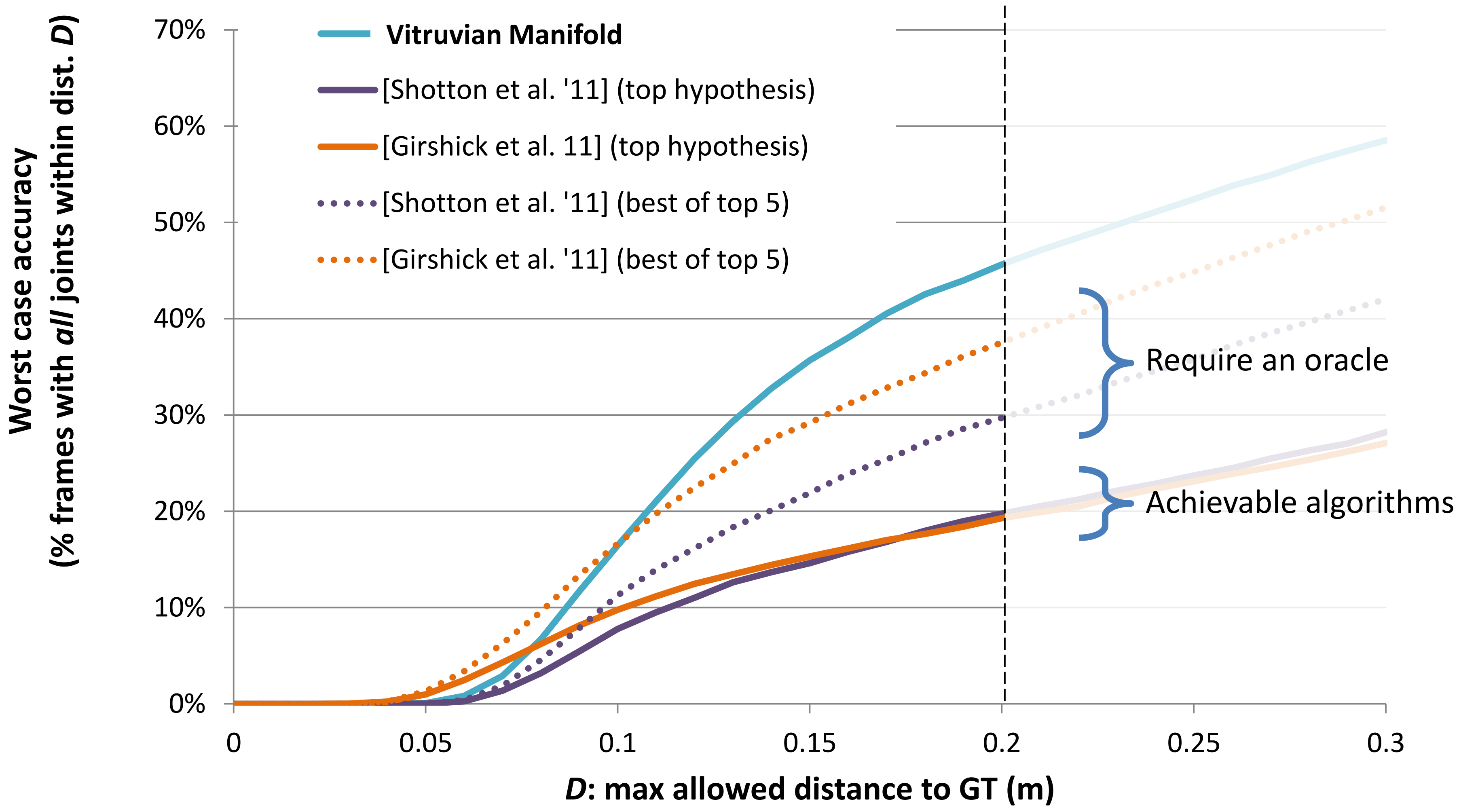



Results on 5000 synthetic images



*D*: 0.09m 0.11m 0.17m 0.21m 0.45m

# Comparison




Results on 5000 synthetic images

# Sequence Result

Depth Image

Predicted Correspondences

Each frame fit independently: no temporal information used

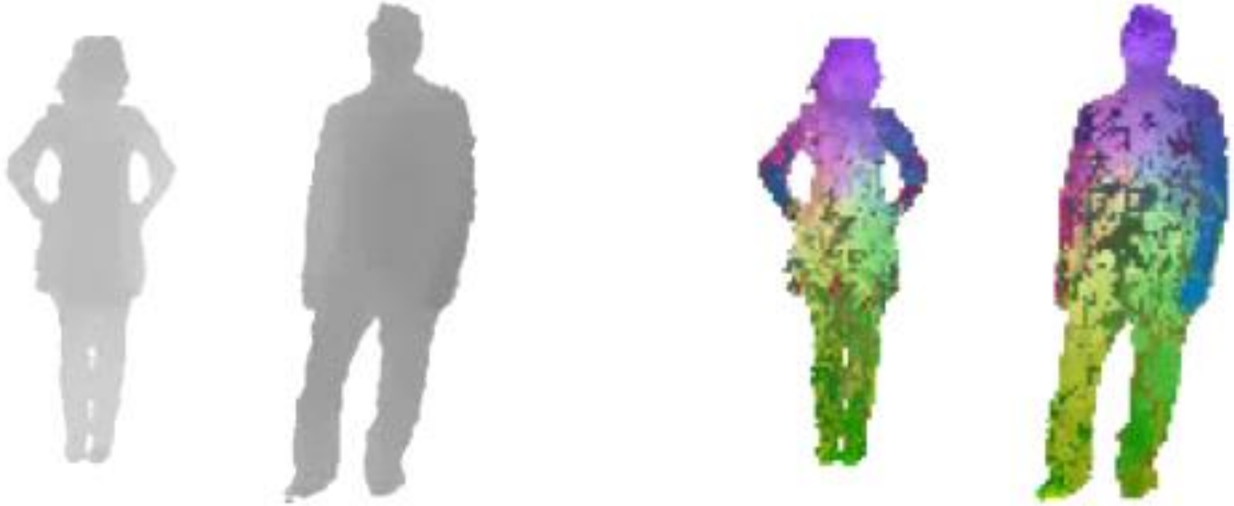
Front

Side

Top

Inferred Skeleton

**Depth Image** | **Predicted Correspondences**



**Note that the algorithm fits the character with strongest signal in each frame.**

**Front** | **Side** | **Top**

**Inferred Skeleton**

# Vitruvian Manifold: Summary

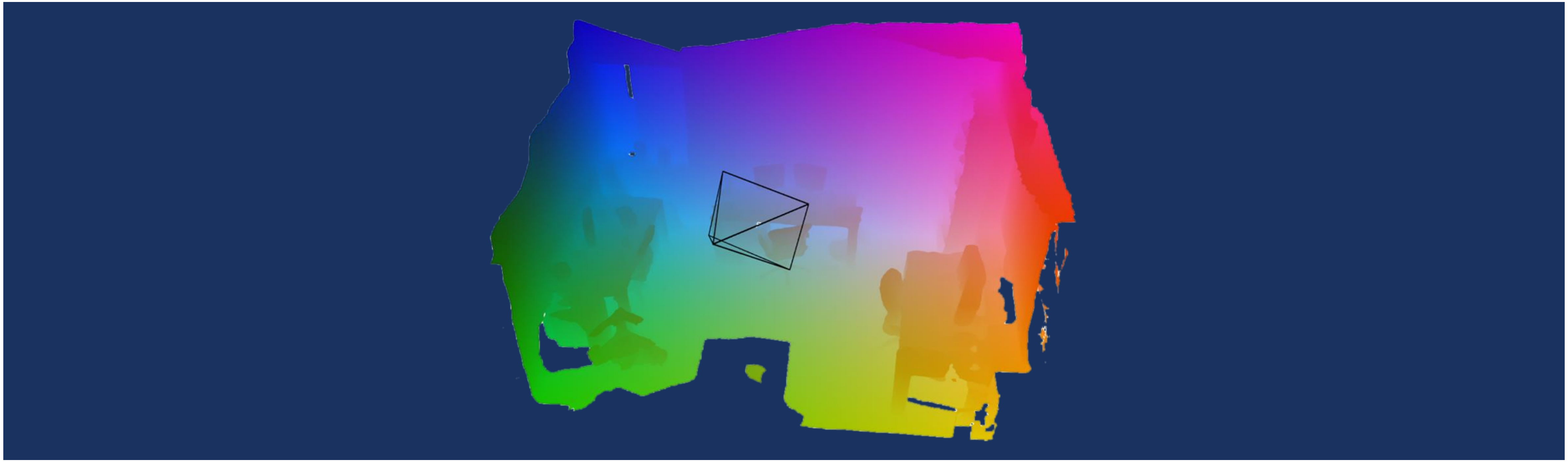
- Predict per-pixel image-to-model correspondences
  – train invariance to body shape, size, and pose

- “One-shot” pose estimation
  – fast, accurate
  – auto-initializes using correspondences

# Scene Coordinate Regression Forests For Camera Relocalization In RGB-D Images

Jamie Shotton Ben Glocker Christopher Zach Shahram Izadi Antonio Criminisi Andrew Fitzgibbon

