KINECT FOR XBOX 360

SPAWN POINT

QUICK INTRODUCTION

- Creative Director on Kinect
- 15 years in gaming industry
- < 3 years at Microsoft

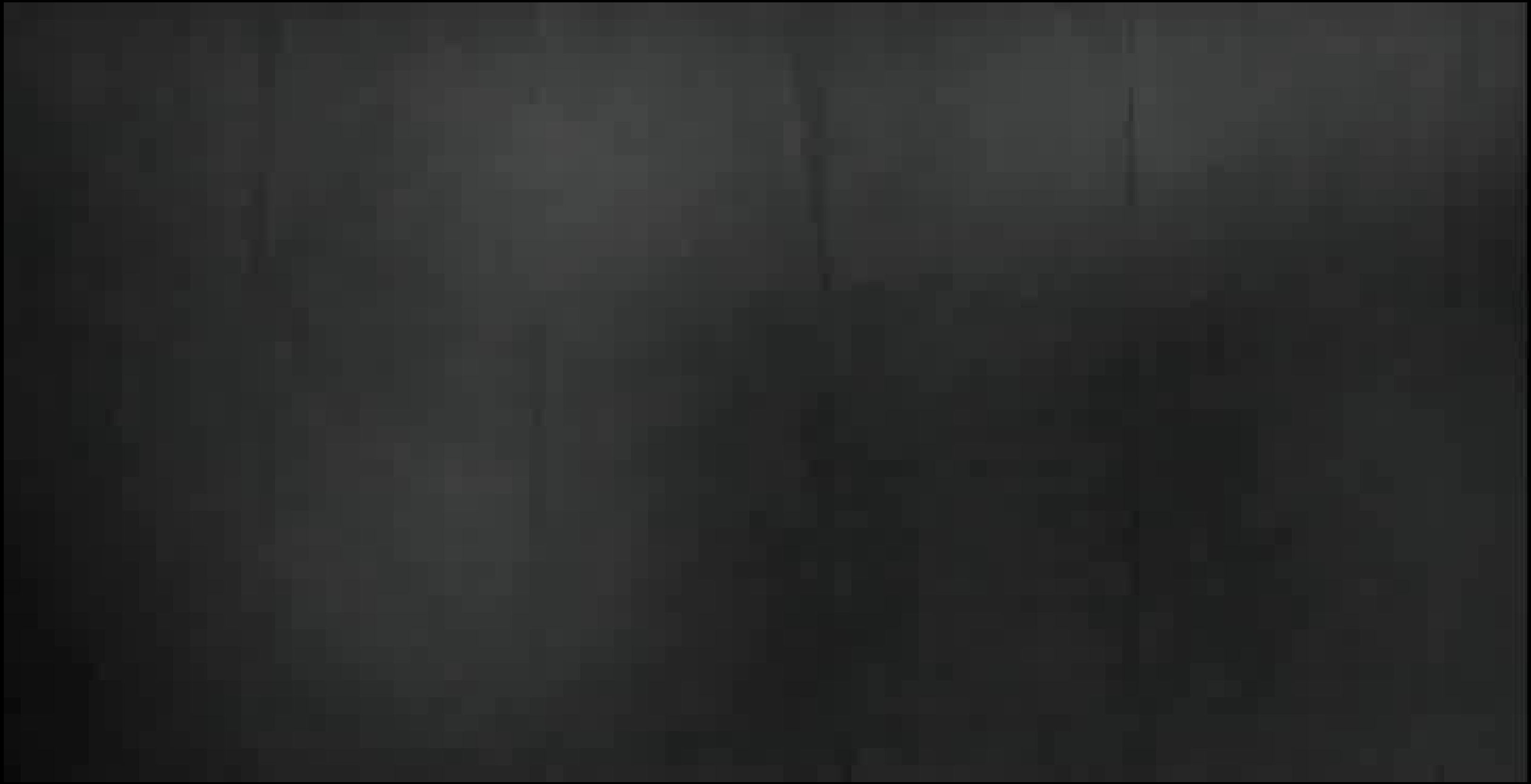
KINECT CONCEPT

" The purpose of Kinect is to make Xbox more accessible to a broader audience"

KINECT KEY PILLARS

Key Pillars:

1. Unique to Kinect
2. Approachable
3. Social
4. As fun to watch as it is to play
5. Play any way you want to
6. Redefine Microsoft approach to broadening



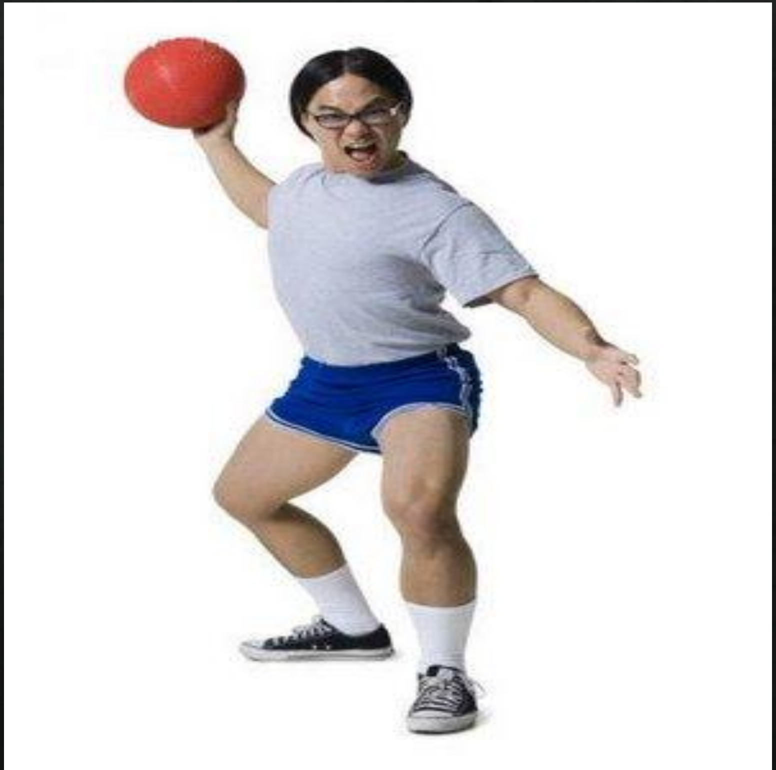
KINECT FEATURES

Key Features:

1. Avateering
2. Voice Rec and Party Chat
3. Recognizing People and Objects
4. Stuff Works!!!!

TRULY BROADENING!!!!!

DEMOS



FINAL THOUGHTS…

Making a natural experience is UN-NATURAL

FINAL THOUGHTS…

MSR is pretty darn AWESOME!!!!

FINAL THOUGHTS…

We are all learning about Kinect together.

FINAL THOUGHTS…

Love to hear more from people at MSR:

Kudot@microsoft.com

Microsoft® Research

Faculty Summit 2010

Microsoft® Research
Faculty Summit 2010

Kinect for Xbox 360
The Innovation Journey

Kudo Tsunoda
Creative Director – Kinect
Microsoft Game Studios

Andrew Fitzgibbon
Principal Researcher
Microsoft Research Cambridge

Microsoft® Research
Faculty Summit 2010

From Natal to Kinect

Andrew Fitzgibbon
Principal Researcher
Microsoft Research Cambridge

Research .... and Product

SIGGRAPH2004

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother* Vladimir Kolmogorov† Andrew Blake‡
Microsoft Research Cambridge, UK

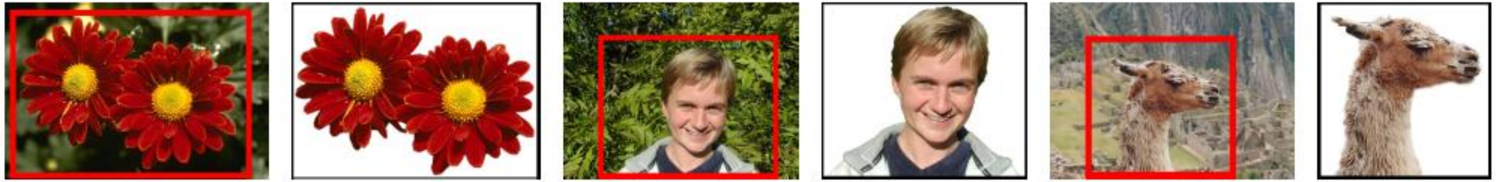


Figure 1: **Three examples of GrabCut .** The user drags a rectangle loosely around an object. The object is then extracted automatically.

Abstract

The problem of efficient, interactive foreground/background segmentation in still images is of great practical importance in image editing. Classical image segmentation tools use either texture (colour) information, e.g. Magic Wand, or edge (contrast) information, e.g. Intelligent Scissors. Recently, an approach based on optimization by graph-cut has been developed which successfully combines both types of information. In this paper we extend the

free of colour bleeding from the source background. In general, degrees of interactive effort range from editing individual pixels, at the labour-intensive extreme, to merely touching foreground and/or background in a few locations.

1.1 Previous approaches to interactive matting

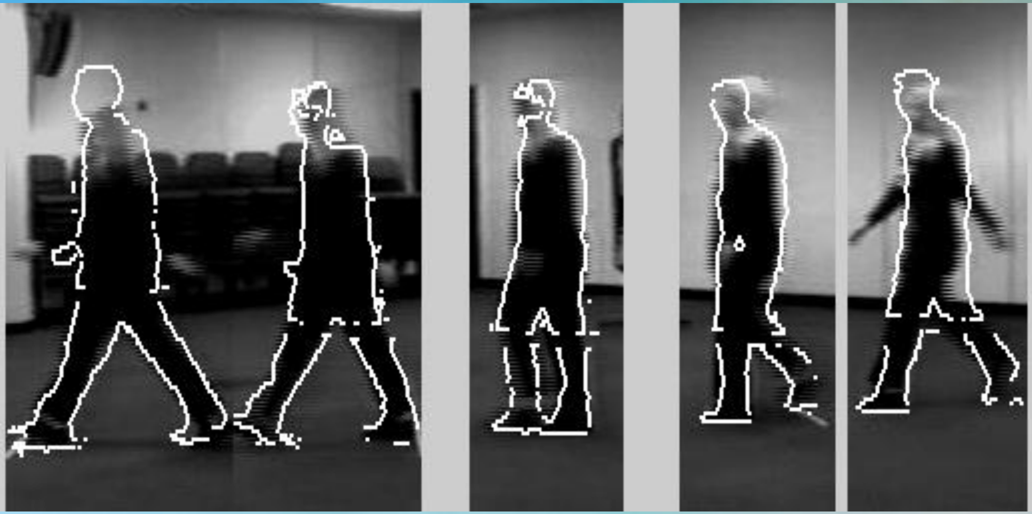
In the following we describe briefly and compare several state of the art interactive tools for segmentation: Magic Wand, Intelligent

Microsoft® Office 2010

F#: Functional Programming goes Mainstream

Don Syme

Human body tracking: Method 1 – search



Andrew Blake, Kentaro Toyama,
Probablisitic tracking in a metric space
Best Paper, IEEE International Conference on Computer Vision, 2001

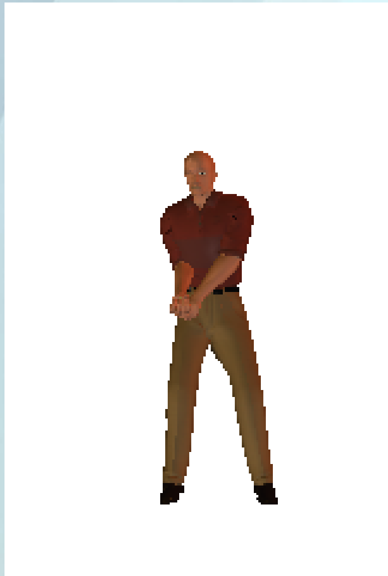
“Search”-based: look up matching exemplar

Human body tracking: Method 2 – regression

Image $\boldsymbol{z}$

$$\theta = f(z)$$

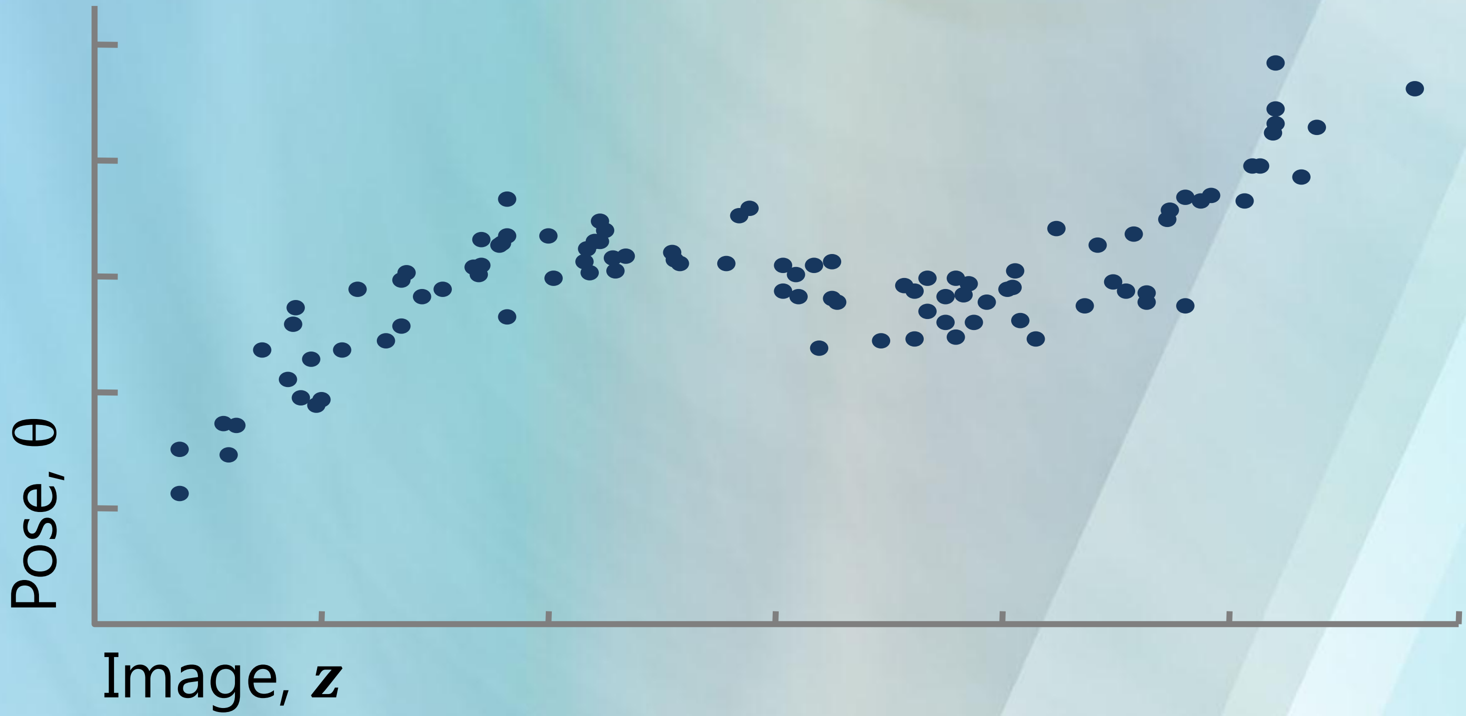
$$f : \mathbb{R}^{100} \mapsto \mathbb{R}^{27}$$



