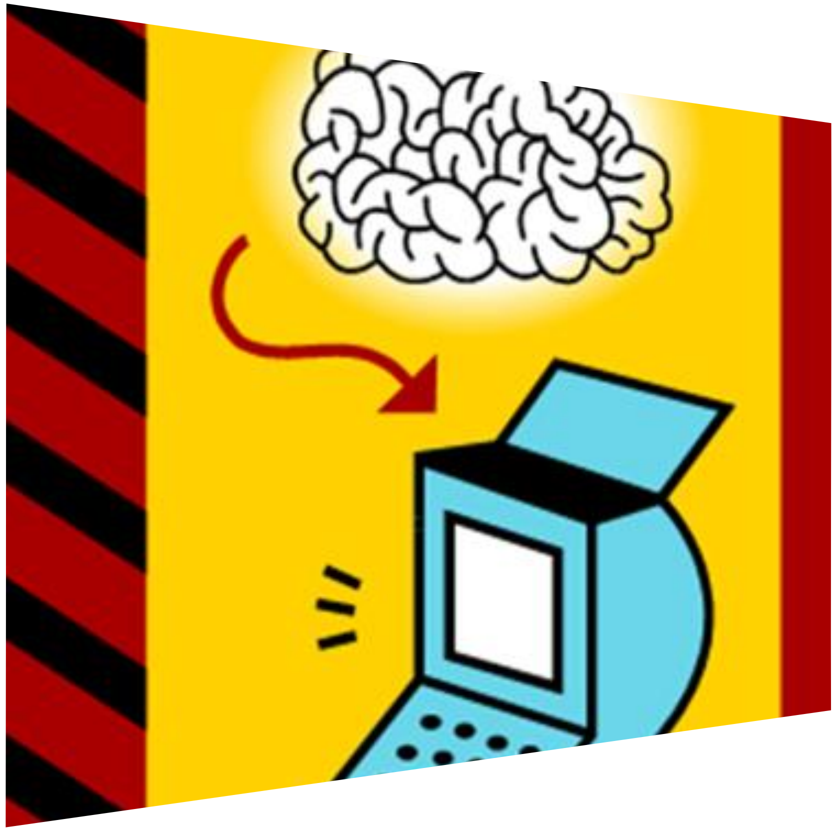
Real-world machine learning: how Kinect gesture recognition works

Alisson Sol
Principal Development Lead
Session 3-704

//build/

Agenda



How K4W gesture recognition works

How we did it

Experience that you can reuse

Recognition

- Many teams contributed to results shown here: Kinect for Windows, Microsoft Research, Xbox, etc.
- This presentation will focus on practical aspects
- Thanks to the Bing team for the flexibility

Non-goal: ML theory

Paper

- A Few Useful Things to Know about Machine Learning.
  Communications of the ACM, 55 (10), 78-87, 2012.