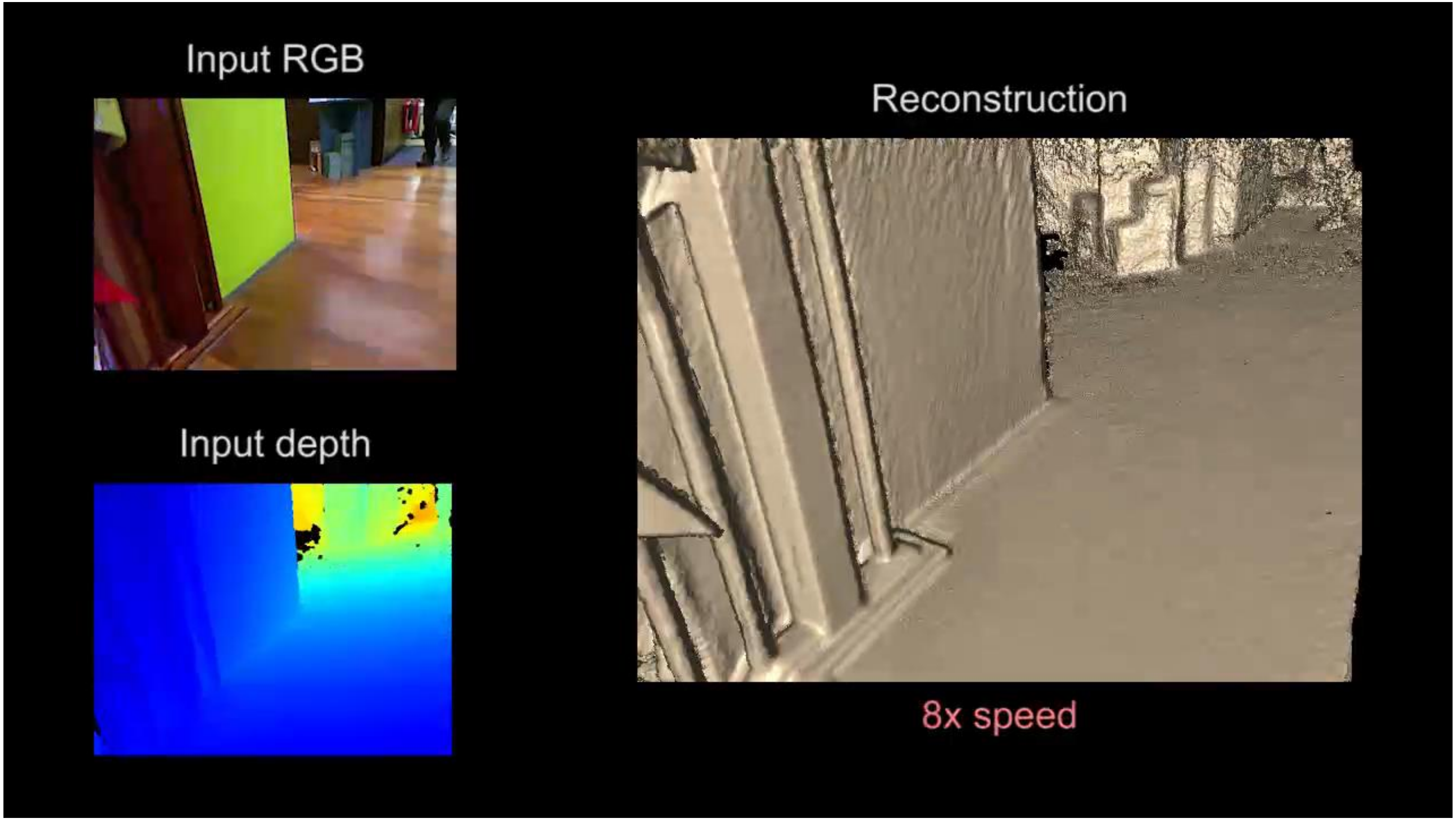
[CVPR 2013]



# KINECTFusion

Joint work with Shahram Izadi, Richard Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Pushmeet Kohli, Steve Hodges, Andrew Davison, Andrew Fitzgibbon. SIGGRAPH, UIST and ISMAR 2011.

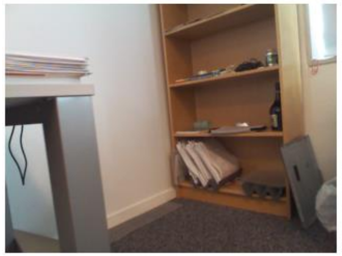
KINECTFusion




Work by: Chen, Bautembach, Izadi. To appear at SIGGRAPH 2013.

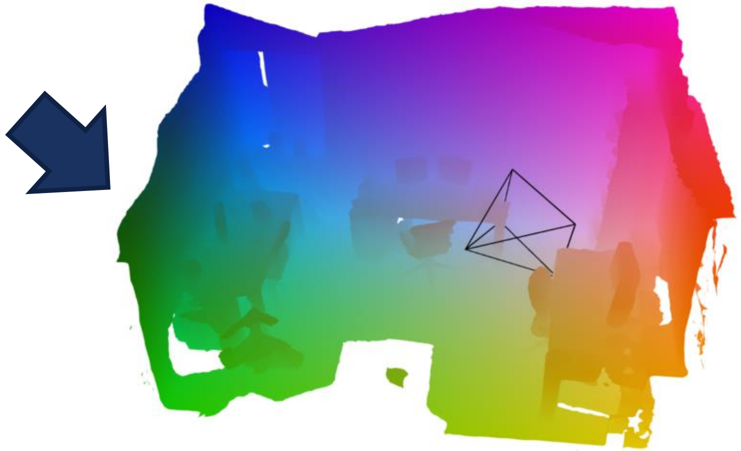
# RELOCALIZATION

- Revisit a known scene
- Observe a single frame of (RGB, Depth)
- Infer the 6D camera pose, $H$
  (camera to scene transformation)



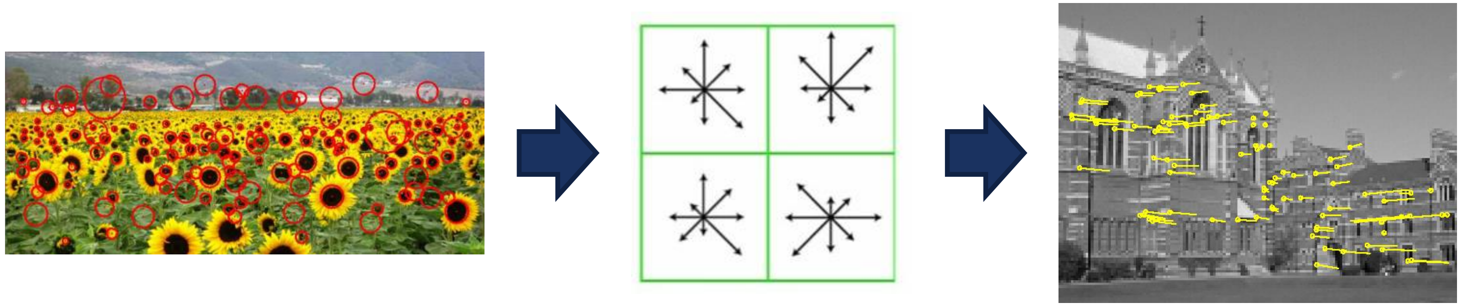
Input RGB

Input Depth



# TYPICAL APPROACHES TO CAMERA LOCALIZATION

- Tracking – alignment relative to previous frame e.g. [Besl & MacKay '92]
- Key point detection → local descriptors → matching → geometric verification
  e.g. [Holzer et al. '12], [Winder & Brown '07], [Lepetit & Fua '06], [Irschara et al. '09]



precise

- Whole key-frame matching e.g. [Klein & Murray 2008] [Gee & Mayol-Cuevas 2012]
- Epitomic location recognition [Ni et al. 2009]

approximate

# PROBLEMS IN REAL WORLD CAMERA LOCALIZATION

- The real world is less exciting than vision researchers might like
  - sparse interest points can fail

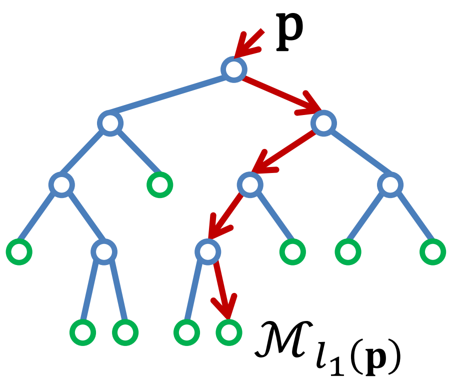
!

- The real world is big

?