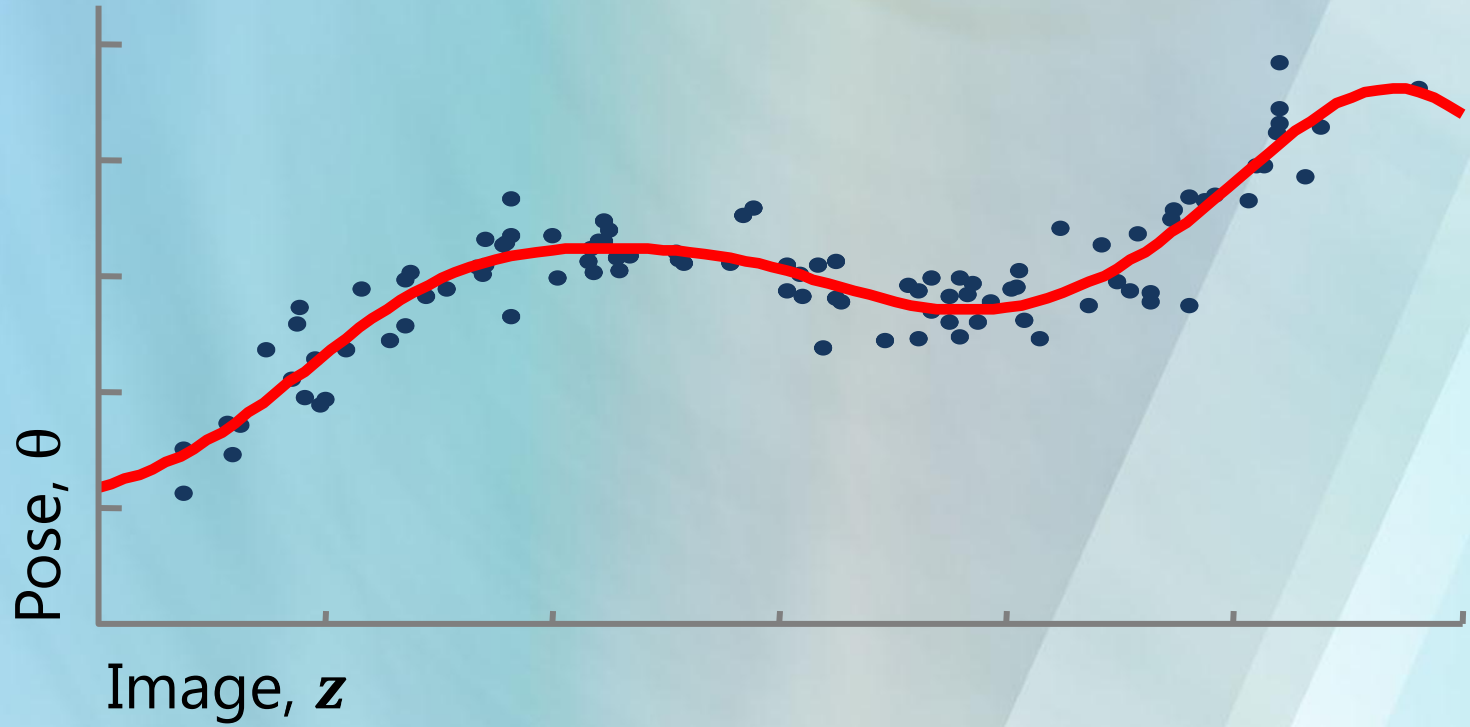
Pose θ

Agarwal & Triggs, CVPR ’04; Urtasun *et al.*, ICCV ’05

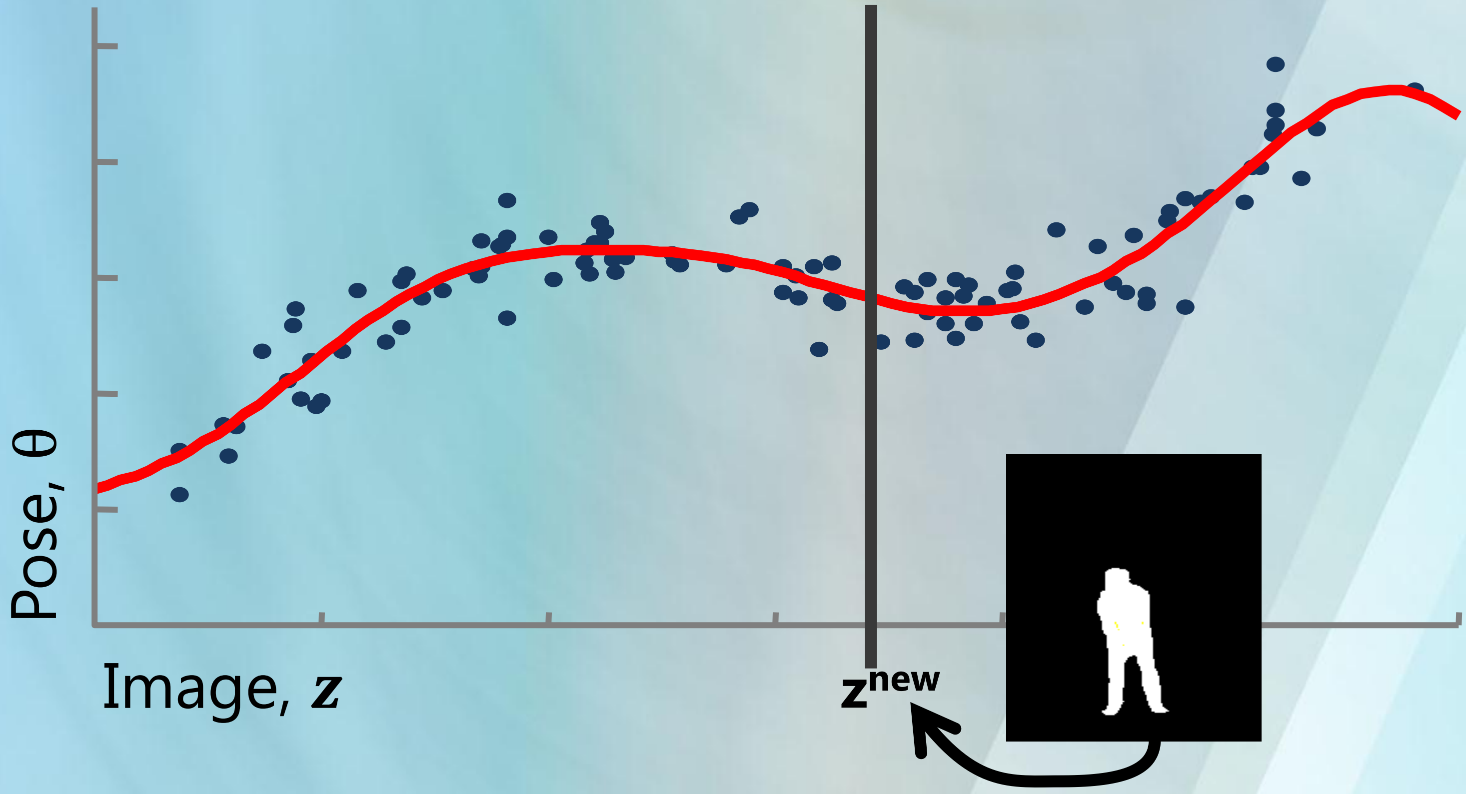
1. Obtain training data $(\boldsymbol{z}_1, \theta_1) \dots (\boldsymbol{z}_n, \theta_n)$



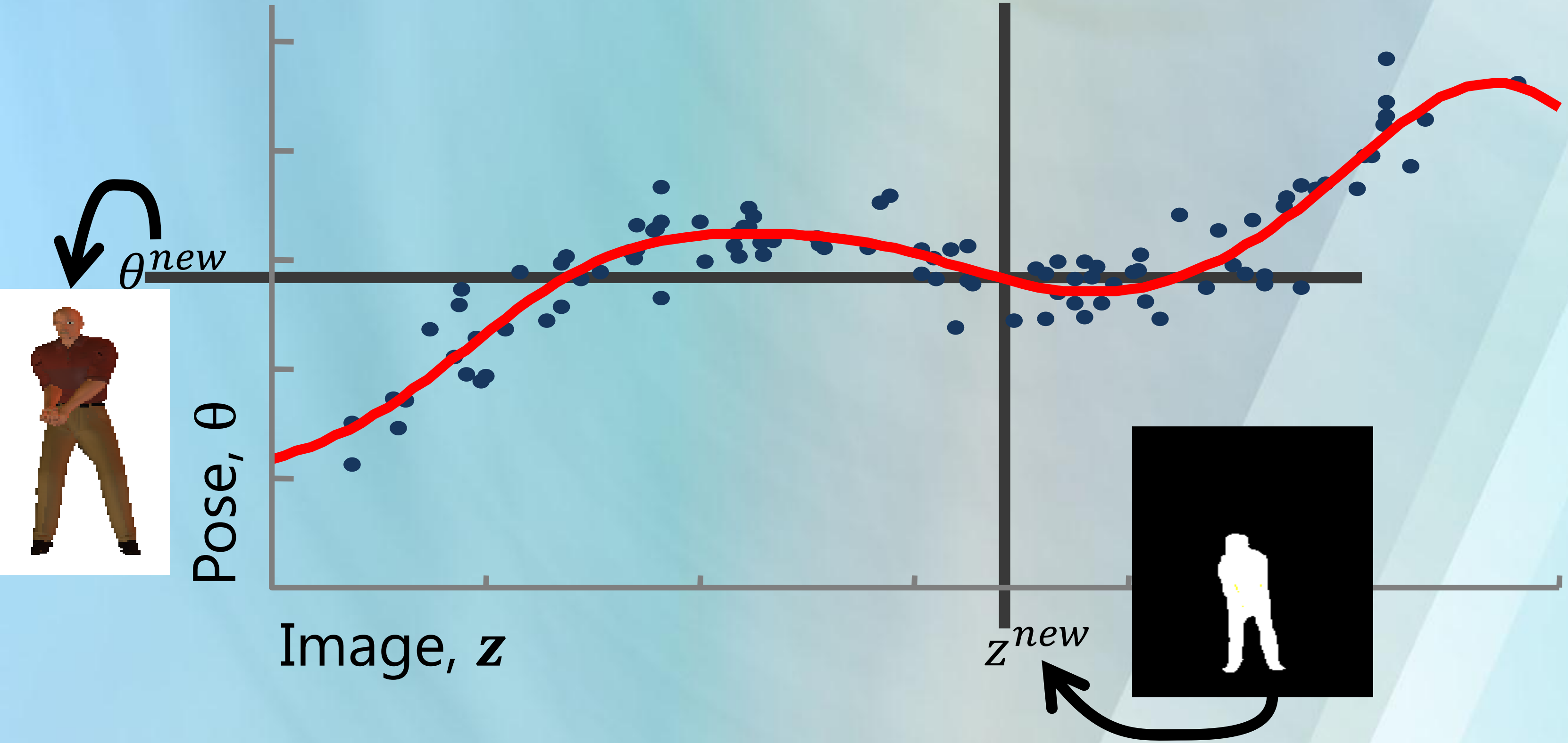
2. Training: Fit function $\theta = f(z)$



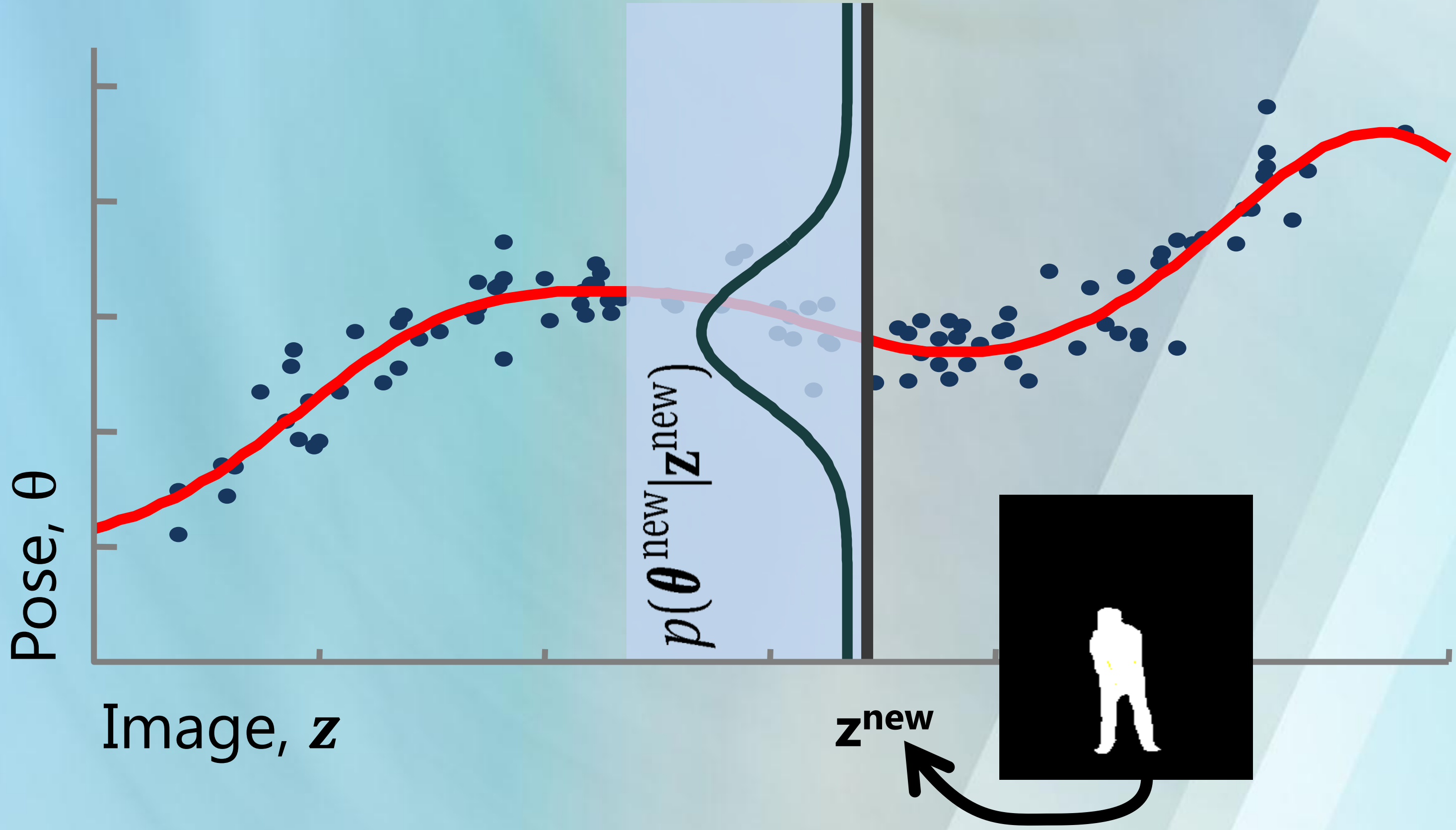
3. Given new image z^{new}, compute $\theta^{new} = f(z^{new})$



3. Given new image z^{new}, compute $\theta^{new} = f(z^{new})$



3. Or, more usefully, compute $p(\theta^{new}|z^{new})$

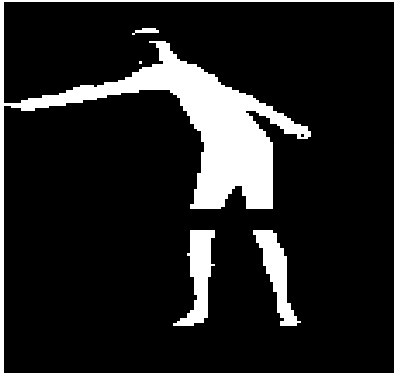


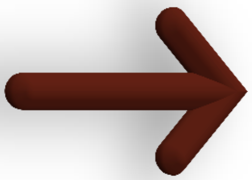
But

Can it ever work?