Books

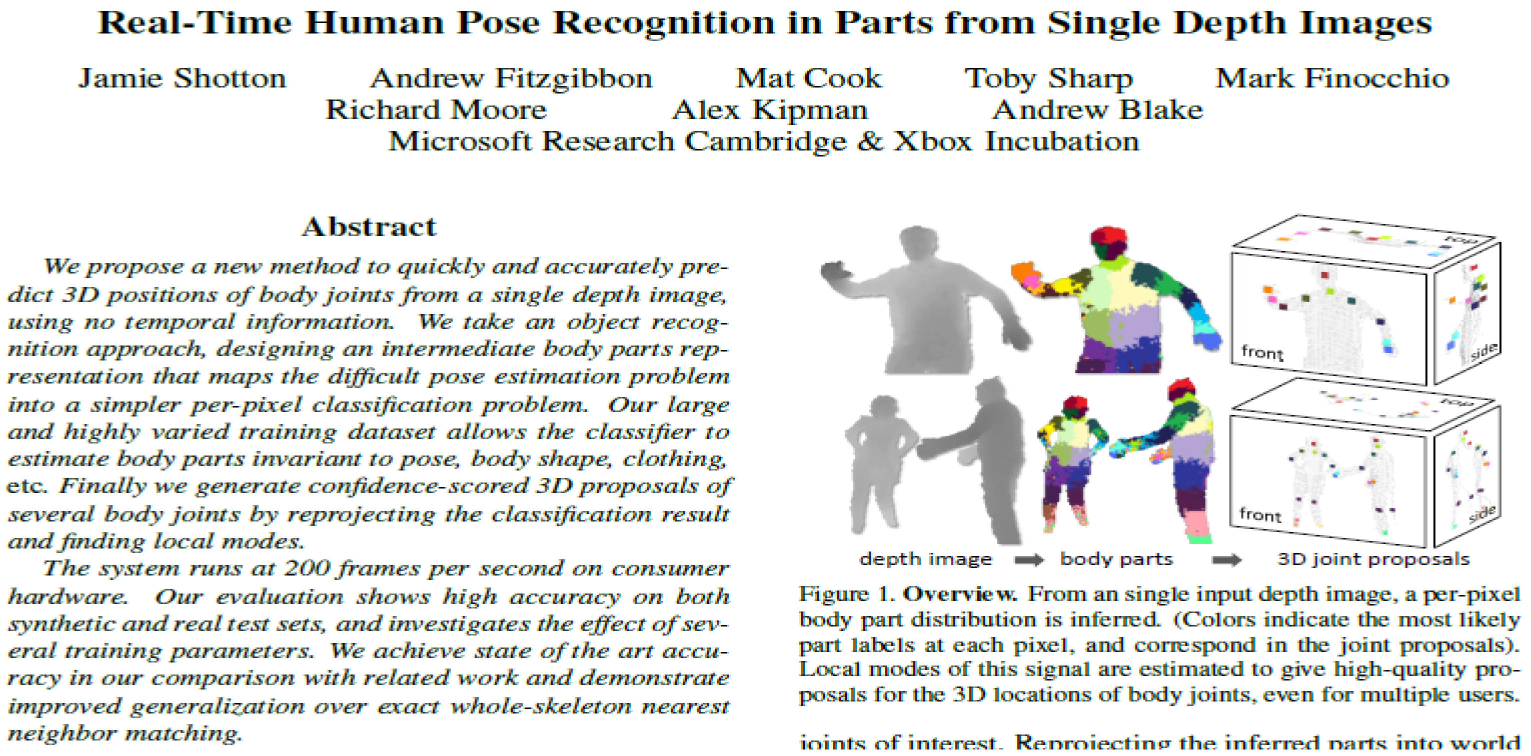
- Learning From Data, Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin
- Machine Learning in Action, Peter Harrington
- Pattern Recognition and Machine Learning, Christopher M. Bishop

Online courses

- https://www.coursera.org/course/machlearning
- https://www.coursera.org/course/ml

Machine learning in Kinect for Windows

- Skeletal tracking



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

Jamie Shotton Andrew Fitzgibbon Mat Cook Toby Sharp Mark Finocchio
Richard Moore Alex Kipman Andrew Blake
Microsoft Research Cambridge & Xbox Incubation

Abstract

We propose a new method to quickly and accurately predict 3D positions of body joints from a single depth image, using no temporal information. We take an object recognition approach, designing an intermediate body parts representation that maps the difficult pose estimation problem into a simpler per-pixel classification problem. Our large and highly varied training dataset allows the classifier to estimate body parts invariant to pose, body shape, clothing, etc. Finally we generate confidence-scored 3D proposals of several body joints by reprojecting the classification result and finding local modes.

The system runs at 200 frames per second on consumer hardware. Our evaluation shows high accuracy on both synthetic and real test sets, and investigates the effect of several training parameters. We achieve state of the art accuracy in our comparison with related work and demonstrate improved generalization over exact whole-skeleton nearest neighbor matching.