# SCENE COORDINATE REGRESSION

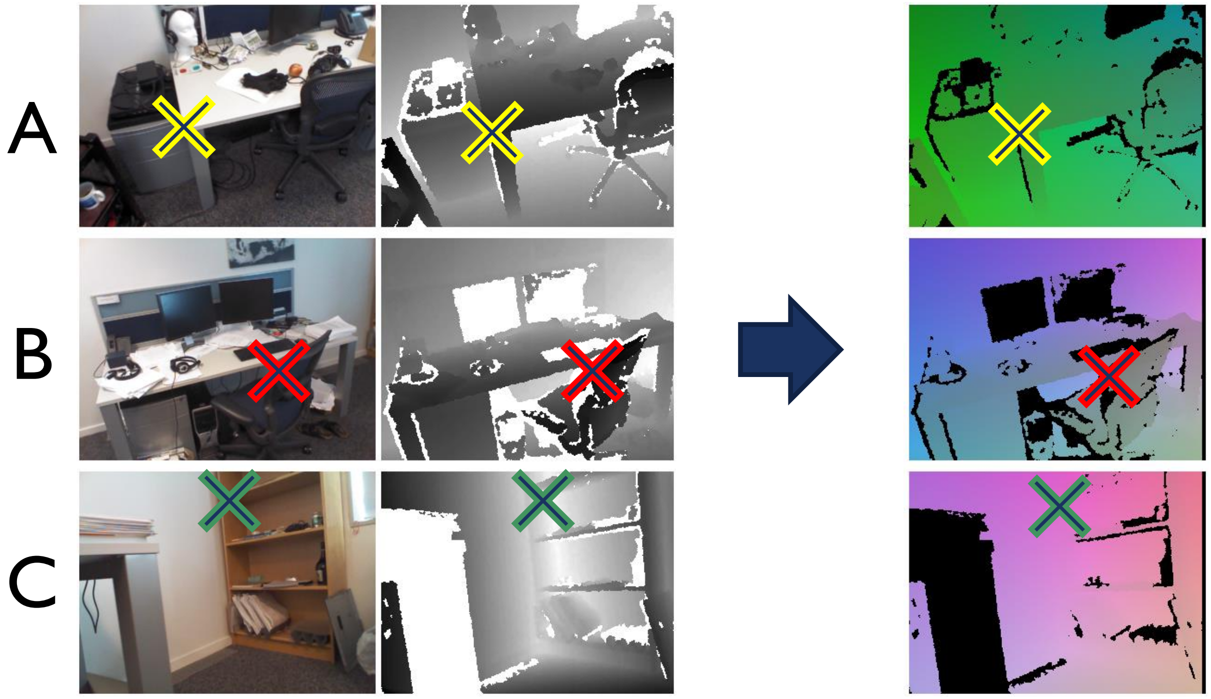
- Offline approach to relocalization
  - observe a scene
  - train a regression forest
  - revisit the scene



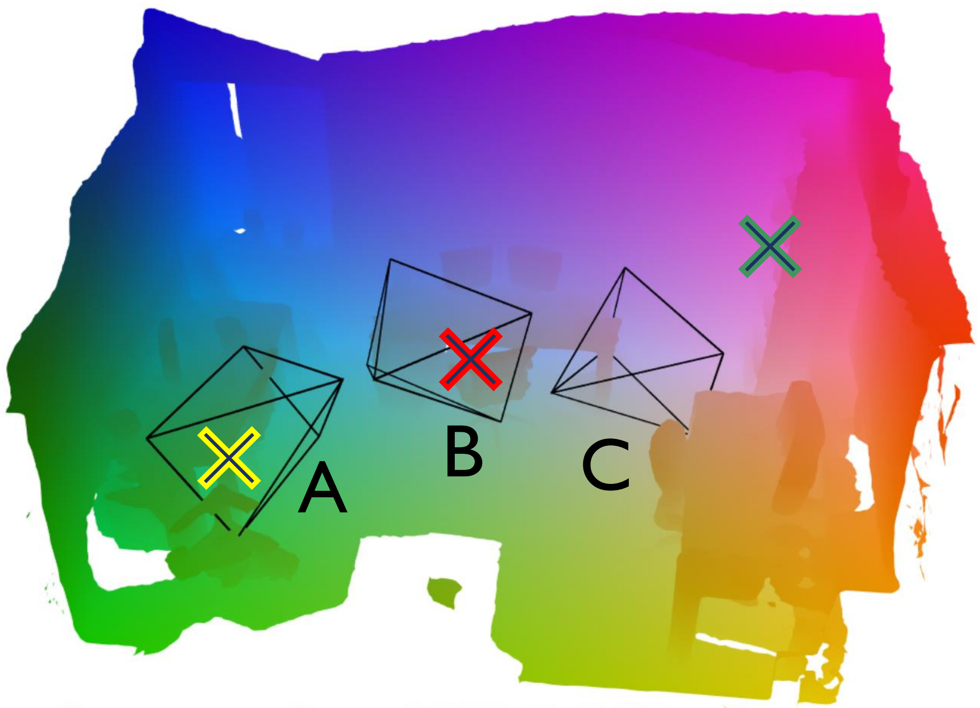

- Aim for really *precise* localization
  - e.g. suitable for AR overlays
  - from a single frame
  - without an explicit 3D model

# SCENE COORDINATE REGRESSION

- Let each pixel predict direct correspondence to 3D point in scene coordinates:



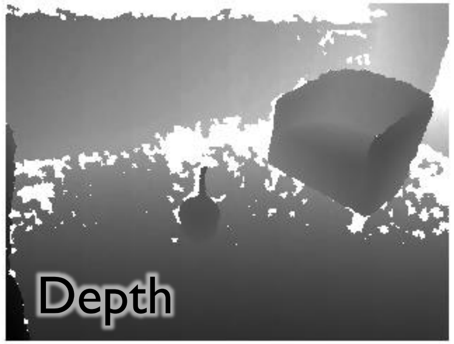

Input RGB Input Depth Desired Correspondences



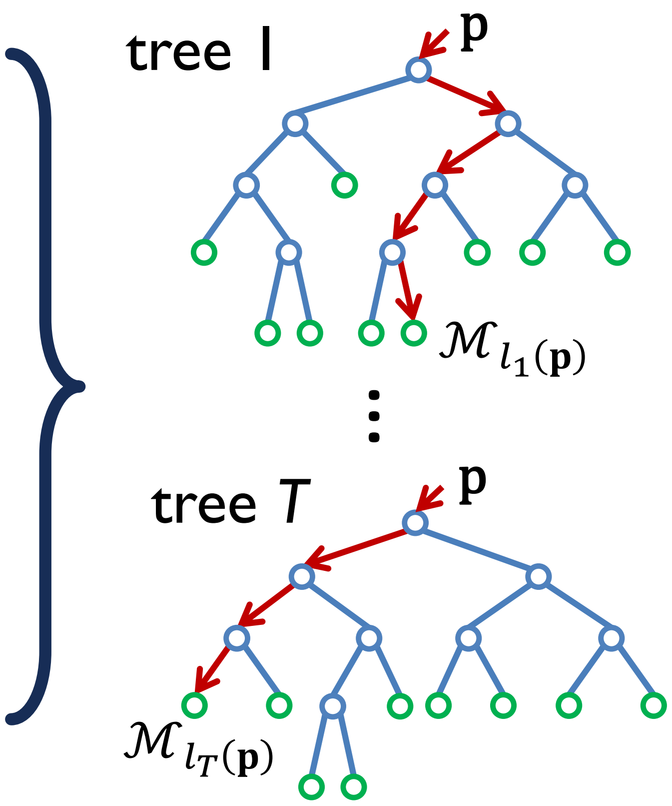

Scene coordinate XYZ ⇔ RGB color space

3D model from KinectFusion
(*only used for visualization*)

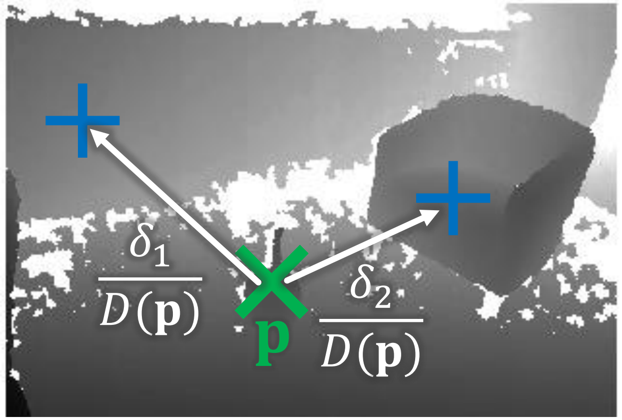
# SCENE COORDINATE REGRESSION (SCoRe) FORESTS


SCoRe Forest




Depth & RGB features




$$f_\phi^{\text{depth}}(\mathbf{p}) = D\left(\mathbf{p} + \frac{\boldsymbol{\delta}_1}{D(\mathbf{p})}\right) - D\left(\mathbf{p} + \frac{\boldsymbol{\delta}_2}{D(\mathbf{p})}\right)$$

$$f_\phi^{\text{da-rgb}}(\mathbf{p}) = I\left(\mathbf{p} + \frac{\boldsymbol{\delta}_1}{D(\mathbf{p})}, c_1\right) - I\left(\mathbf{p} + \frac{\boldsymbol{\delta}_2}{D(\mathbf{p})}, c_2\right)$$