- f is multivalued
- z and θ high dimensional

Multivalued *f*:

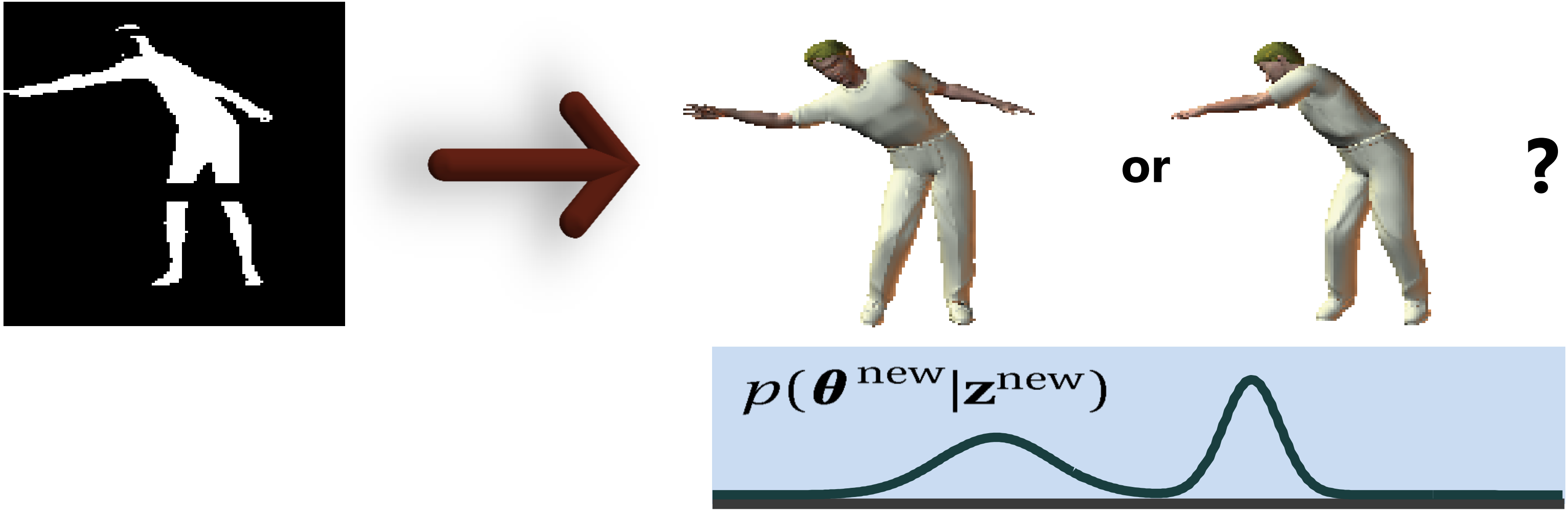




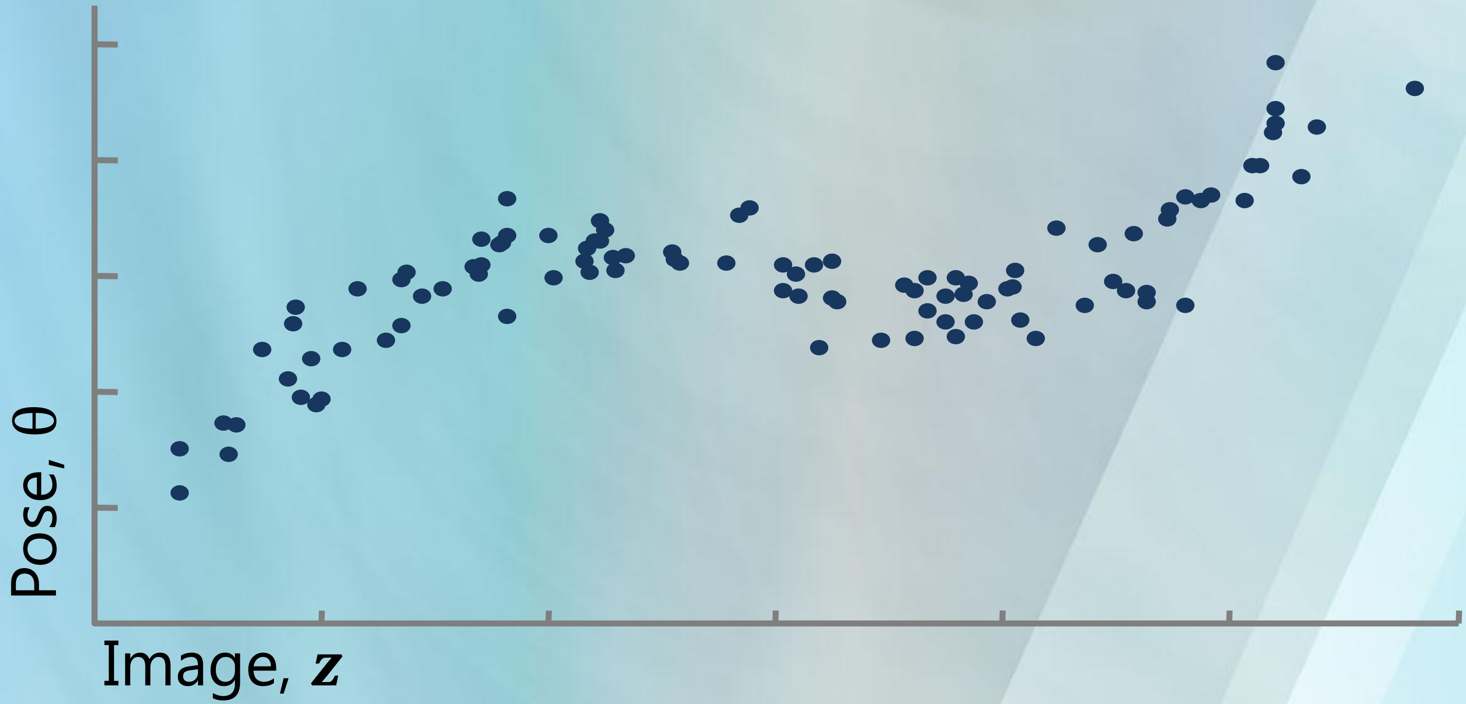
or

?

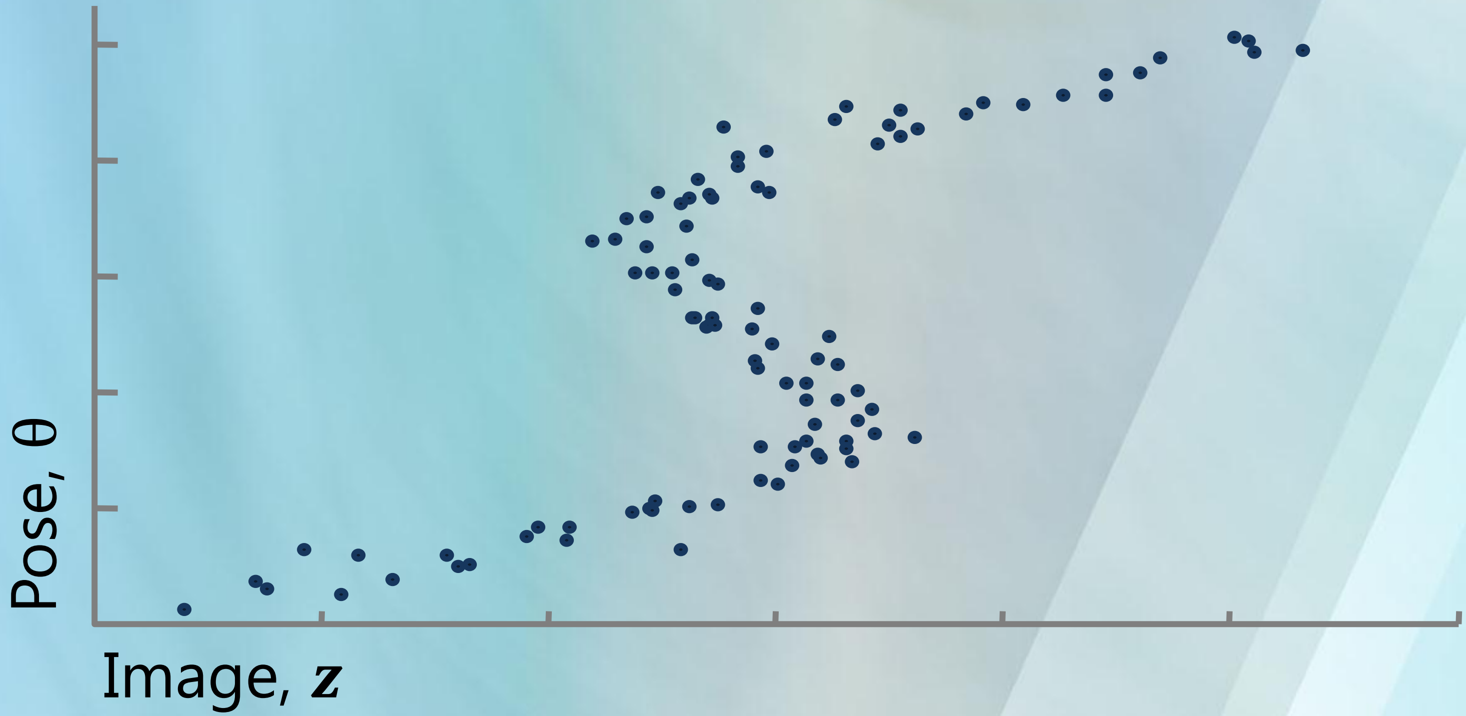
Multivalued f:



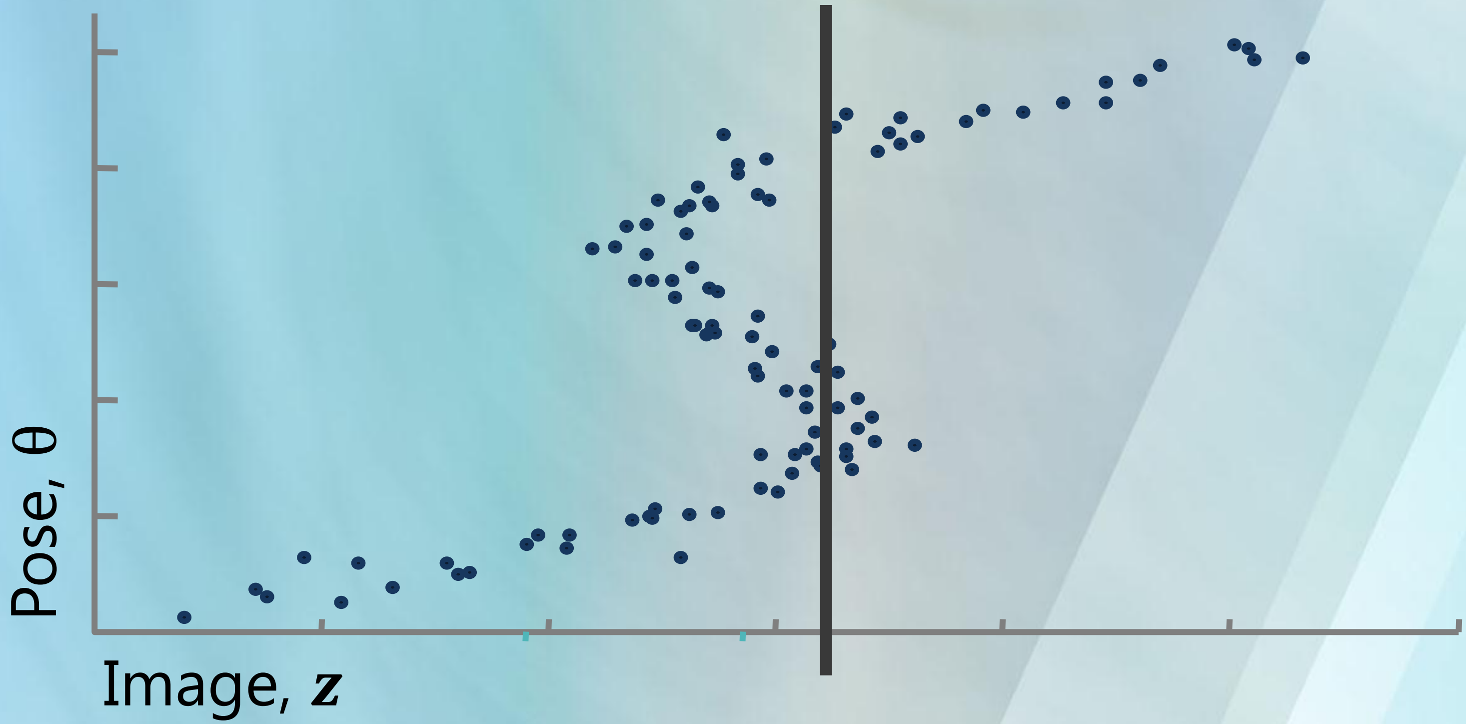
Instead of this:



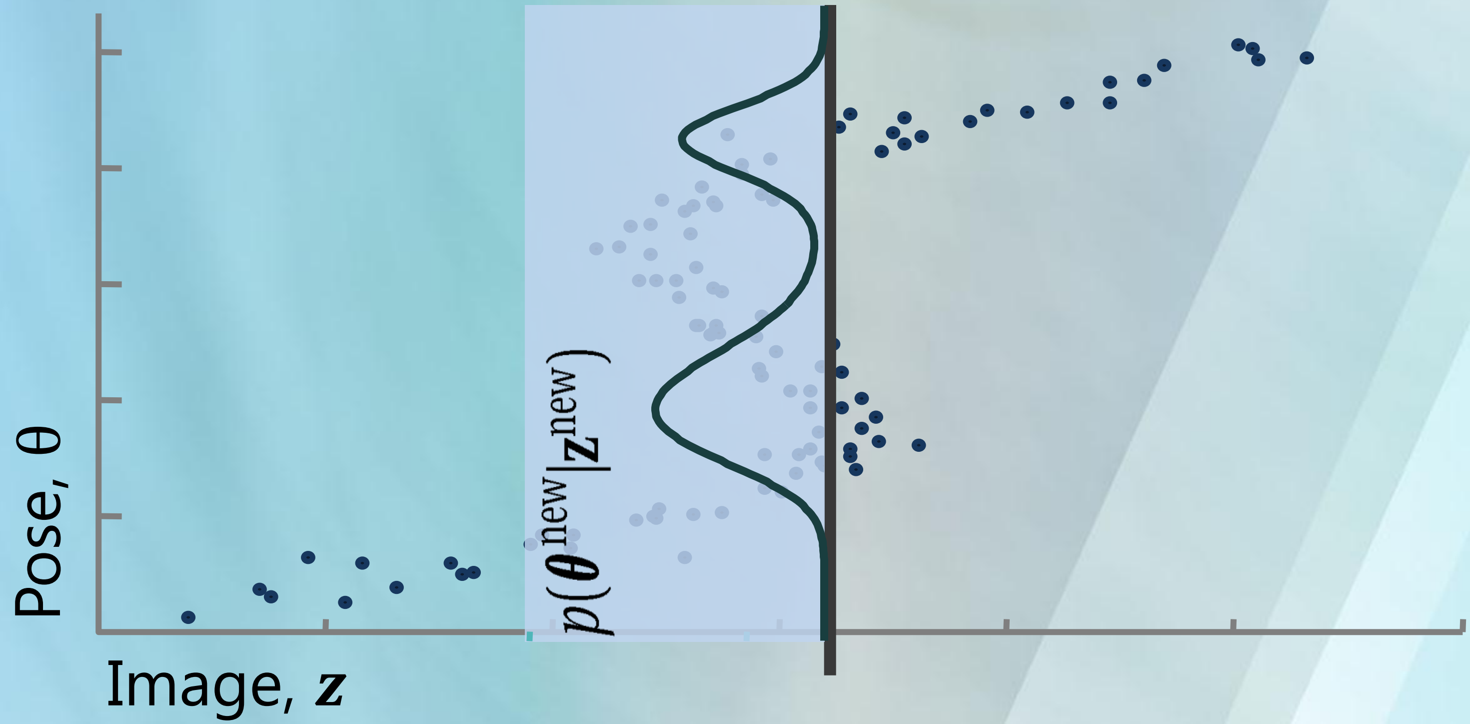
We have this:



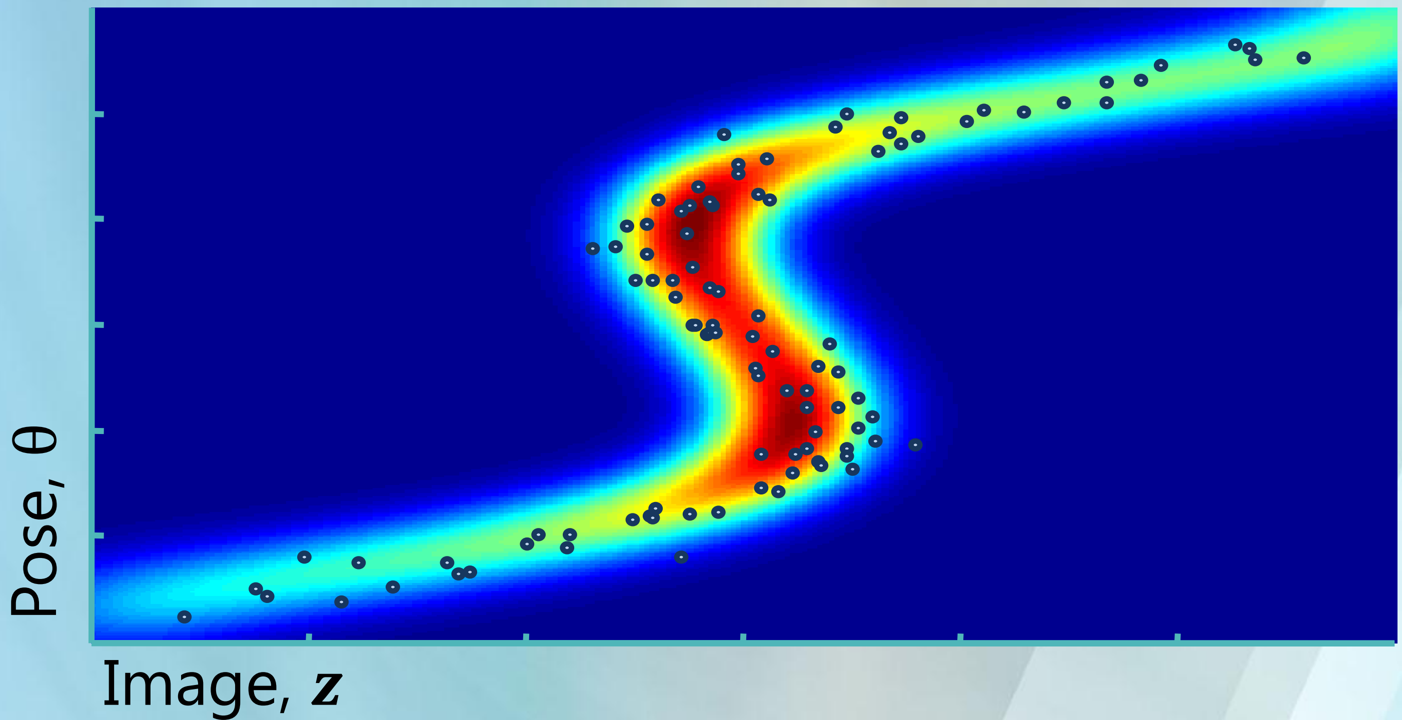
We have this:



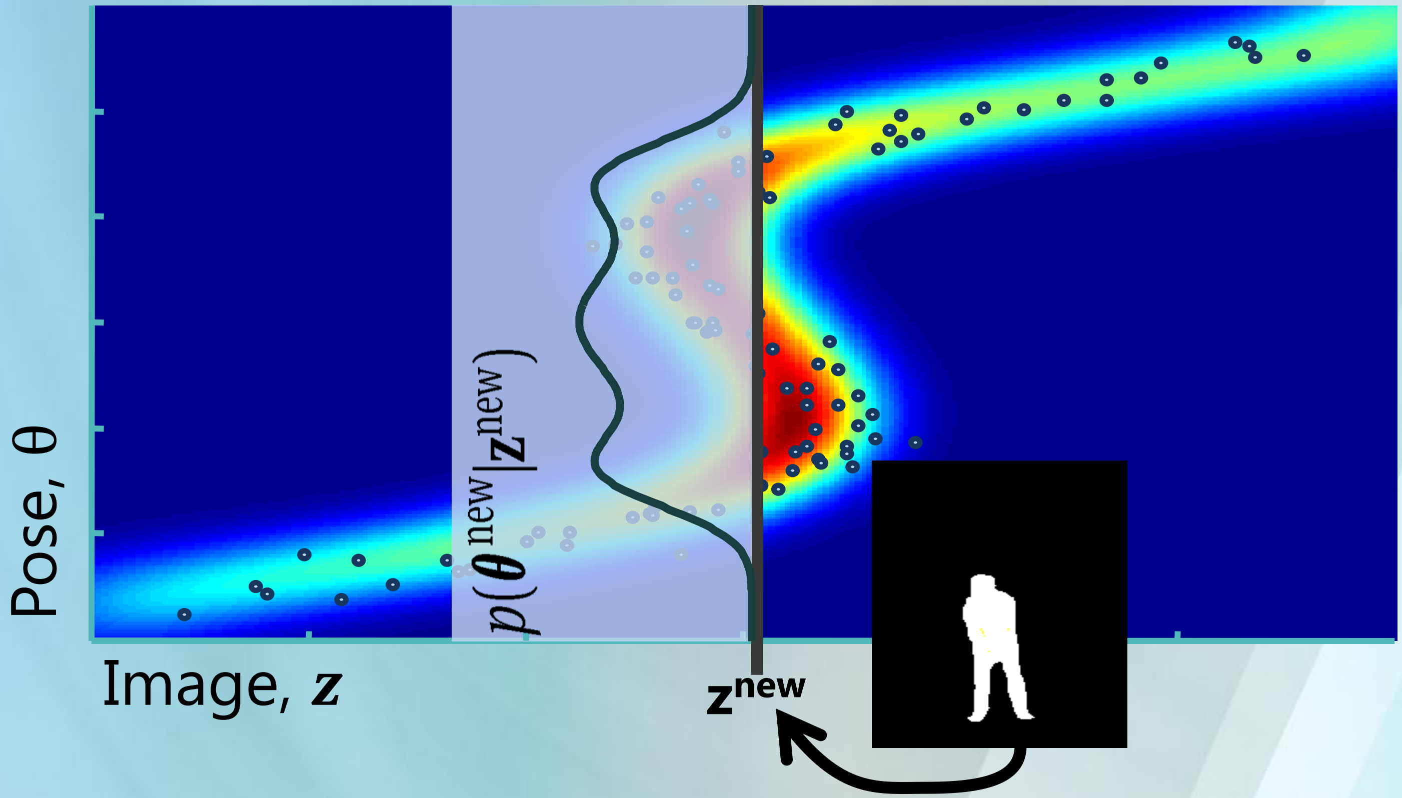
We have this:



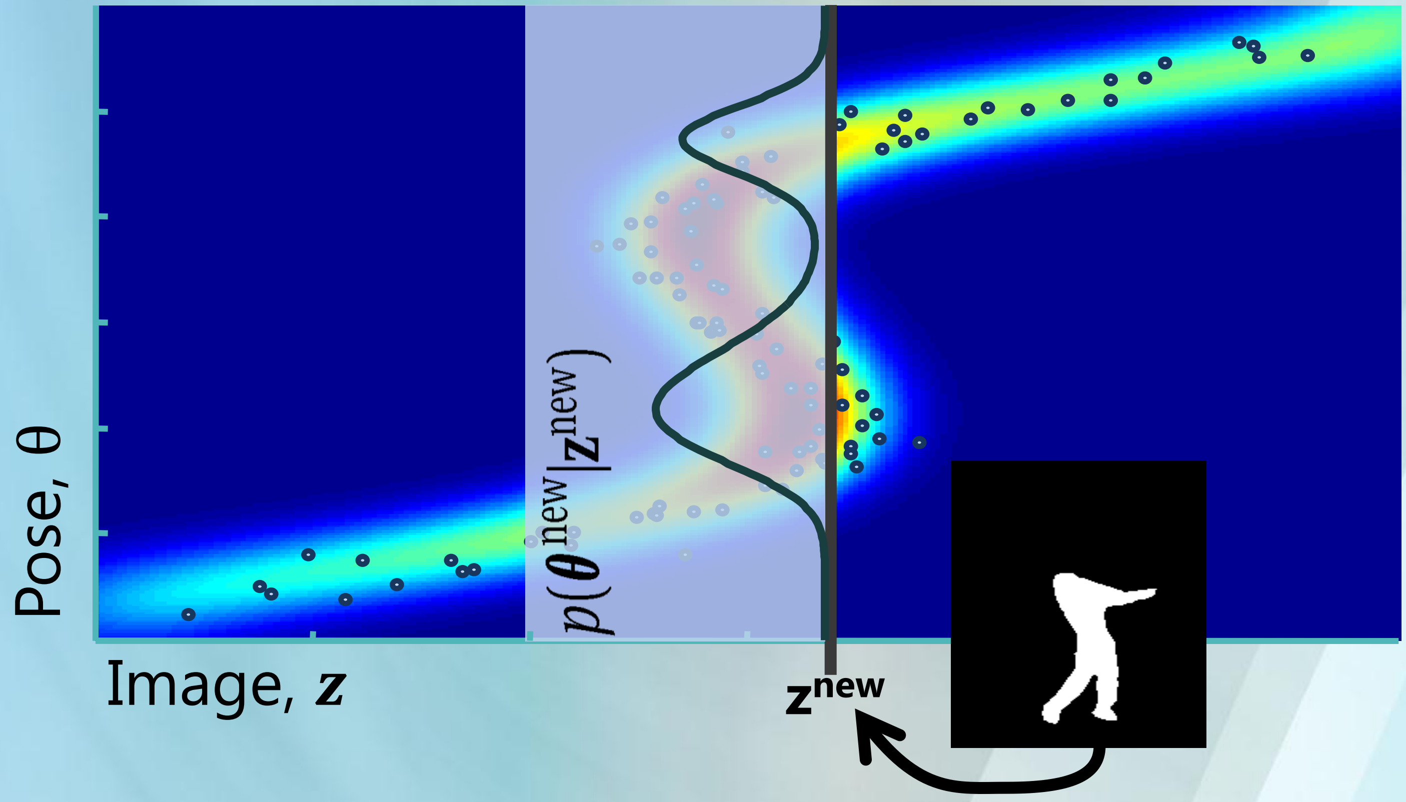
Joint, not conditional: fit $p(\theta, z)$, not $p(\theta|z)$



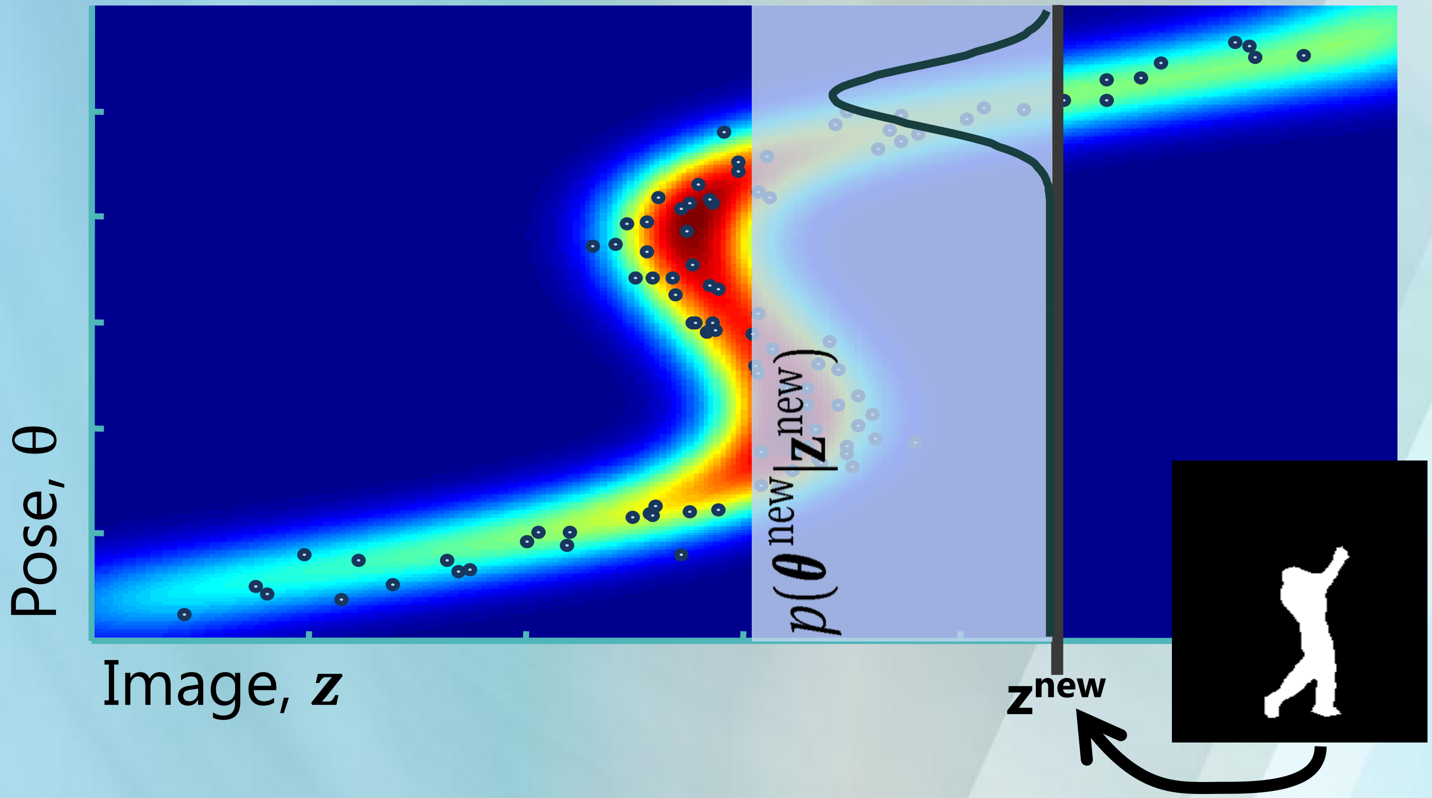
Given z^{new}, compute $p(\theta^{new}, z^{new})$