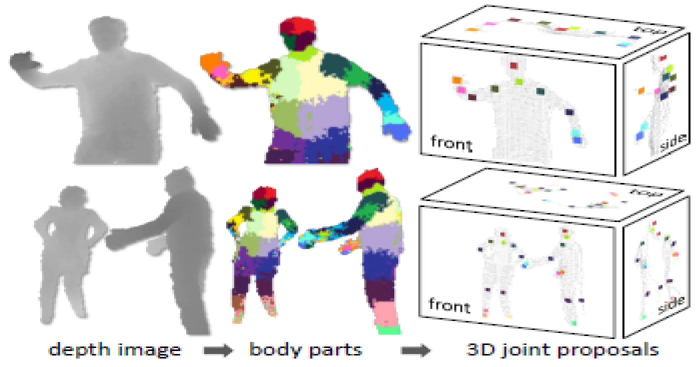


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

joints of interest. Reprojecting the inferred parts into world

- Hand events (Grip, GripRelease)

The goal: NUI demo

How we planned…

1) Skeletal tracking finds your hand
2) Window of NxN pixels around each hand is “extracted”
3) Machine learning executes and generates events

What needs to be done (v1)

- Interface with other components
- Data collection and tagging
- Machine learning training code

Our status was: 2nd order of ignorance

0th: Lack of ignorance (have the answer)

1st: Lack of knowledge (have the question)

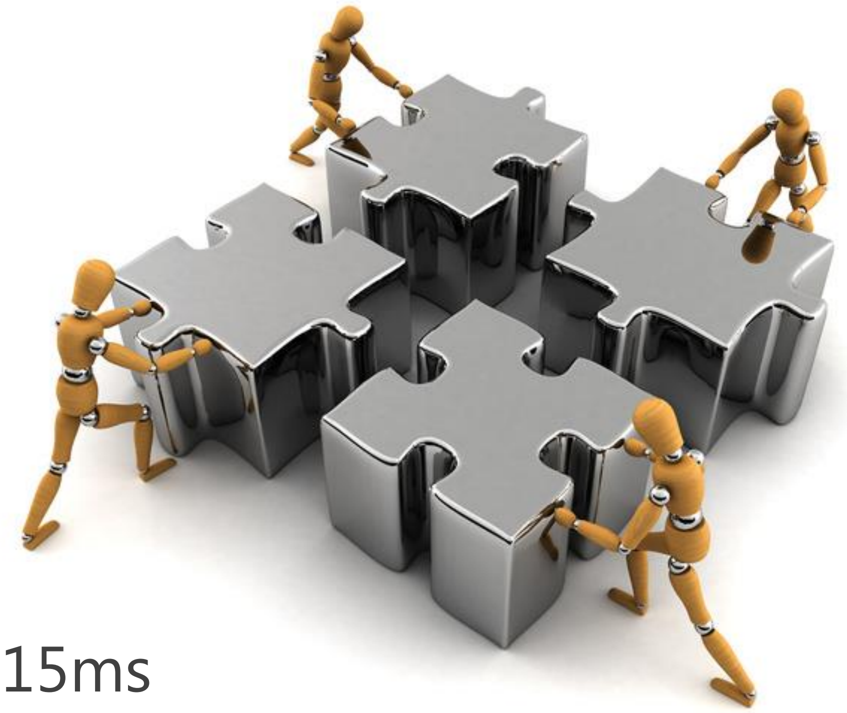
2nd: Lack of awareness (don’t know what you don’t know)

3rd: Lack of process (no way out of lack of knowledge)

4th: Meta ignorance (not aware of these 5 levels)

Live preview demo

Divide + conquer = puzzle



Hand position is not stable

Runtime performance goals

- 30 fps ~ 33ms per frame
- For minimum hardware, skeletal tracking taking ~15ms
- Goal for 4 hands =~2 ms