Leaf Predictions $\mathcal{M}_l \subset \mathbb{R}^3$

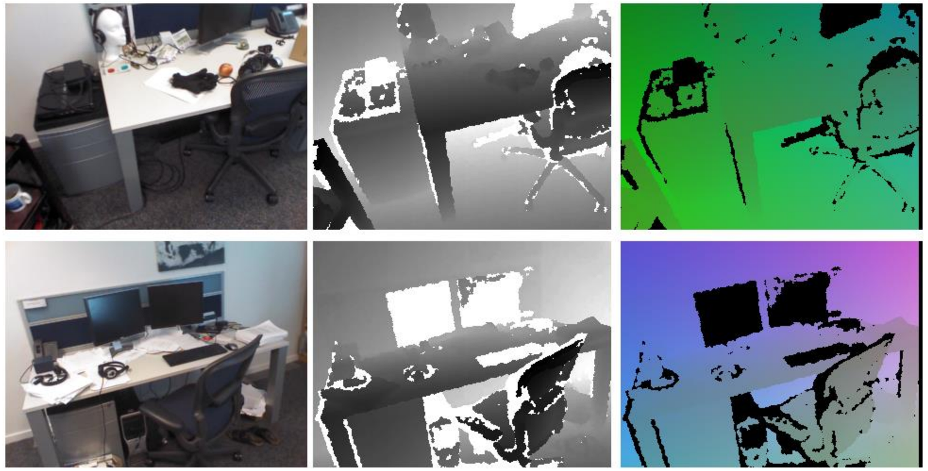
Forest Predictions $\mathcal{M}(\mathbf{p}) = \bigcup_t \mathcal{M}_{l_t(\mathbf{p})}$

# TRAINING A SCoRe FOREST

## Training Data

- RGB-D frames with known camera poses $H$
- Generate 3D pixel labels automatically:

$$\mathbf{m} = H\mathbf{x}$$



RGB | Depth $\{\mathbf{x}\}$ | Labels $\{\mathbf{m}\}$

## Learning (standard)

- Greedily train tree
- Reduction in spatial variance objective:

$$Q(\mathcal{S}_n, \boldsymbol{\theta}) = V(\mathcal{S}_n) - \sum_{d \in \{\mathrm{L},\mathrm{R}\}} \frac{|\mathcal{S}_n^d(\boldsymbol{\theta})|}{|\mathcal{S}_n|} V(\mathcal{S}_n^d(\boldsymbol{\theta}))$$

$$\text{with} \quad V(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{p},\mathbf{m}) \in \mathcal{S}} \|\mathbf{m} - \bar{\mathbf{m}}\|_2^2$$

- Regression, not classification
- Mean shift to summarize distribution at leaf $l$ into small set $\mathcal{M}_l \subset \mathbb{R}^3$

# SCoRE FORESTS: PROPERTIES

- A single-step alternative to the traditional pipeline
  - interest point detection ⇨ description ⇨ matching
- In theory, only three 3D⇔3D correspondences needed to infer 6D camera pose
  - Kabsch algorithm (a.k.a. orthogonal Procrustes alignment)
- Thus, only need to apply forest at three test image pixels
  - *any* three pixels will do
  - sparseness gives efficiency
  - in practice, noise in prediction means we use more than three pixels

# ROBUST CAMERA POSE OPTIMIZATION

## Energy Function

robust
error function

$$E(H) = \sum_{i \in \mathcal{I}} \rho \left( \min_{\mathbf{m} \in \mathcal{M}_i} \|\mathbf{m} - H\mathbf{x}_i\|_2 \right)$$

camera pose

pixel index

correspondences
predicted
by forest
at pixel $i$

## Optimization

- Preemptive RANSAC

  [Nistér ICCV 2003]

- With pose refinement

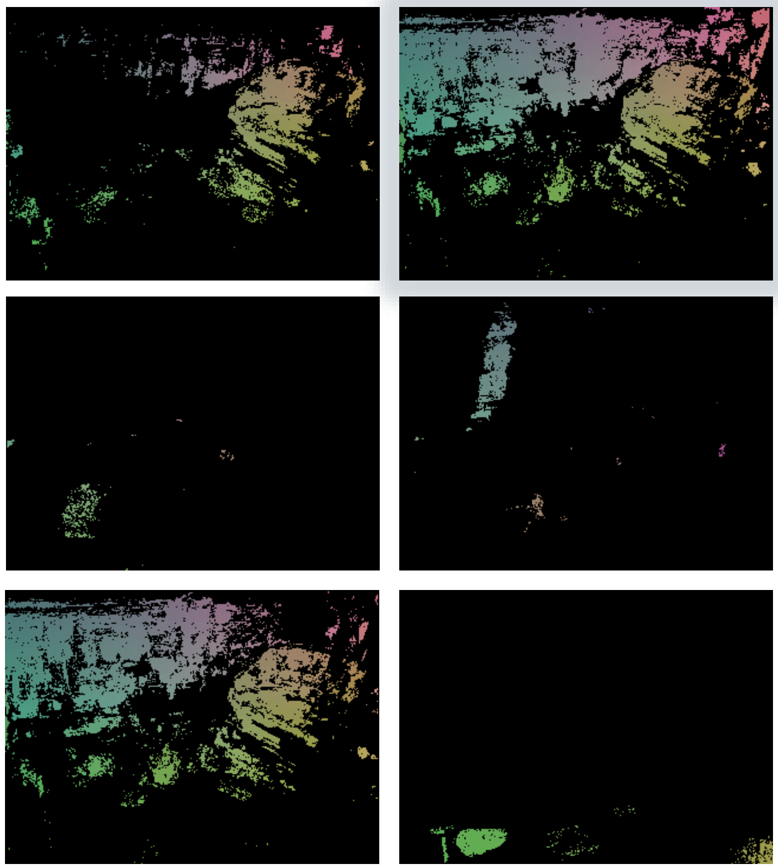
  [Chum et al. DAGM 2003]