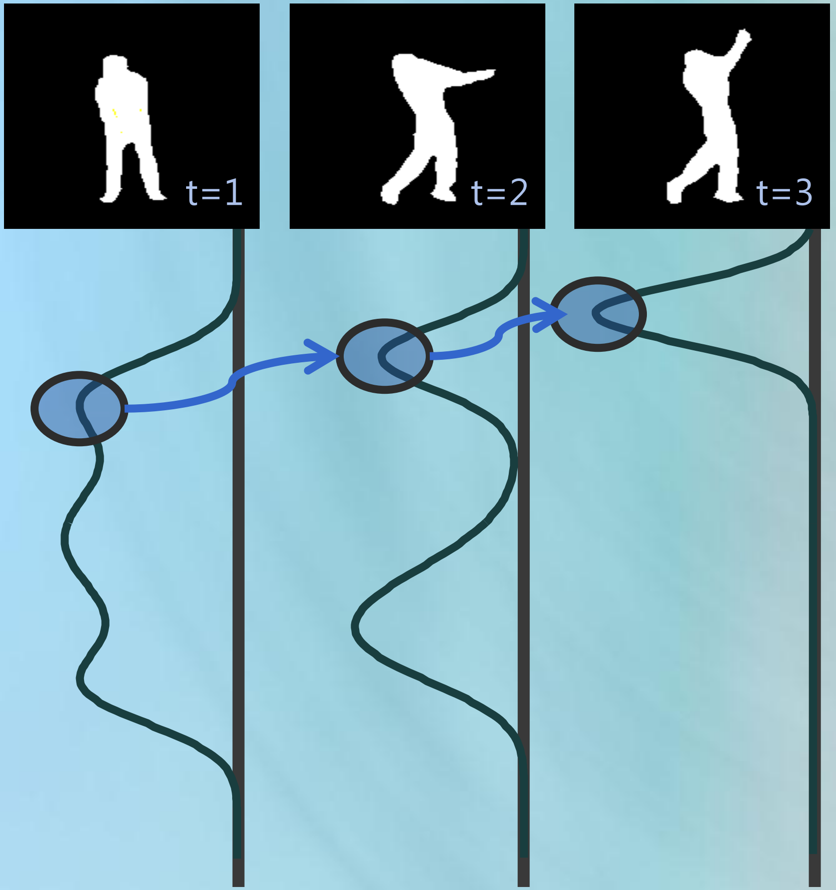
Given z^{new}, compute $p(\theta^{new}, z^{new})$



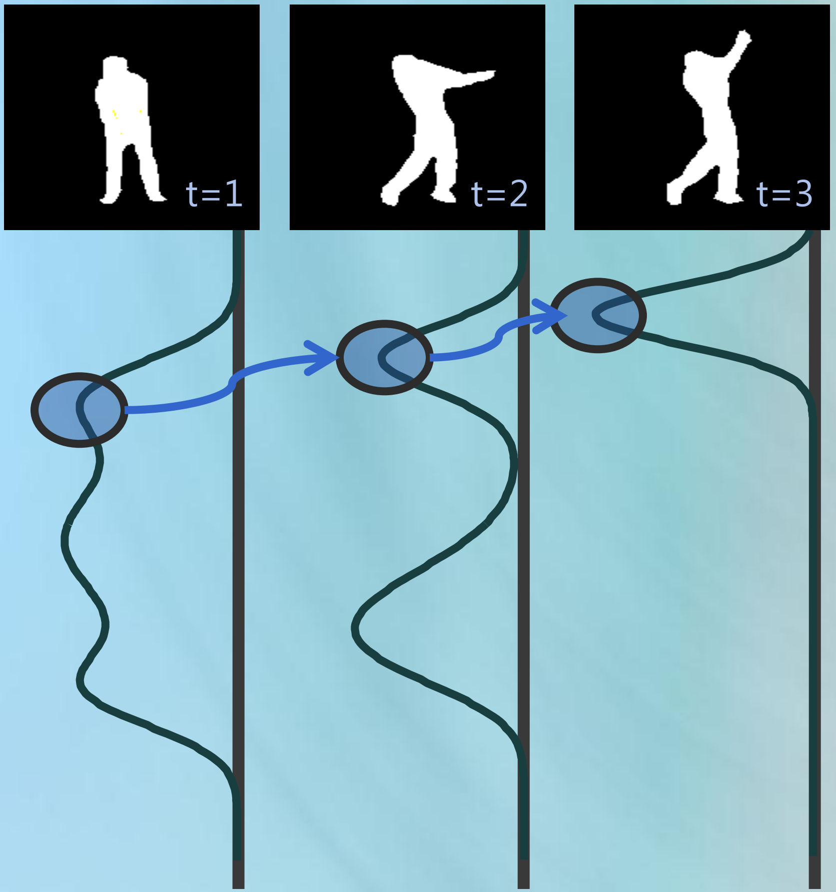
Given z^{new}, compute $p(\theta^{new}, z^{new})$



And filter over time…

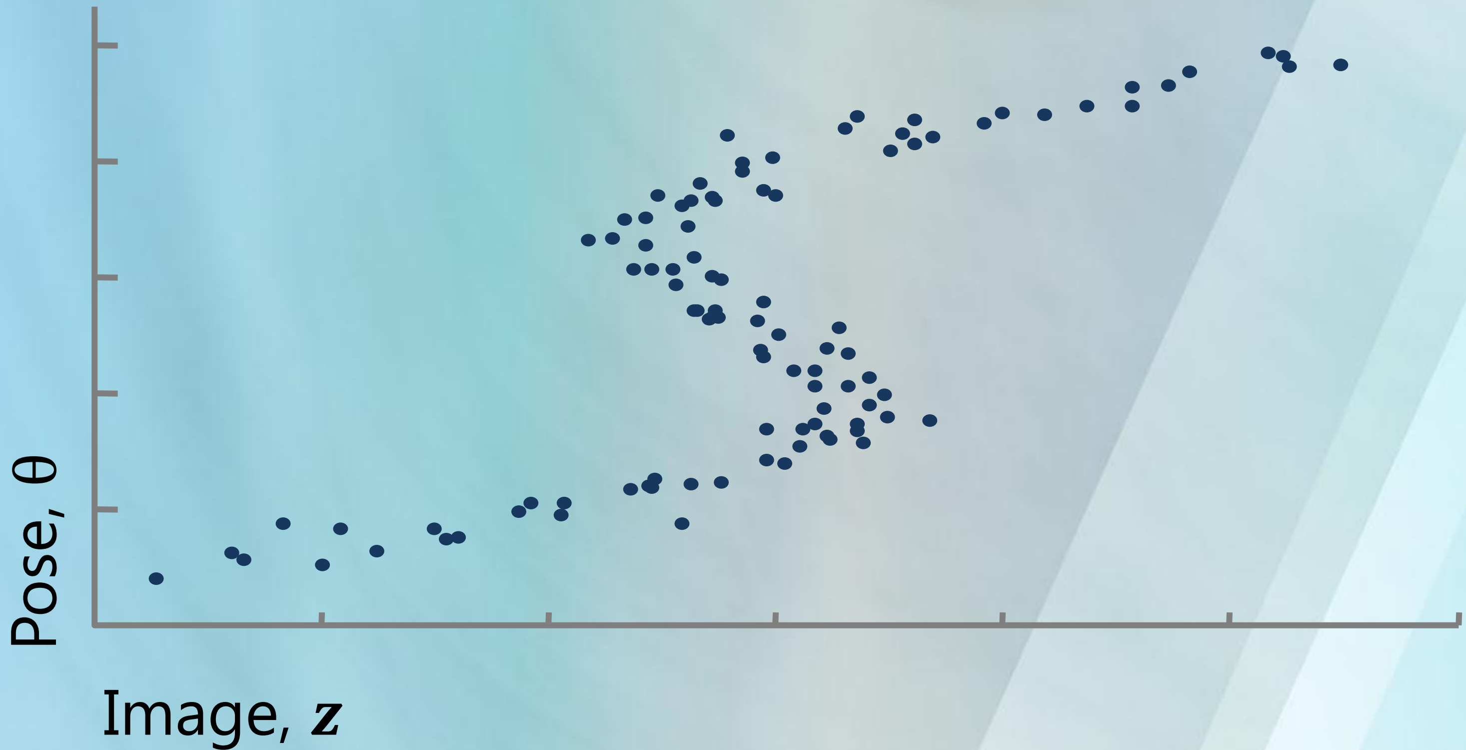


And filter over time…

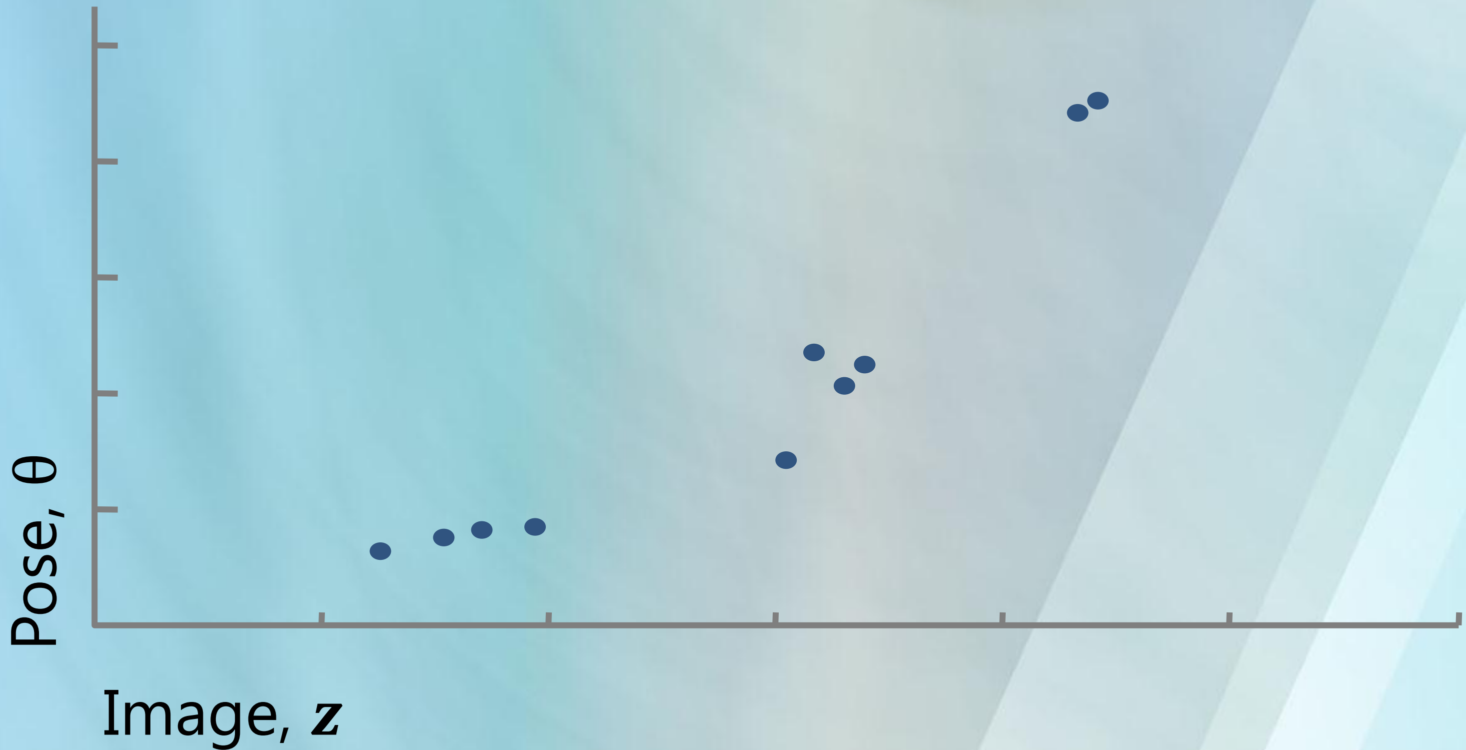


but…

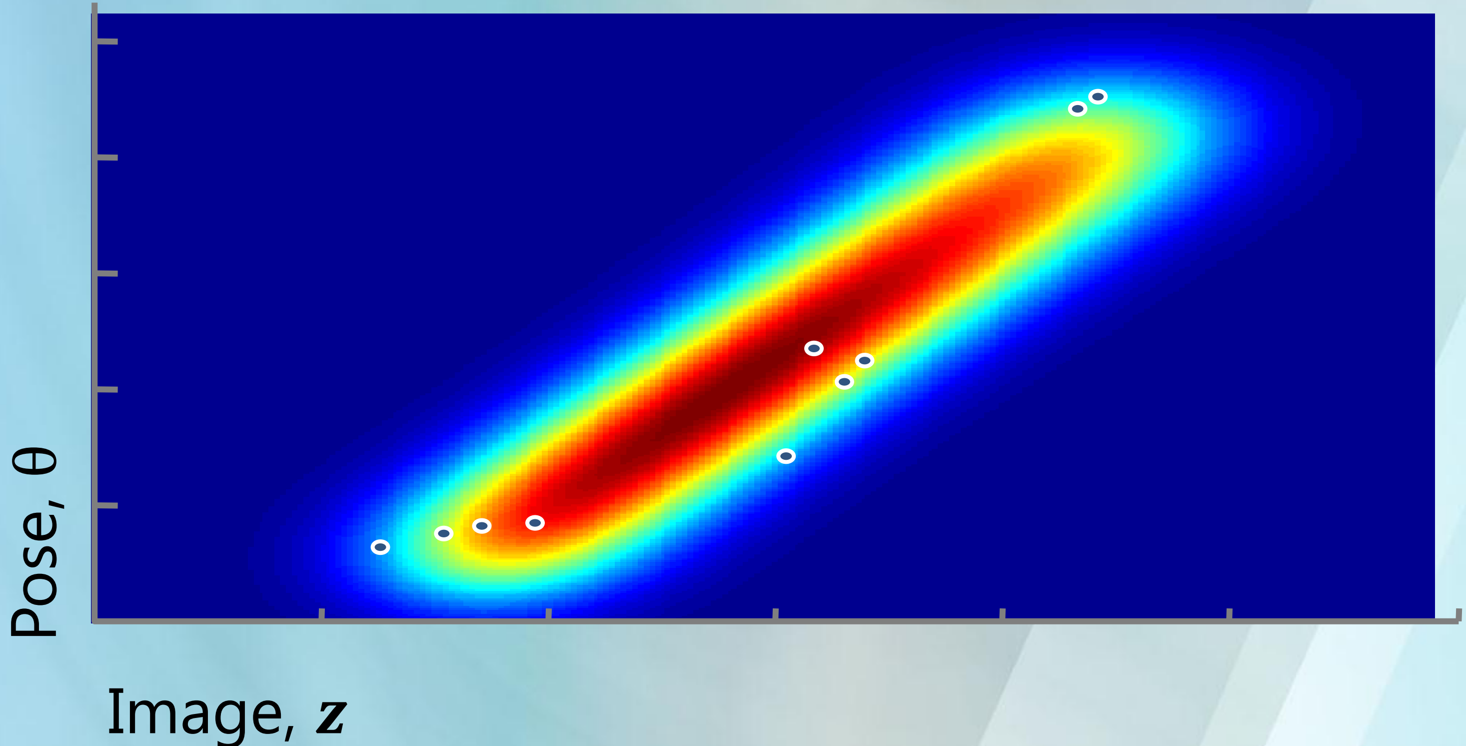
We don't have this:



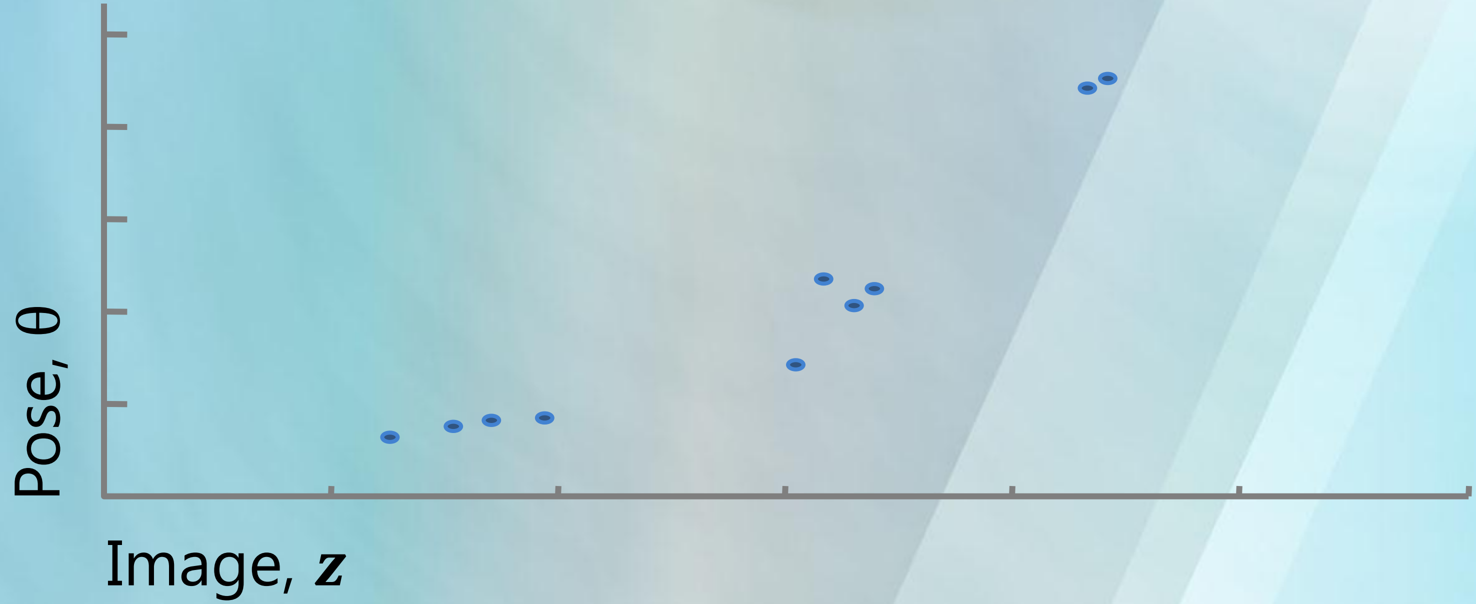
We have this:



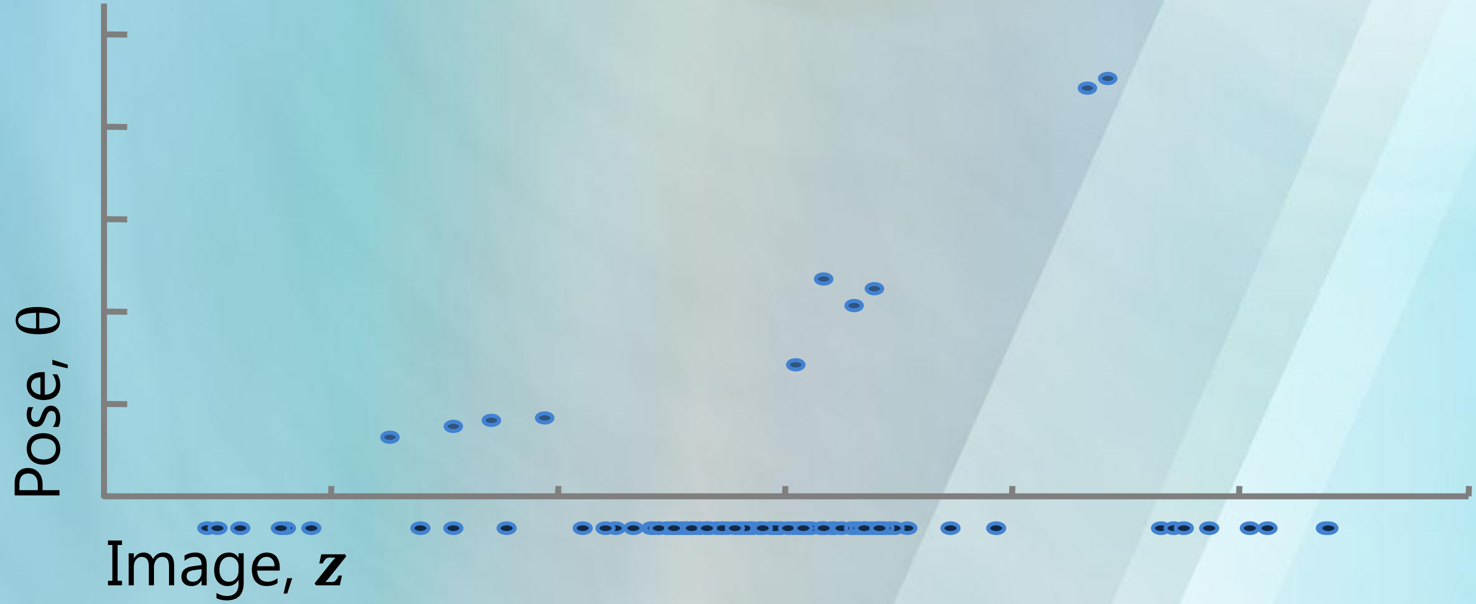
Of which a not unreasonable model is:



We have too few labelled (z, θ) pairs

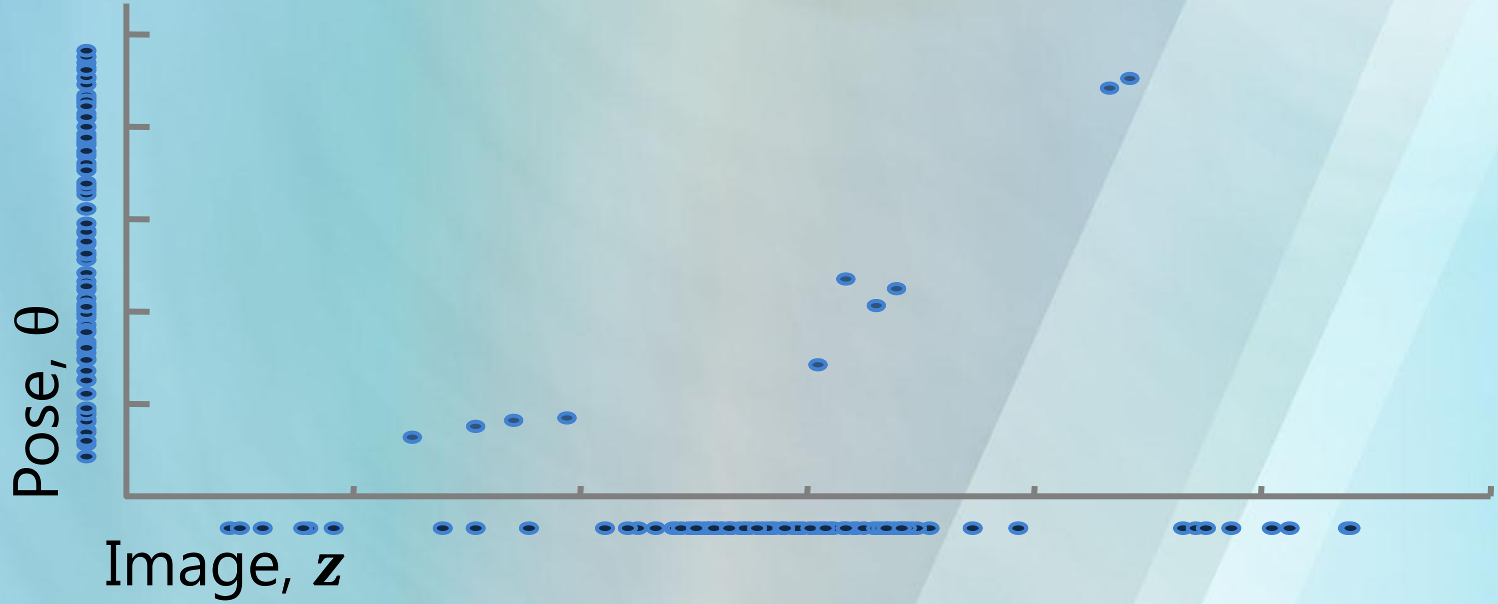


We have too few labelled (z, θ) pairs



But we can easily capture more ***unlabelled*** images, i.e. $(\boldsymbol{z}, *)$ pairs

We have too few labelled (z, θ) pairs



And we can easily obtain more motion capture data i.e. more $(*, \theta)$ pairs

Marginal statistics

$p(\boldsymbol{\theta}) = \int p(\boldsymbol{z}, \boldsymbol{\theta}) d\boldsymbol{z}$

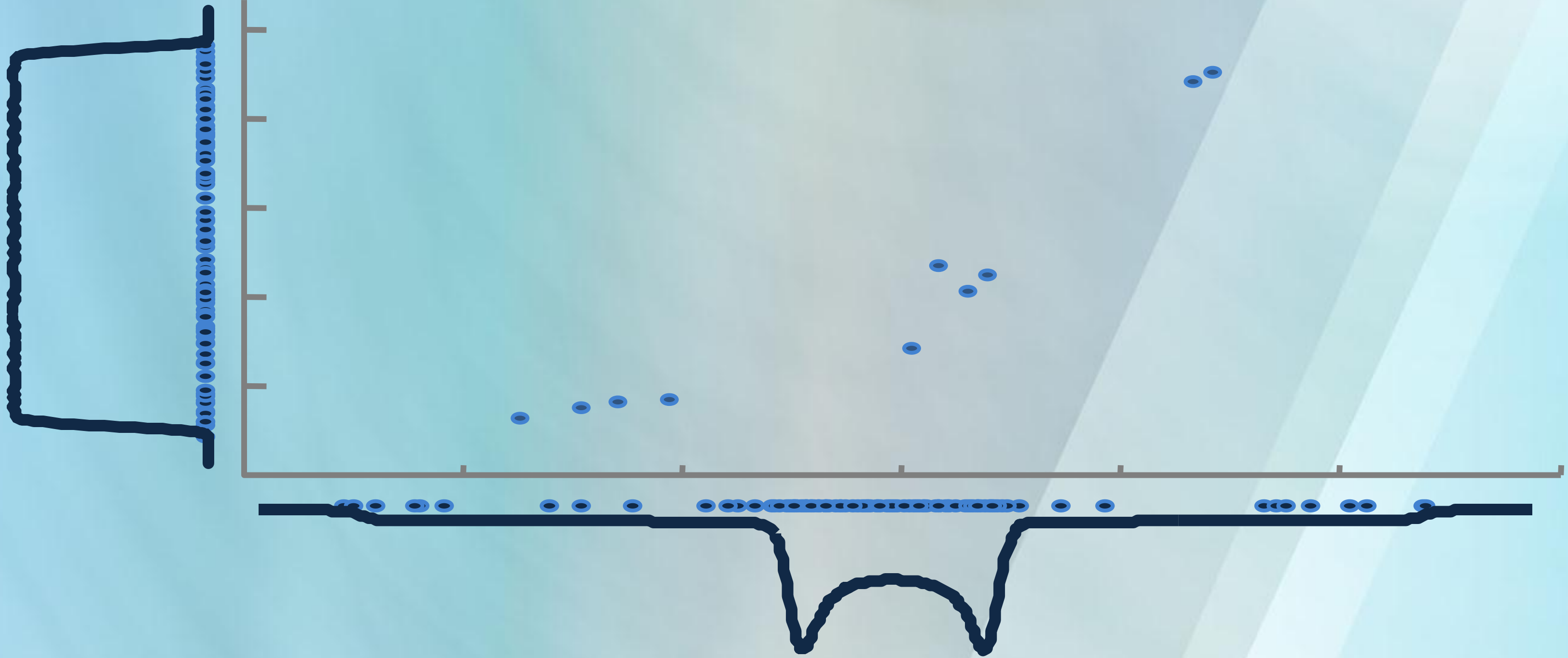
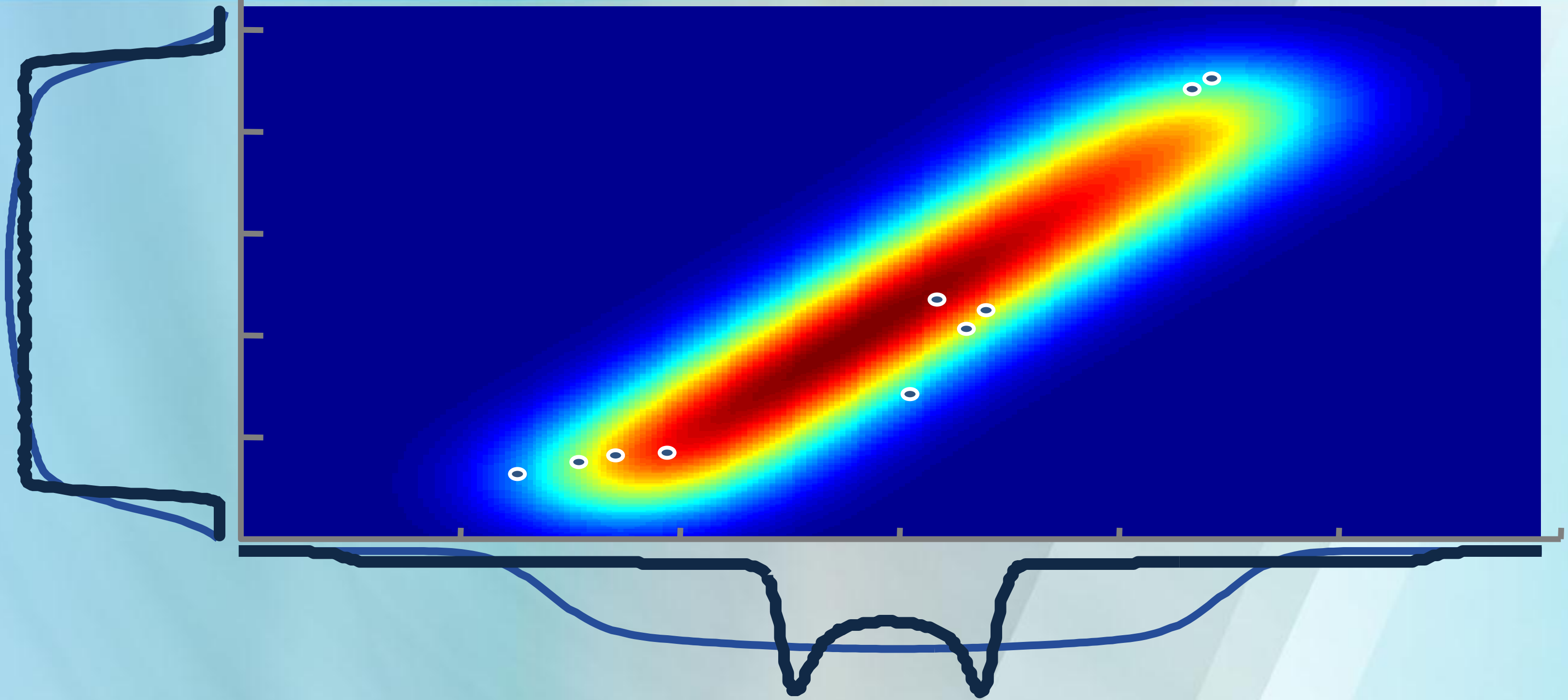
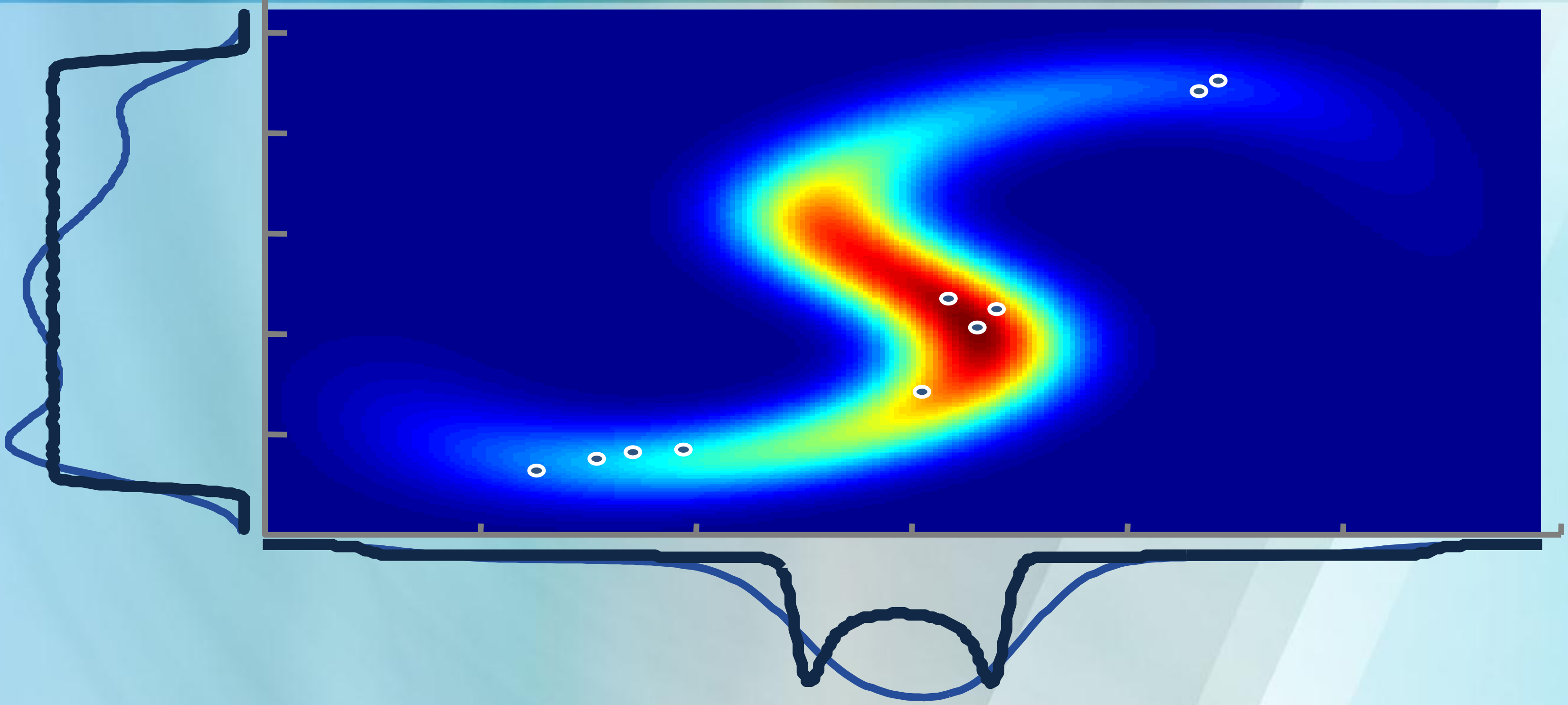