Past versus future

- 3 teams failed in the past
- Fast pace of releases/milestones/sprints

Kinect for Windows major releases

1.0: Feb/01/2012

1.5: May/21/2012

1.6: Oct/04/2012

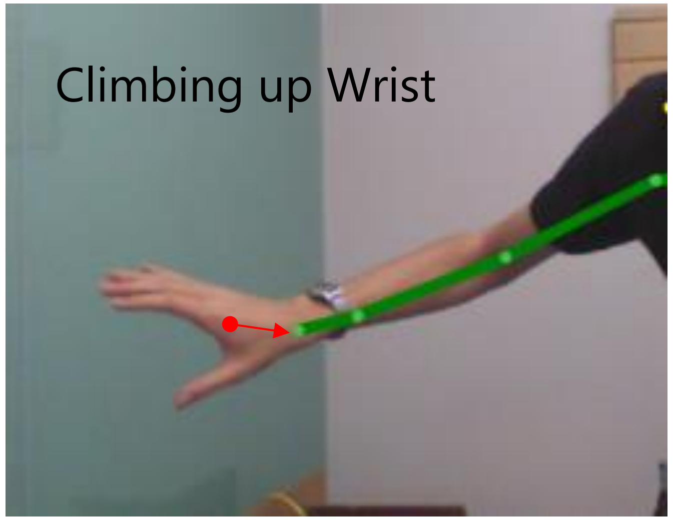
1.7: Mar/18/2013

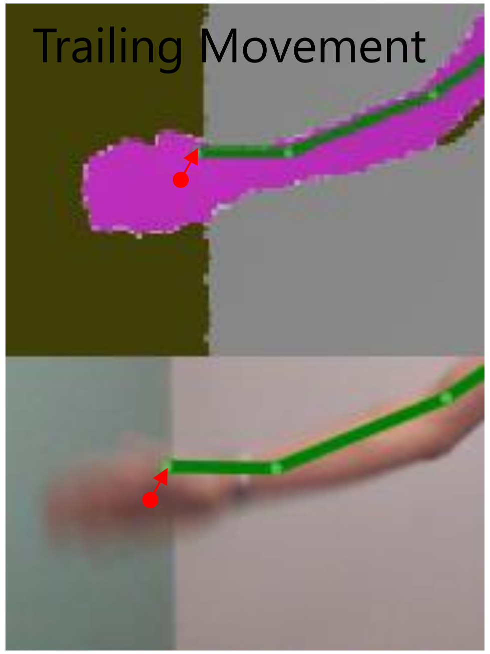
- Recognizer ended: Feb/11/2013
- TAP (Technical Adoption Program) release: Dec/19/2012

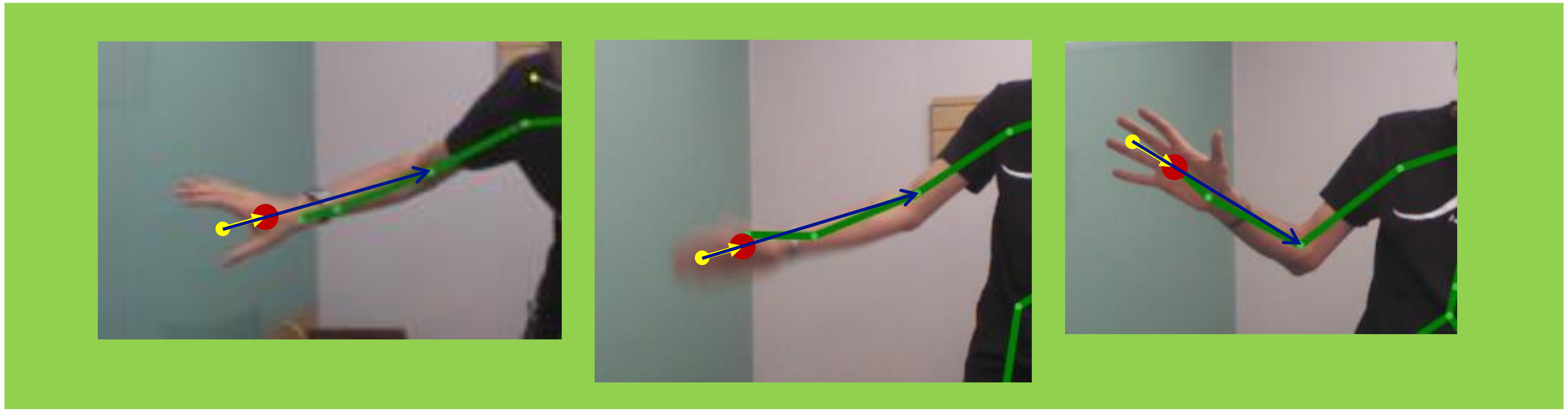
What needs to be done (v2)

- ~~Interface with other components~~
- Hand stabilization
- Data collection and tagging
- Machine learning training code
- Performance and accuracy tools