  - efficient updates to means & covariances used by Kabsch SVD

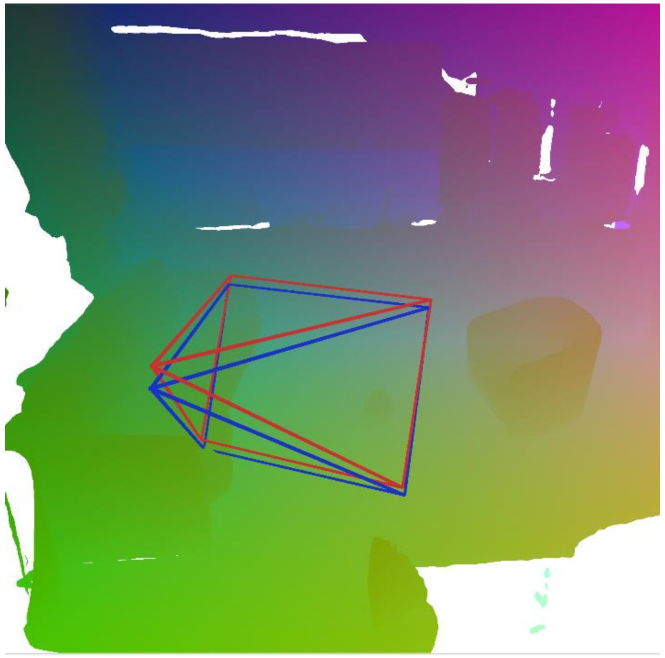
- Only a small subset of pixels used

# INLYING FOREST PREDICTIONS

Test images

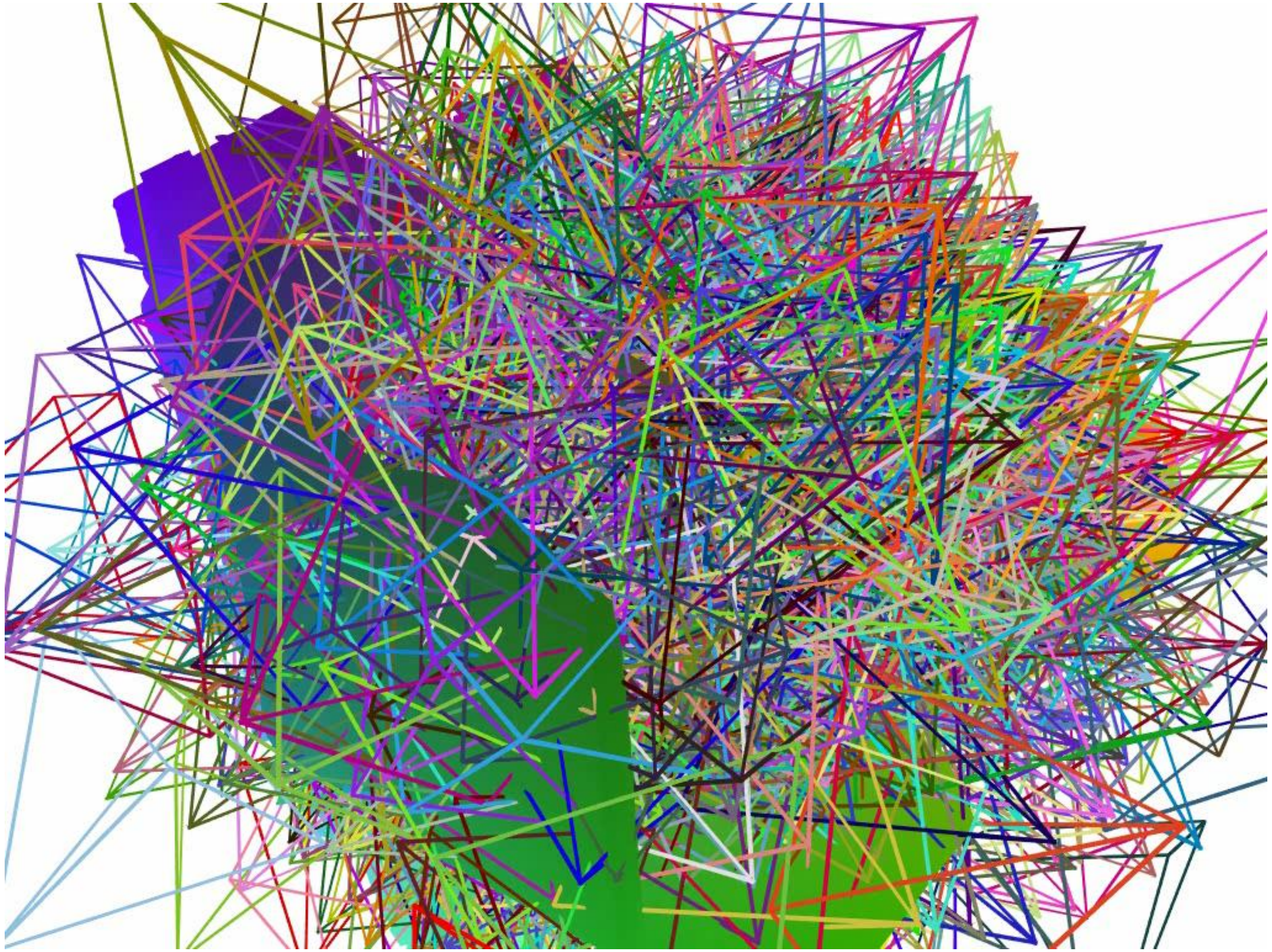


Inliers for six hypotheses



Inferred camera pose

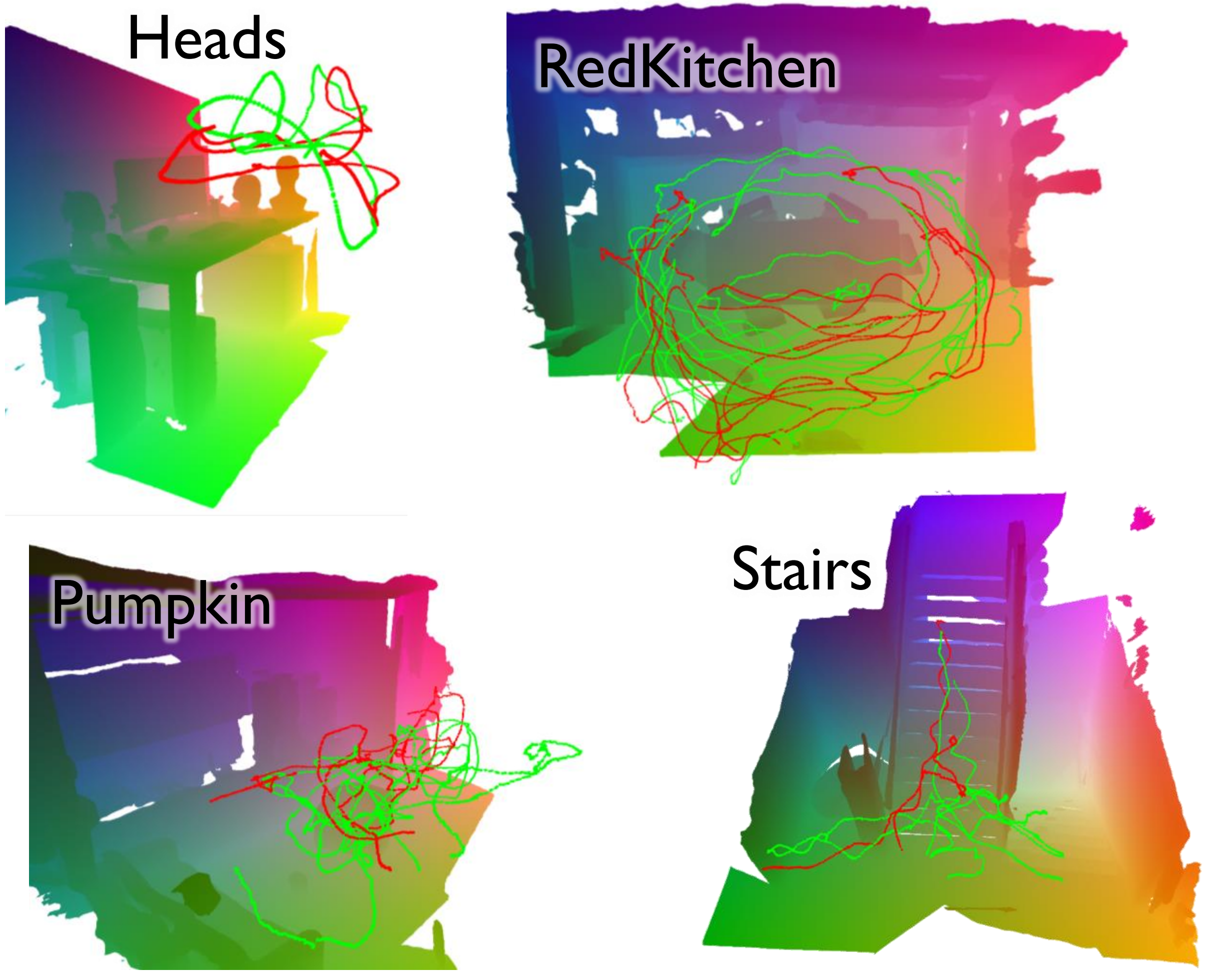
# PREEMPTIVE RANSAC OPTIMIZATION



# THE 7SCENES DATASET

| Scene | Spatial Extent | # Frames Train | # Frames Test |
|---|---|---|---|
| Chess | $3m^3$ | 4k | 2k |
| Fire | $4m^3$ | 2k | 2k |
| Heads | $2m^3$ | 1k | 1k |
| Office | $5.5m^3$ | 6k | 4k |
| Pumpkin | $6m^3$ | 4k | 2k |
| RedKitchen | $6m^3$ | 7k | 5k |
| Stairs | $5m^3$ | 2k | 1k |

Dataset to be released at CVPR



# BASELINES FOR COMPARISON

## Sparse Key-Points (RGB only)

- ORB matching
  [Rublee et al. ICCV 2011]
  - FAST detector
  - Rotation aware BRIEF descriptor
  - Hashing for matching
- Geometric verification
  - RANSAC & perspective 3 point
  - Final refinement given inliers

## Tiny-Image Key-Frames (RGB & Depth)

- Downsample to 40x30 pixels
- Blur
- Normalized Euclidean distance
- Brute-force search
- Interpolation of 100 closest poses

[Klein & Murray ECCV 2008]

[Gee & Mayol-Cuevas BMVC 2012]

# QUANTITATIVE COMPARISON

Metric:

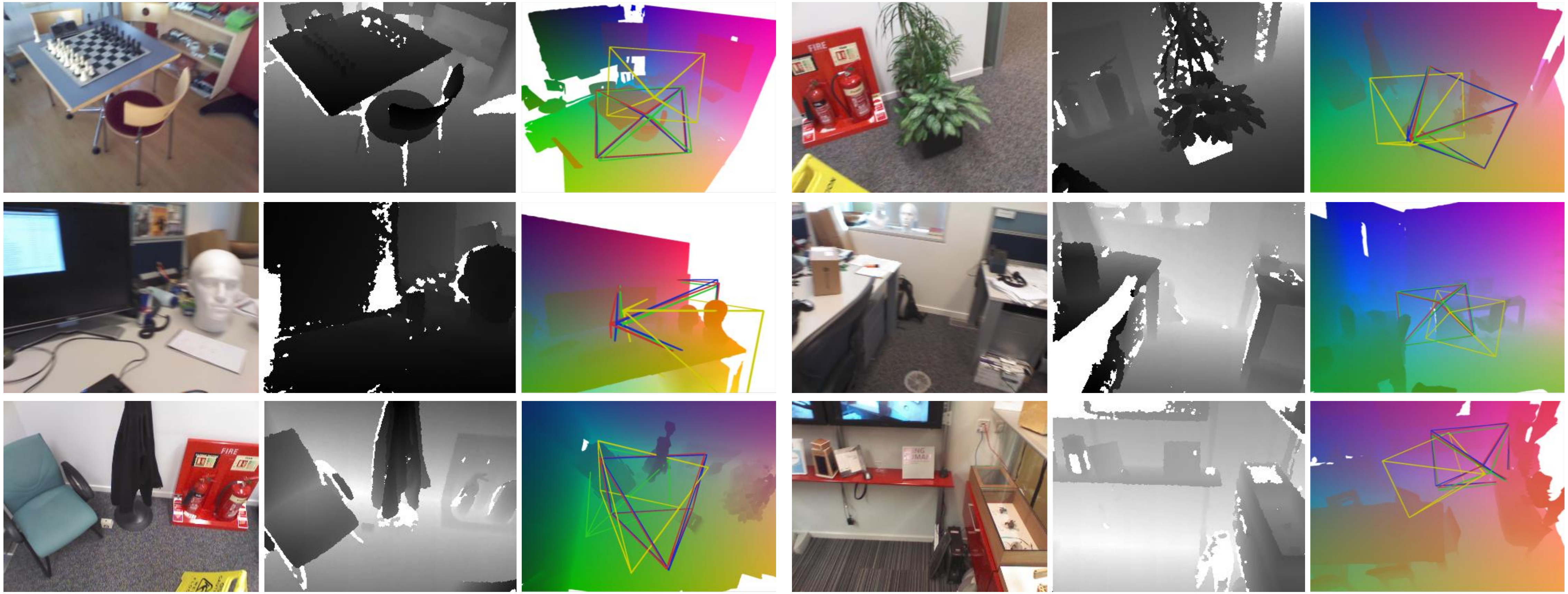
Proportion of test frames with < 0.05m translational error and < 5° angular error