Image marginal, $p(\boldsymbol{z}) = \int p(\boldsymbol{z}, \boldsymbol{\theta}) d\boldsymbol{\theta}$

Marginal statistics *which contradict our earlier guess*

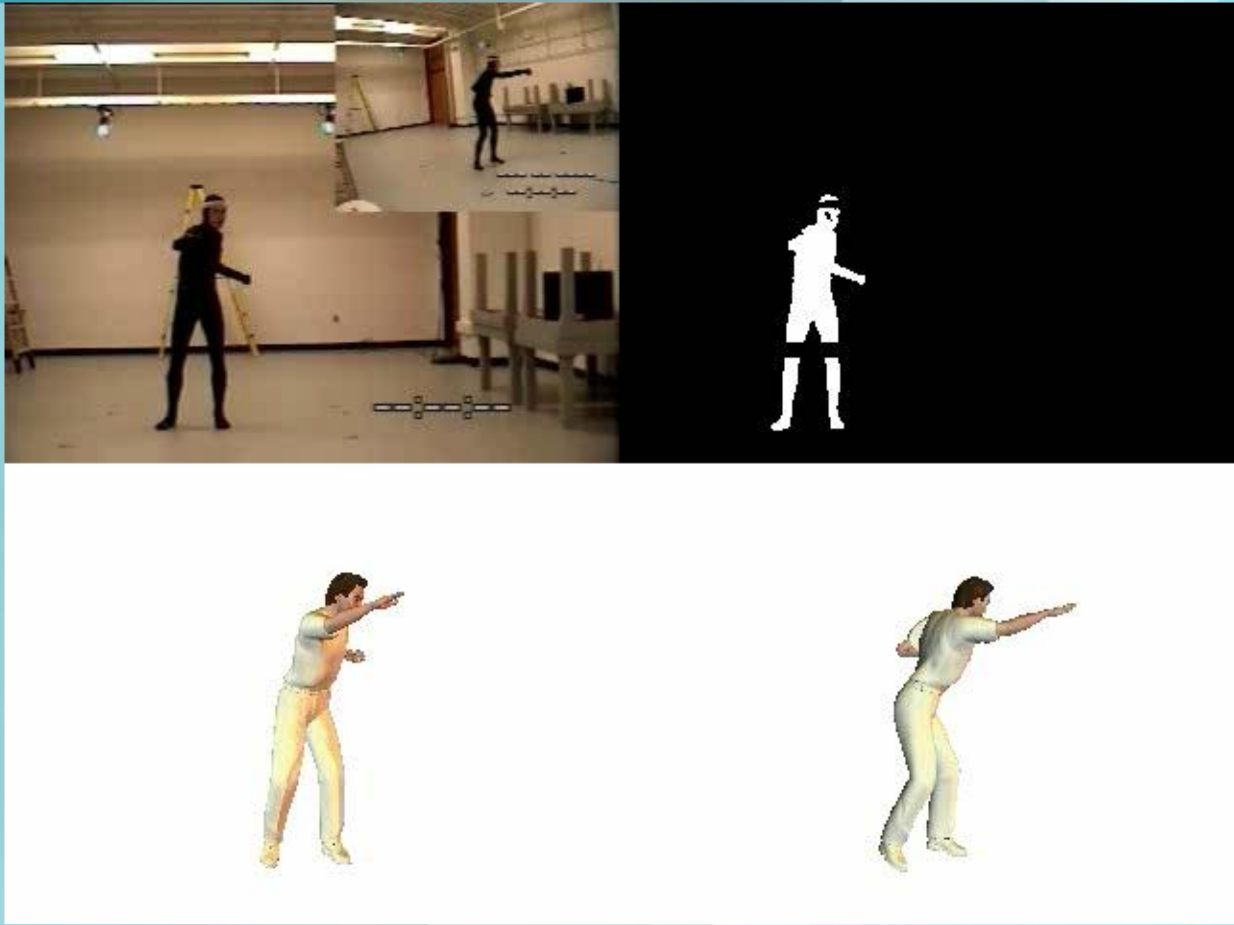


Requiring consistent marginals gives this:



Research: Human body tracking

- ☑ Wide range of motion
- ☒ But limited agility
- ☒ And not realtime



R Navaratnam, A Fitzgibbon, R Cipolla
The Joint Manifold Model for Semi-supervised Multi-valued Regression
IEEE Intl Conf on Computer Vision, 2007

The Call: September 2008

"We need a body tracker with

- ☑ All motions…
- ☑ All agilities…
- ☑ 10x Realtime…
- ☑ For multiple players…

The Call: September 2008

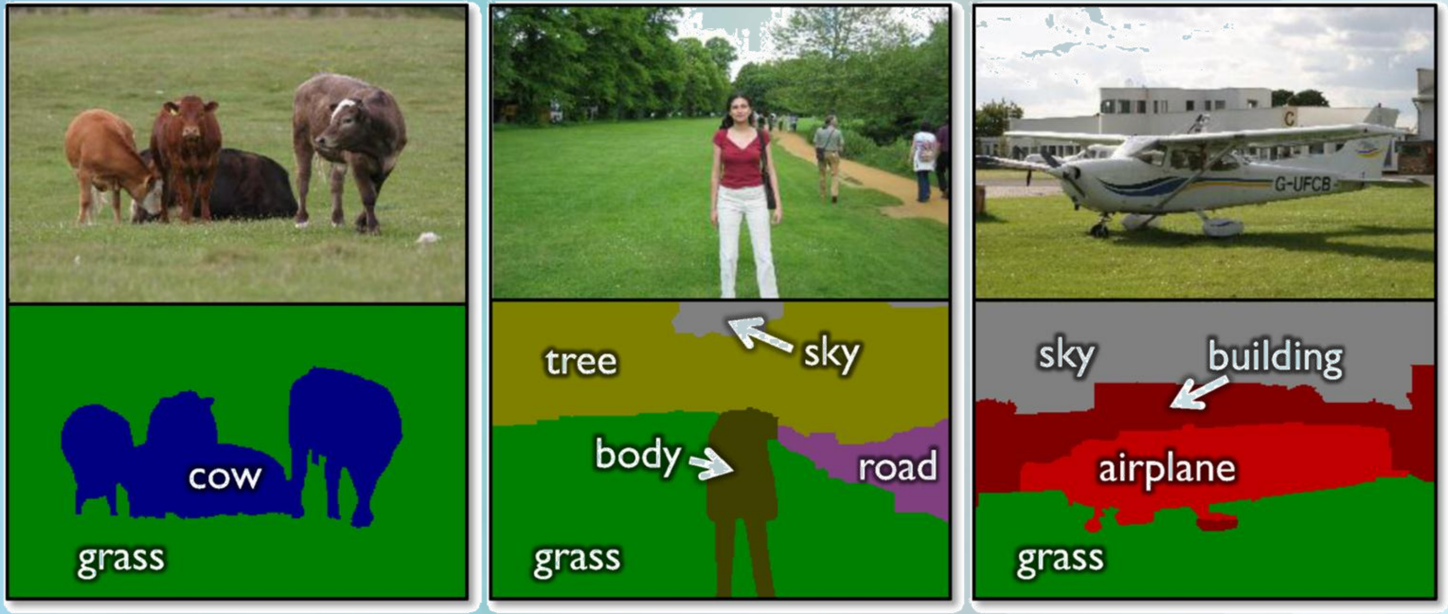
"We need a body tracker with
☑ All motions…
☑ All agilities…
☑ 10x Realtime…
☑ For multiple players…

… but you have got 3D ☺"

Step 1: Collect training data

- Teams visit households across the globe, filming real users
- Hollywood motion capture studio generates billions of CG images
- Researchers and devs think…

Aside: Object Recognition



object classes	building	grass	tree	cow	sheep	sky	airplane	water	face	car
bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat

J. Shotton, J. Winn, C. Rother, A. Criminisi, ***TextonBoost*: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation.** European Conference on Computer Vision, 2006