Hand stabilization



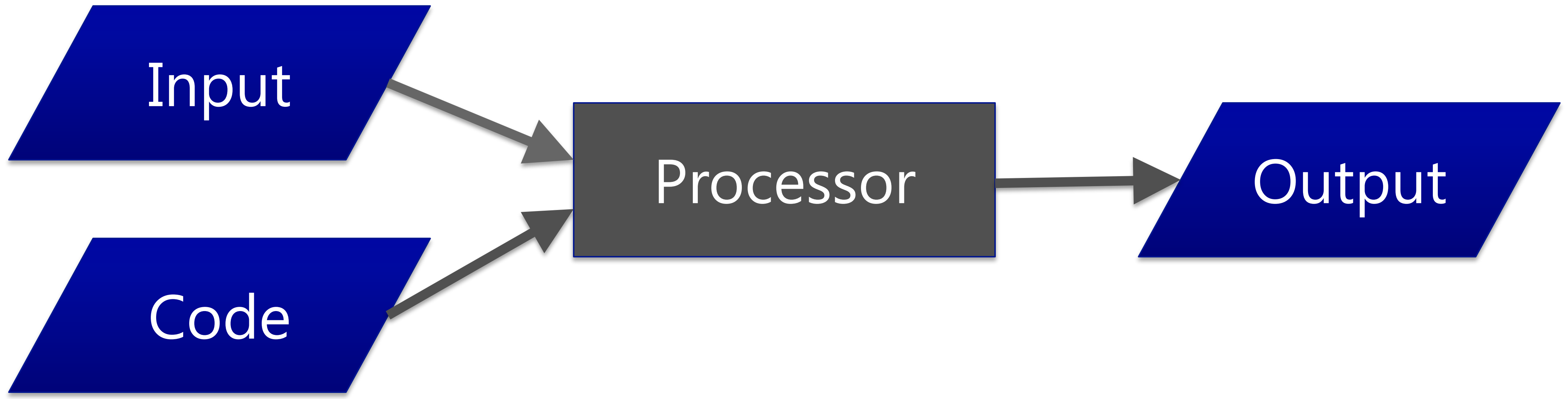




What needs to be done (v2)

- ~~Interface with other components~~
- ~~Hand stabilization~~
- Data collection and tagging
- Machine learning training code
- Performance and accuracy tools

Usual development ML development



ML applications

- Classification
- Grouping
- Ranking

…

ML definitions

- Feature
- Training
- Verification x Validation

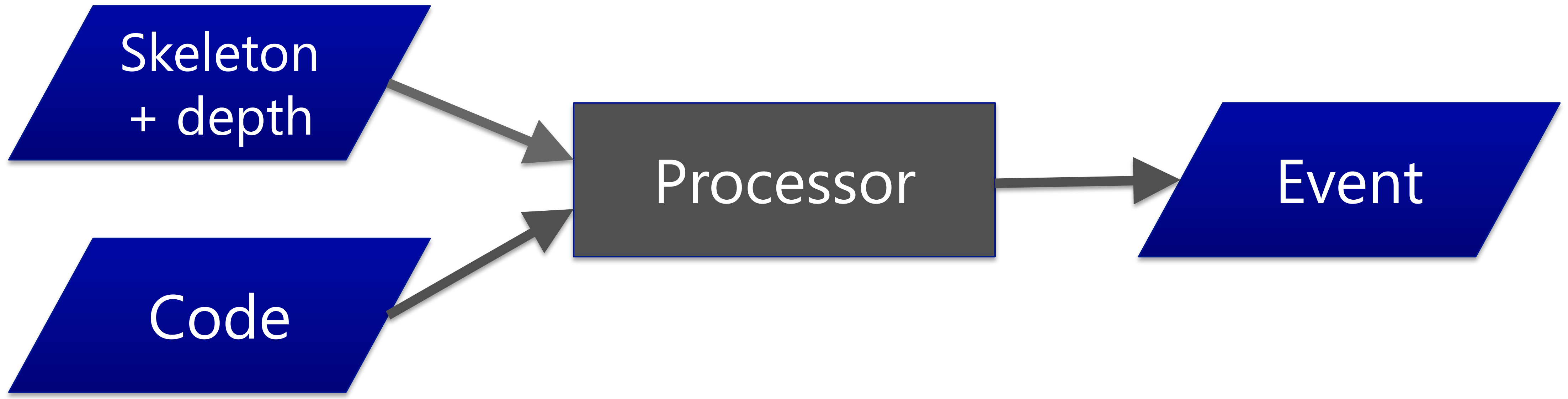
An example from real estate

ZIP	Beds	Baths	Sq ft	Built	Lot	Update	Value
98052	4	3	2,200	1968	10,000	-	400,000
98004	4	3	2,100	1968	10,000	-	500,000
98004	5	3	2,400	1970	9,000	2005	600,000
98008	4	2.5	2,200	1980	5,000	-	500,000
…	…	…	…	…	…	…	…
98052	4	2	2,200	1990	7,000	-	???