Results:

| | Baselines | | Our Results | | |
|---|---|---|---|---|---|
| **Scene** | Tiny-image RGB-D | Sparse RGB | Depth | DA-RGB | DA-RGB + D |
| Chess | 0.0% | 70.7% | 82.7% | **92.6%** | 91.5% |
| Fire | 0.5% | 49.9% | 44.7% | **82.9%** | 74.7% |
| Heads | 0.0% | **67.6%** | 27.0% | 49.4% | 46.8% |
| Office | 0.0% | 36.6% | 65.5% | 74.9% | **79.1%** |
| Pumpkin | 0.0% | 21.3% | 58.6% | **73.7%** | 72.7% |
| RedKitchen | 0.0% | 29.8% | 61.3% | 71.8% | **72.9%** |
| Stairs | 0.0% | 9.2% | 12.2% | **27.8%** | 24.4% |

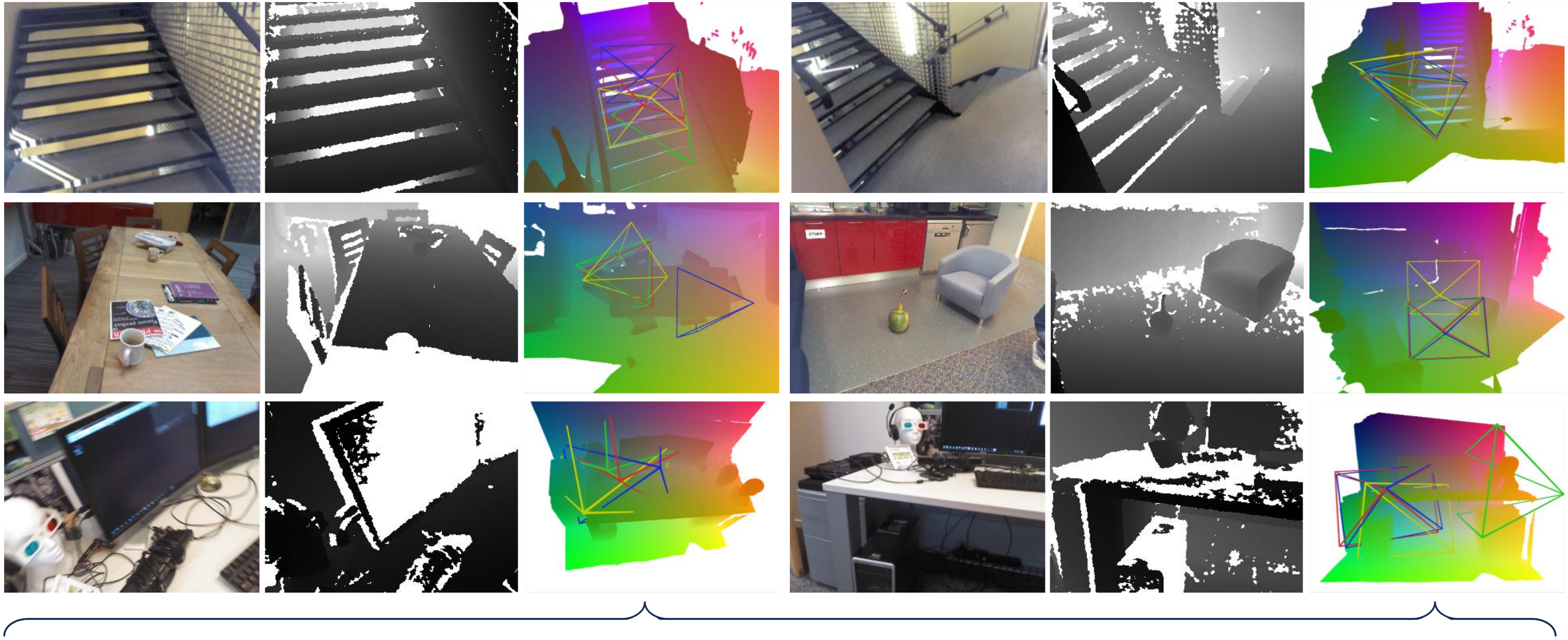
Choice of different image features

# QUALITATIVE COMPARISON



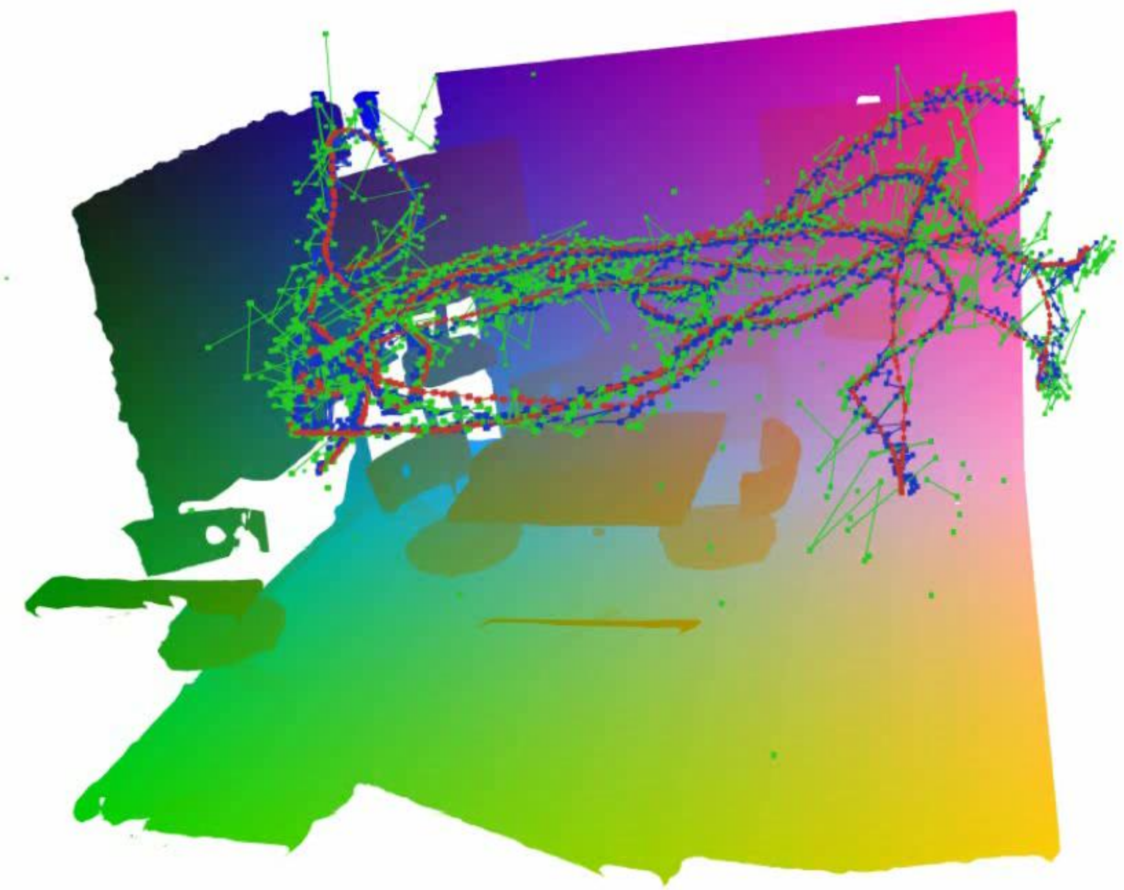

ground truth DA-RGB SCoRe forest sparse baseline closest training pose

# QUALITATIVE COMPARISON



ground truth DA-RGB SCoRe forest sparse baseline closest training pose

# TRACK VISUALIZATION VIDEOS



ground truth

**DA-RGB SCoRe forest**

RGB sparse baseline

single frame at a time – no tracking

# AR VISUALIZATION

RGB input
+ AR overlay

depth input
+ AR overlay

rendering of model
from inferred pose

single frame at a time – no tracking

# SIMPLE ROBUST TRACKING

- Add a single extra hypothesis to optimization: the result from previous frame

| Scene | Our Results: Depth | Our Results: DA-RGB | Our Results: DA-RGB + D | Frame-to-Frame Tracking |
|---|---|---|---|---|
| Chess | 82.7% | **92.6%** | 91.5% | 95.5% |
| Fire | 44.7% | **82.9%** | 74.7% | 86.2% |
| Heads | 27.0% | 49.4% | 46.8% | 50.7% |
| Office | 65.5% | 74.9% | **79.1%** | 86.8% |
| Pumpkin | 58.6% | **73.7%** | 72.7% | 76.1% |
| RedKitchen | 61.3% | 71.8% | **72.9%** | 82.4% |
| Stairs | 12.2% | **27.8%** | 24.4% | 39.2% |