Aside: Object Recognition

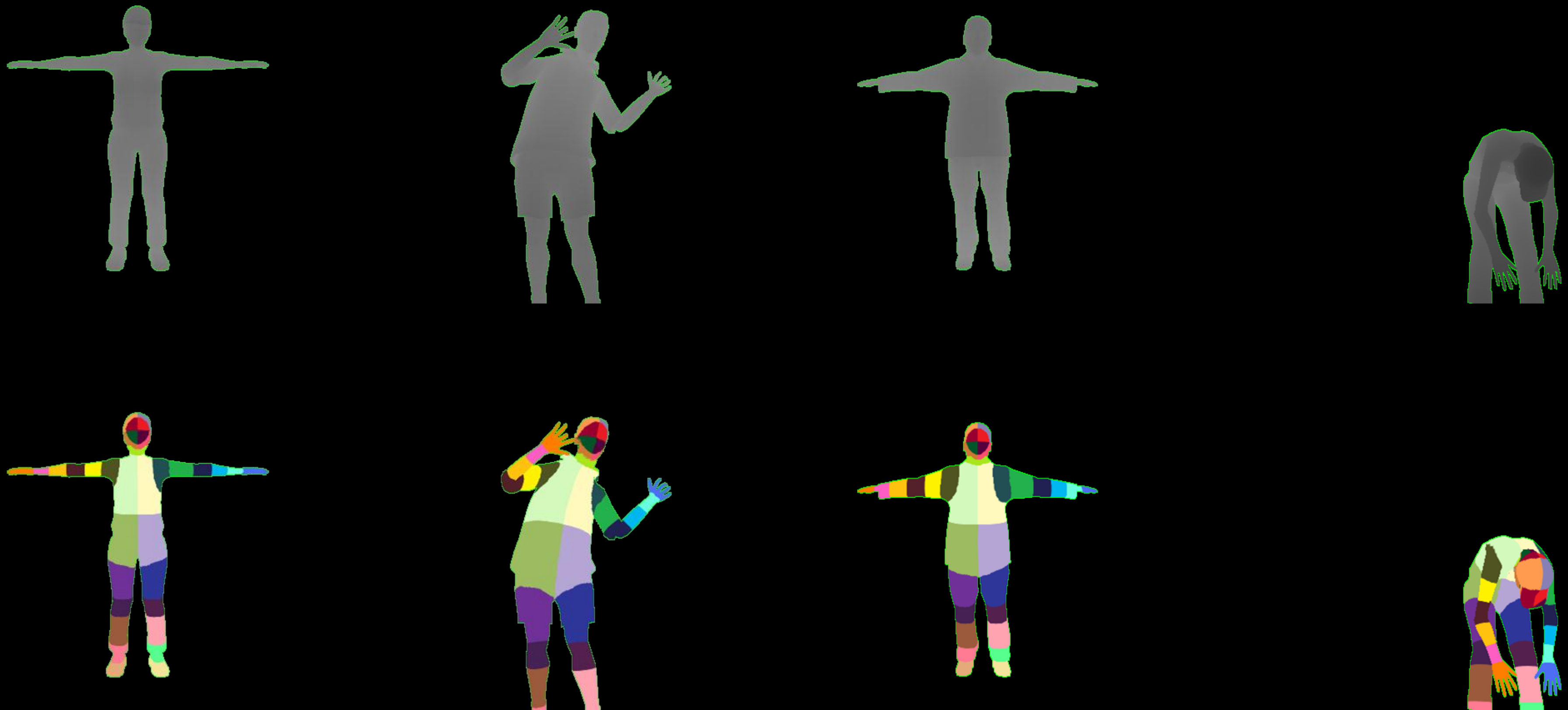
Real-Time Semantic Segmentation

Jamie Shotton
Matthew Johnson
Roberto Cipolla

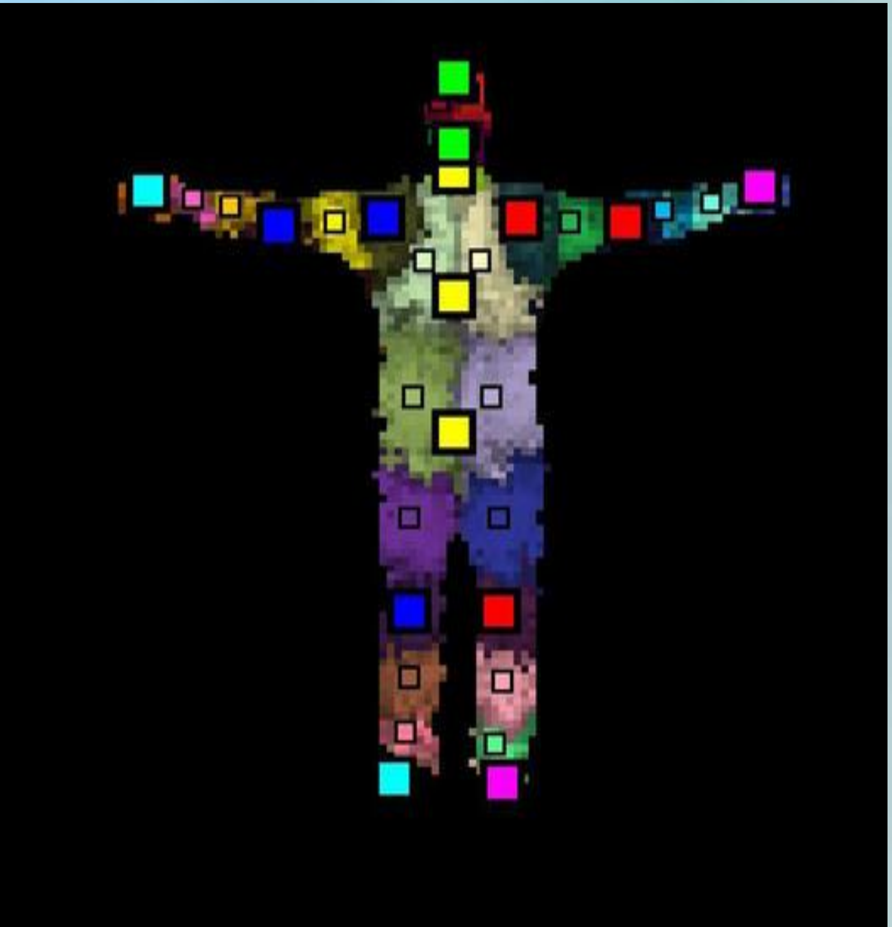
Step 1: Collect training data

- Teams visit households across the globe, filming real users
- Hollywood motion capture studio generates billions of CG images
- Researchers and devs think… if only I had a hammer.

Training data

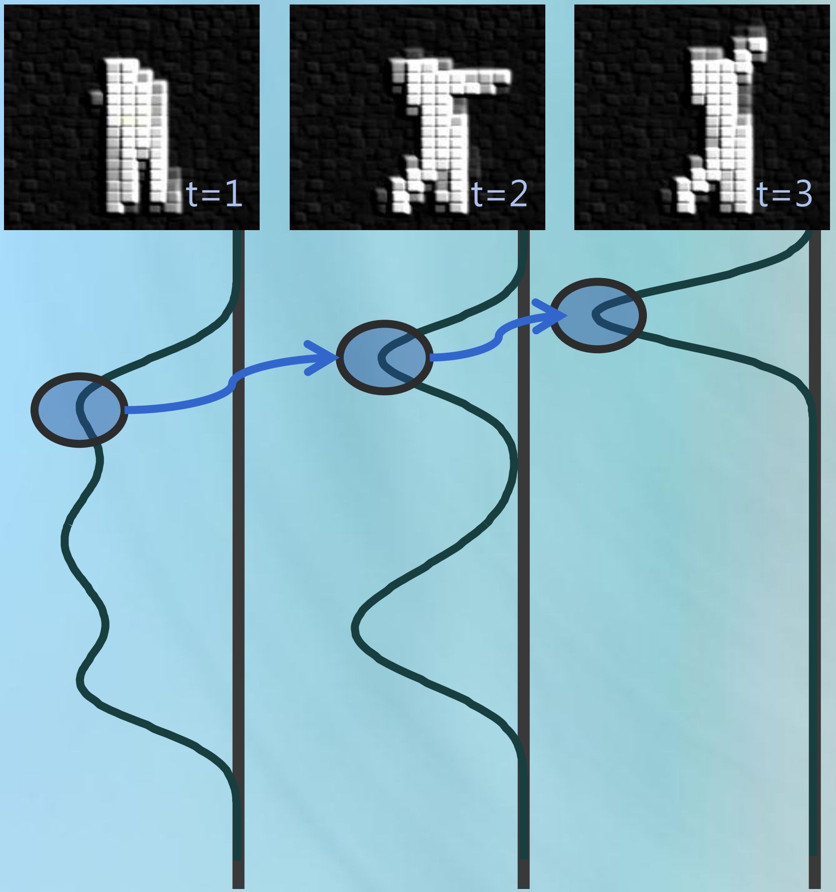


How it works



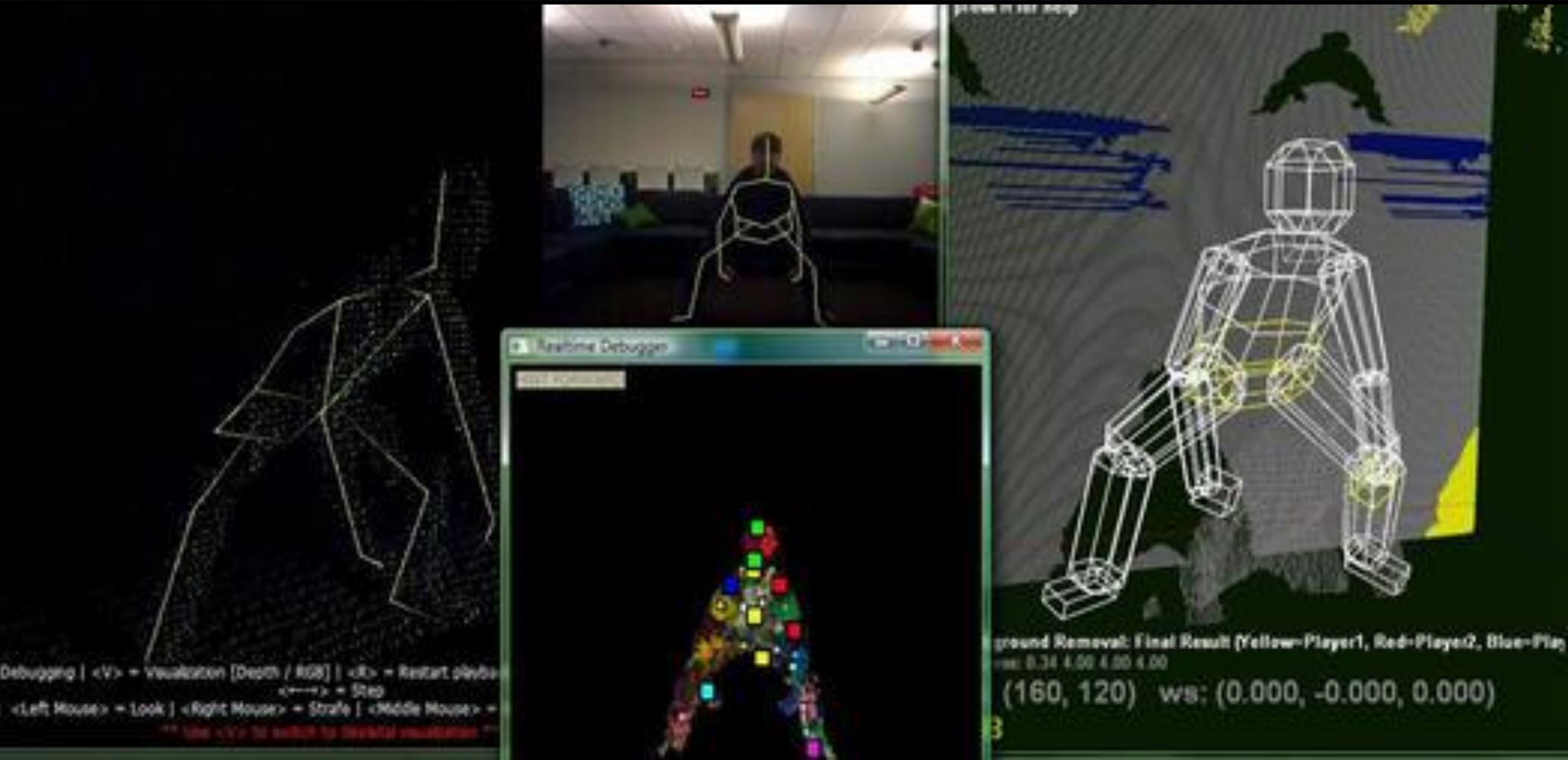
1. Classify each pixel's probability of being each of 32 body parts
2. Determine probabilistic cluster of body configurations consistent with those parts
3. Present the most probable to the user

How it works



1. Classify each pixel's probability of being each of 32 body parts
2. Determine probabilistic cluster of body configurations consistent with those parts
3. Present the most probable to the user

Under the bonnet



Training

Millions of training images -> millions of classifier parameters

- Very far from “embarrassingly parallel”
- New algorithm for distributed decision-tree training
- Major use of DryadLINQ [available for download]

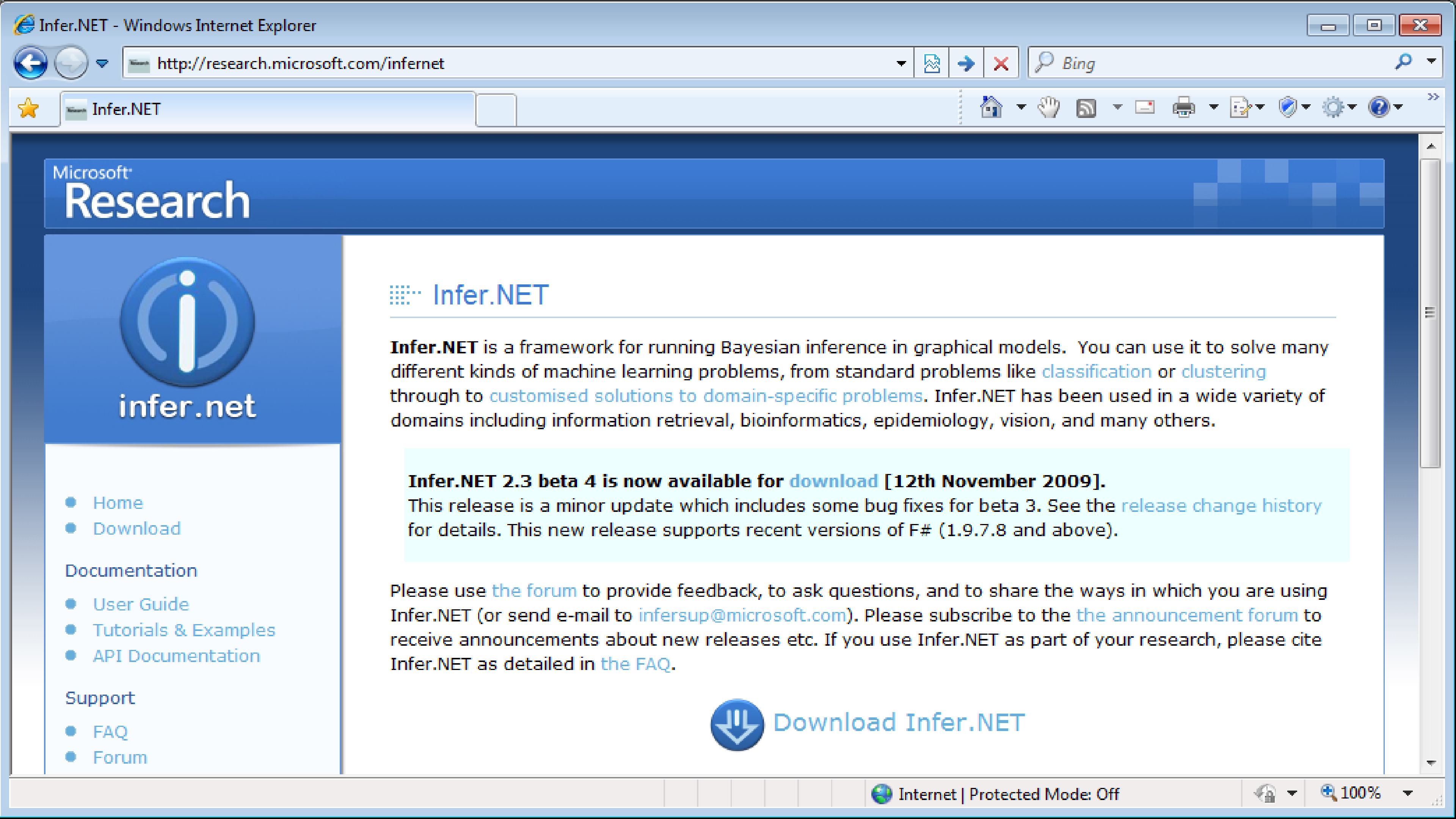
Distributed Data-Parallel Computing Using a High-Level Programming Language
M Isard, Y Yu
International Conference on Management of Data (SIGMOD), July 2009

Conclusions

Machine learning loves hard problems

Games programmers are amazing

Blue skies research can be quickest to market



Infer.NET - Windows Internet Explorer
http://research.microsoft.com/infernet
Bing
Infer.NET
Internet | Protected Mode: Off
100%
Microsoft Research
infer.net
Home
Download
Documentation
User Guide
Tutorials & Examples
API Documentation
Support
FAQ
Forum
Infer.NET
Infer.NET is a framework for running Bayesian inference in graphical models. You can use it to solve many different kinds of machine learning problems, from standard problems like classification or clustering through to customised solutions to domain-specific problems. Infer.NET has been used in a wide variety of domains including information retrieval, bioinformatics, epidemiology, vision, and many others.
Infer.NET 2.3 beta 4 is now available for download [12th November 2009].
This release is a minor update which includes some bug fixes for beta 3. See the release change history for details. This new release supports recent versions of F# (1.9.7.8 and above).
Please use the forum to provide feedback, to ask questions, and to share the ways in which you are using Infer.NET (or send e-mail to infersup@microsoft.com). Please subscribe to the the announcement forum to receive announcements about new releases etc. If you use Infer.NET as part of your research, please cite Infer.NET as detailed in the FAQ.
Download Infer.NET

KINECT™
for XBOX 360.

Q&A