Dev verification x QA validation

Ground truth demo

Recognizer development



Problem size

Features

- Skeleton data, deltas (among joints), time deltas (across frames), depth differences (in frame, over time), etc…

Depth data per hour of recording

- 60 min/hour x 60 sec/min x 30 fps =~ 108,000 frames
- Each frame = 640 x 480 = 300 K pixels
- Section around a hand = 128 x 128 = 16 K pixels
- Frame depth differences =~ 16 K x 16 K x 0.5 =~ 128 M features

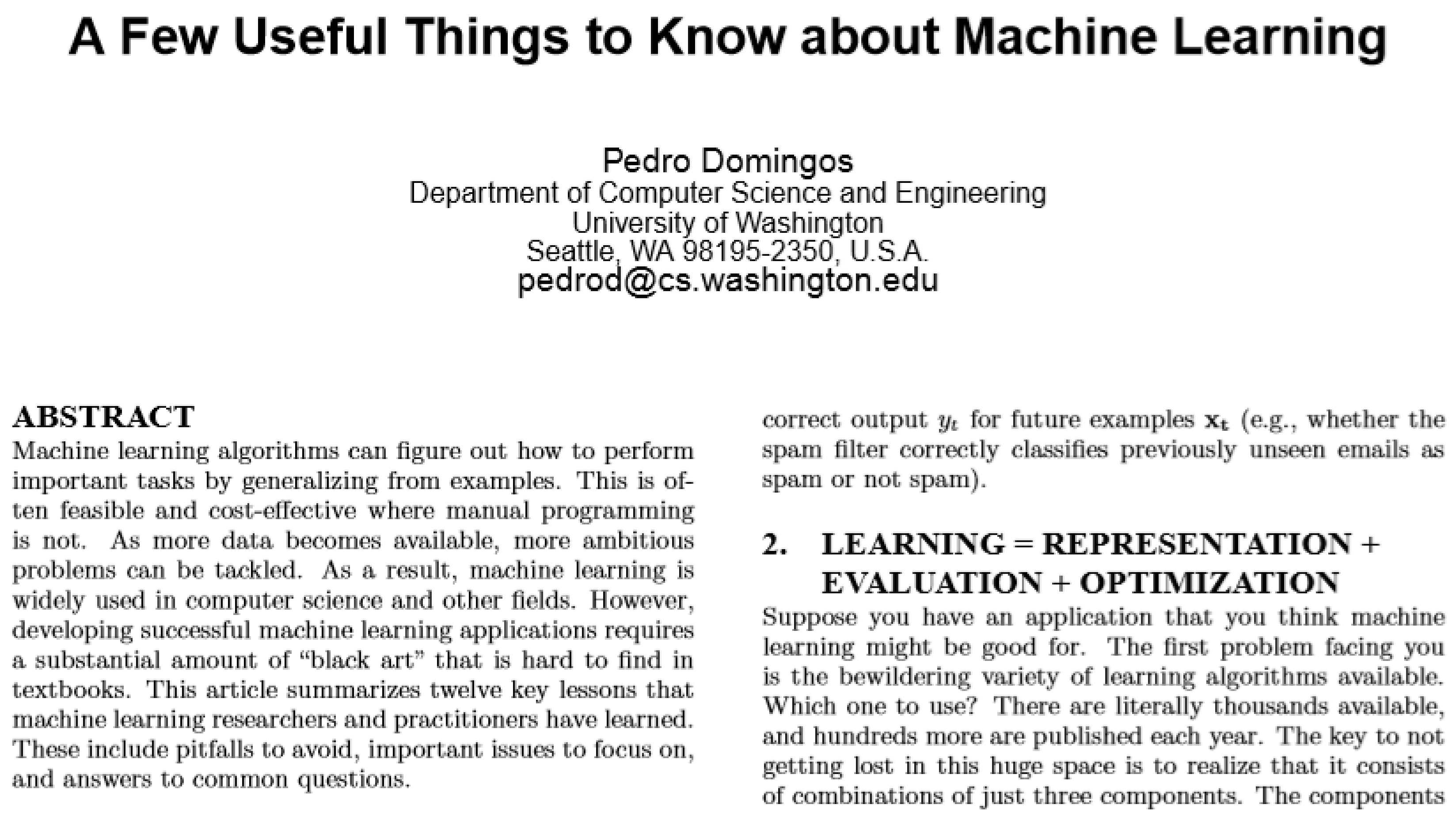
What needs to be done (v2)