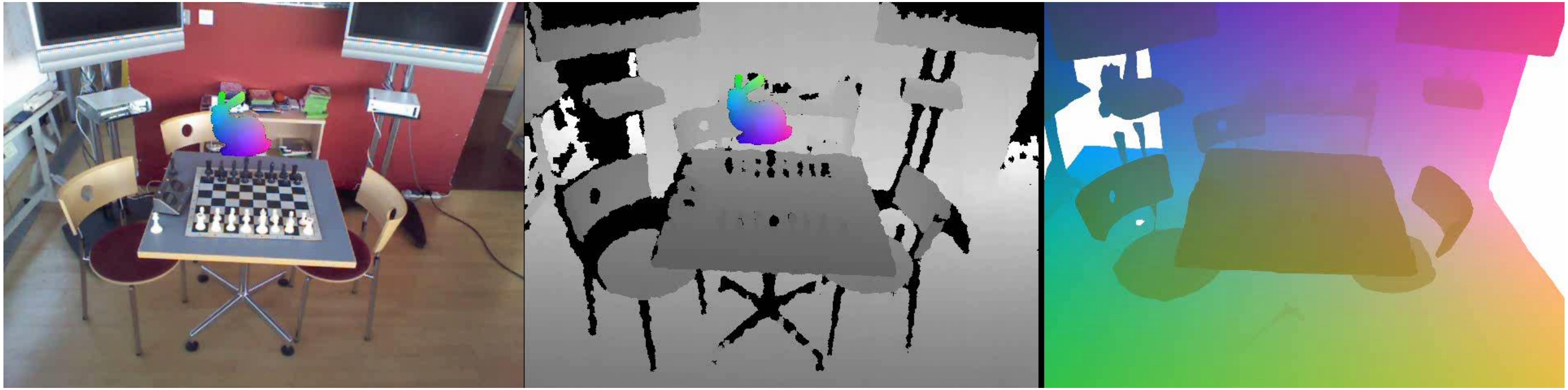
Single frame

# AR VISUALIZATION WITH TRACKING

RGB input
+ AR overlay

depth input
+ AR overlay

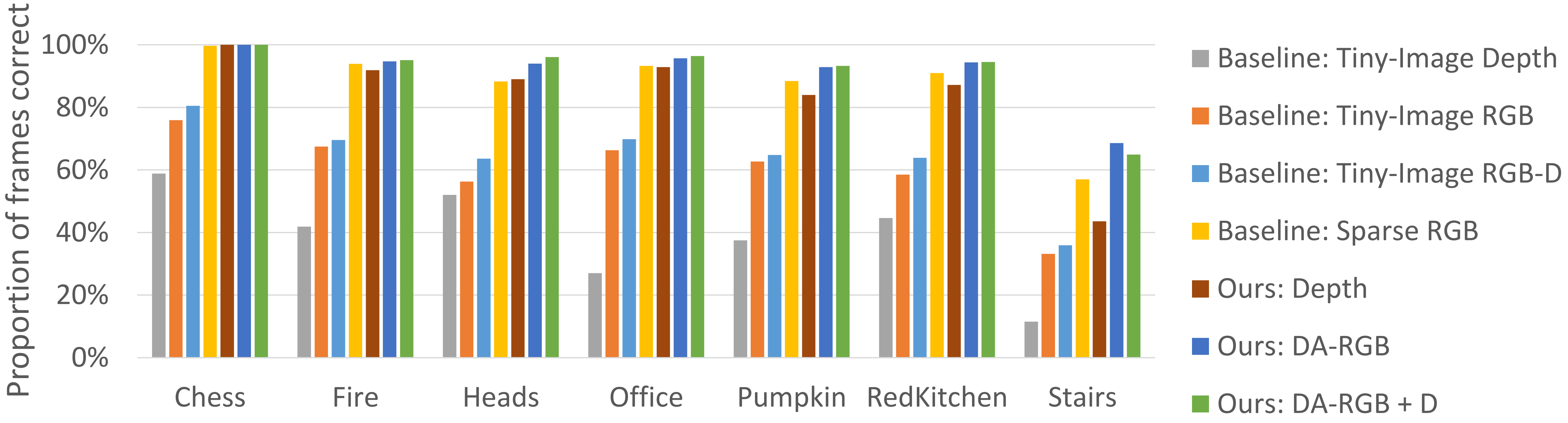
rendering of model
from inferred pose



simple robust frame-to-frame tracking enabled

# MODEL-BASED REFINEMENT

- Model-based refinement [Besl & McKay PAMI 1992]
  - requires 3D model of scene
  - run ICP from our inferred pose between observed image and model

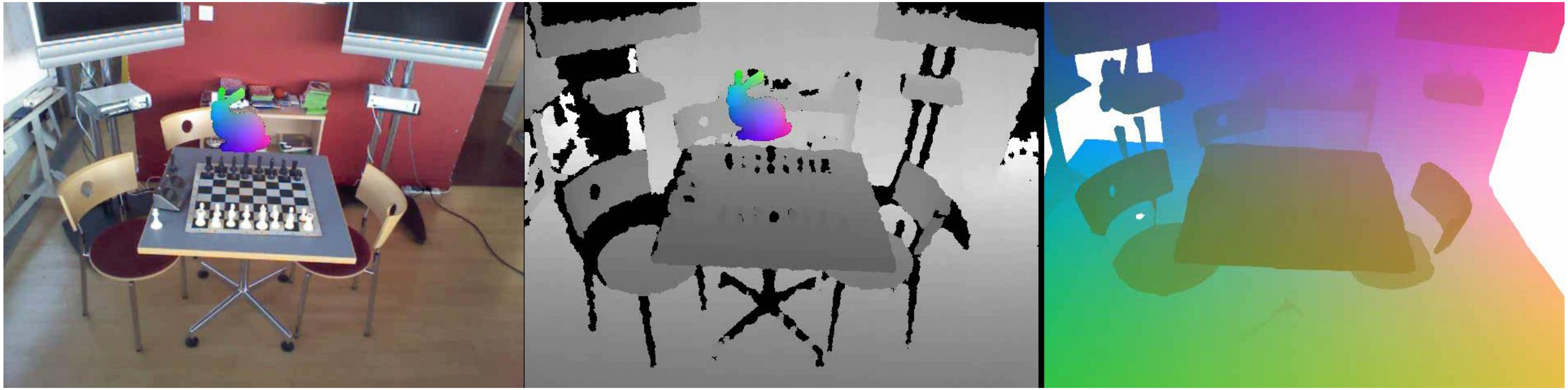


# AR VISUALIZATION WITH TRACKING AND REFINEMENT

RGB input
+ AR overlay

depth input
+ AR overlay

rendering of model
from inferred pose



simple robust frame-to-frame tracking and ICP-based model refinement enabled

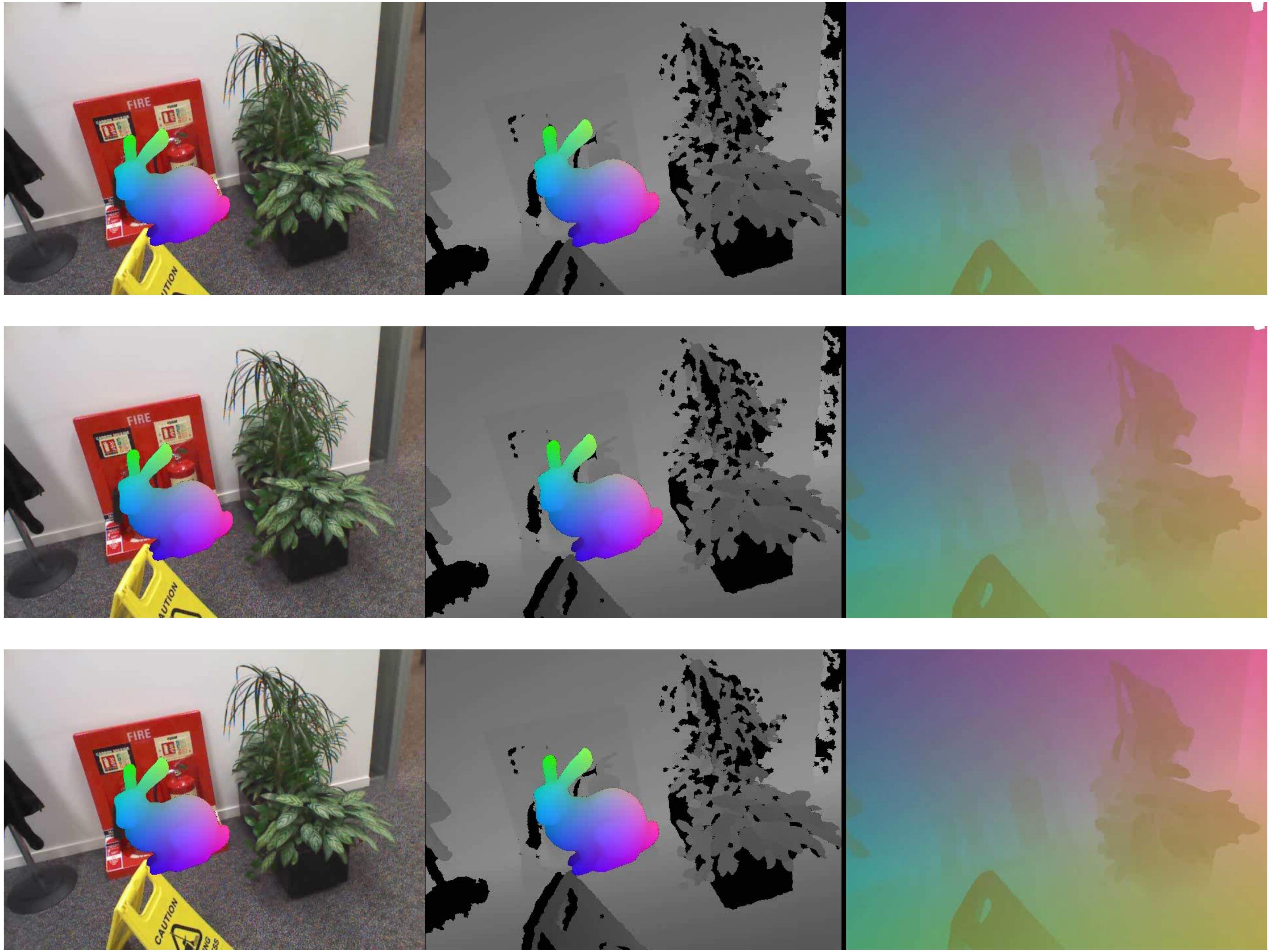
# Fire Scene

RGB input + AR overlay | depth input + AR overlay | rendering of model from inferred pose

SCoRe Forest (single frame at a time)

SCoRe Forest + simple robust frame-to-frame tracking

SCoRe Forest + simple robust frame-to-frame tracking + ICP refinement to 3D model



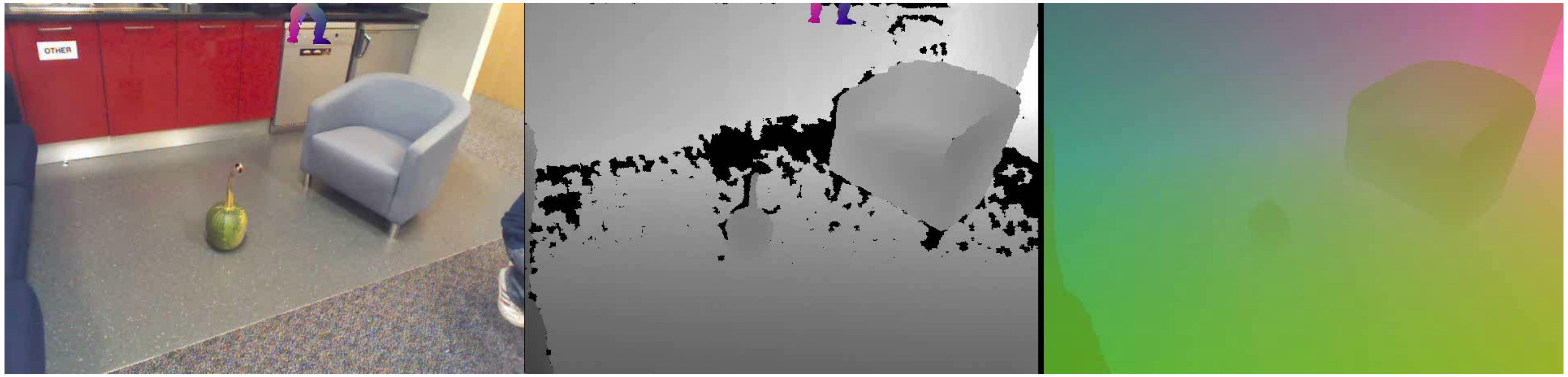
# Pumpkin Scene

RGB input + AR overlay | depth input + AR overlay | rendering of model from inferred pose

SCoRe Forest
(single frame at a time)

SCoRe Forest
+
simple robust
frame-to-frame tracking

SCoRe Forest
+
simple robust
frame-to-frame tracking
+
ICP refinement to 3D model



# SCENE RECOGNITION

- Train one SCoRe Forest per scene
- Test frame against all scenes
- Scene with lowest energy wins
- Single frame only

| | Chess | Fire | Heads | Office | Pumpkin | RedKitchen | Stairs |
|---|---|---|---|---|---|---|---|
| Chess | 100.0% | | | | | | |
| Fire | 2.0% | 98.0% | | | | | |
| Heads | | | 100.0% | | | | |
| Office | | 0.5% | 4.0% | 95.5% | | | |
| Pumpkin | | | | | 100.0% | | |
| RedKitchen | 2.8% | 1.2% | 3.6% | | | 92.4% | |
| Stairs | | | 10.0% | | | | 90.0% |

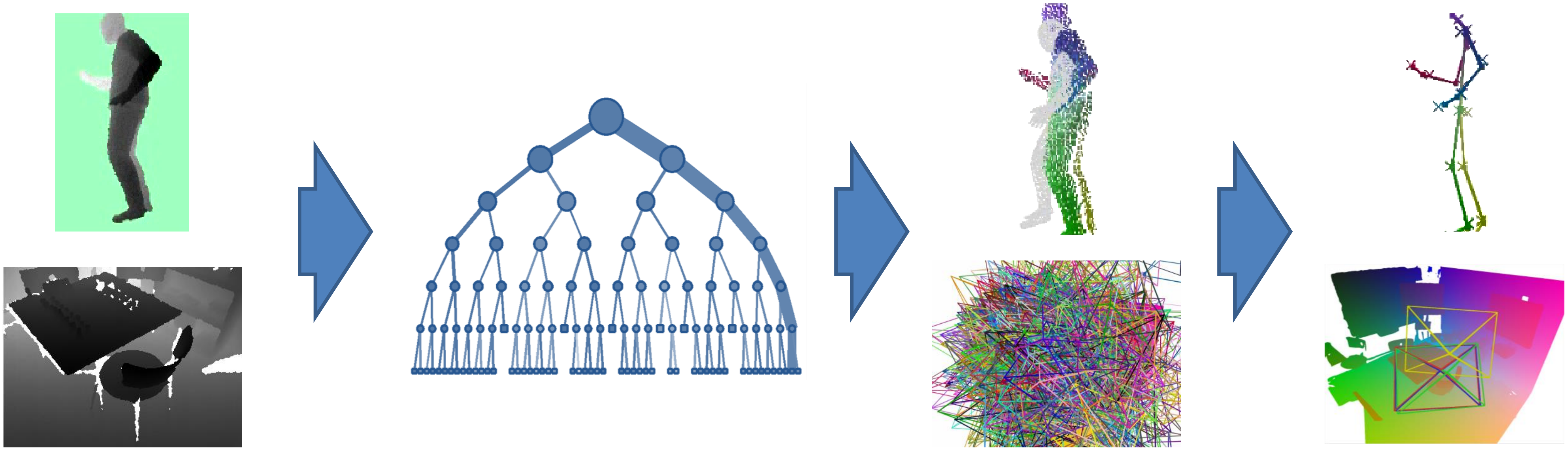
# SCENE COORDINATE REGRESSION - SUMMARY

- Scene coordinate regression forests
  - allow accurate relocalization without explicit 3D model
  - provide a single-step alternative to detection/description/matching pipeline
  - can be applied at any valid pixel, not just at interest points

- Tracking-by-detection is approaching temporal tracking accuracy

# Wrap Up

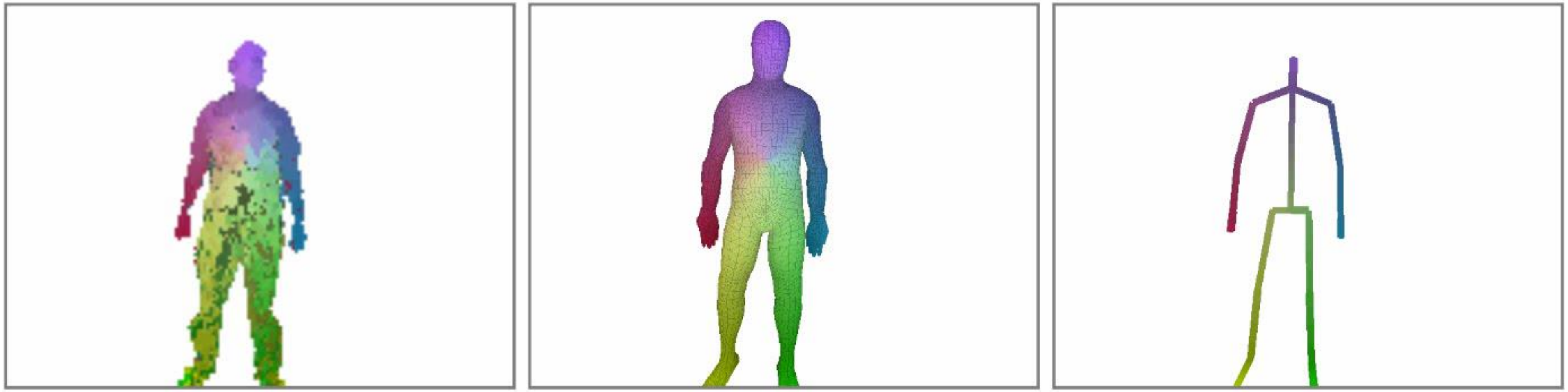
- New depth sensors
- Machine learning + big (synthetic) data
- Per-pixel regression *and* per-image model fitting



XBOX ONE

Coming November…

# Thank you!




## With thanks to:

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

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