- ~~Interface with other components~~
- ~~Hand stabilization~~
- ~~Data collection and tagging~~
- Machine learning training code
- Performance and accuracy tools

Training methods

kNN
SVM
Decision Forest
Bayes
Neural Net

Selecting a training method



A Few Useful Things to Know about Machine Learning

Pedro Domingos
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
pedrod@cs.washington.edu

ABSTRACT

Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not. As more data becomes available, more ambitious problems can be tackled. As a result, machine learning is widely used in computer science and other fields. However, developing successful machine learning applications requires a substantial amount of "black art" that is hard to find in textbooks. This article summarizes twelve key lessons that machine learning researchers and practitioners have learned. These include pitfalls to avoid, important issues to focus on, and answers to common questions.

correct output y_t for future examples $\mathbf{x_t}$ (e.g., whether the spam filter correctly classifies previously unseen emails as spam or not spam).

2. LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

Suppose you have an application that you think machine learning might be good for. The first problem facing you is the bewildering variety of learning algorithms available. Which one to use? There are literally thousands available, and hundreds more are published each year. The key to not getting lost in this huge space is to realize that it consists of combinations of just three components. The components

Right method == correctly implemented

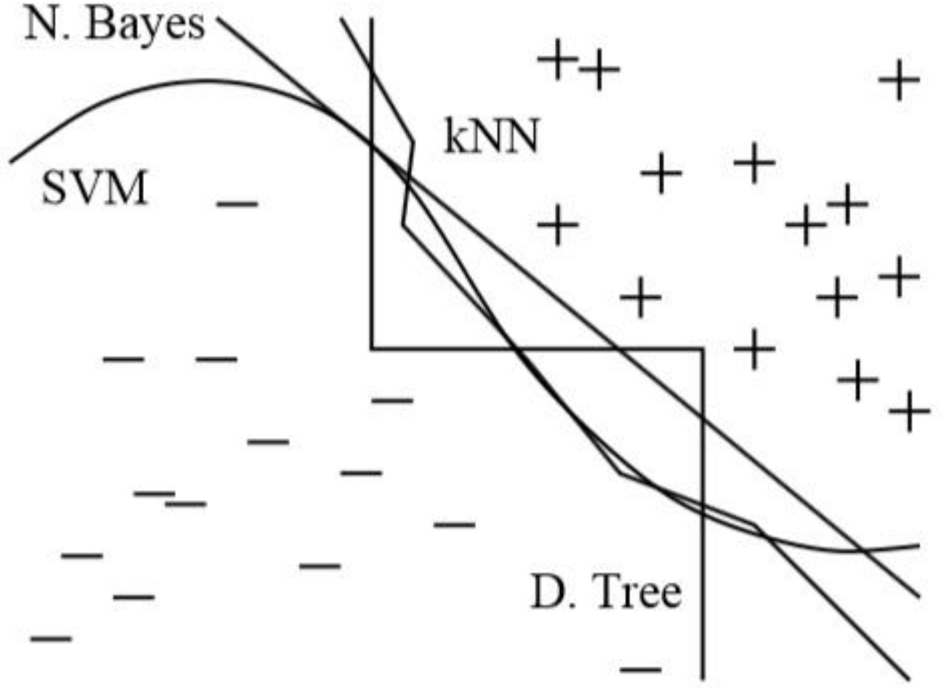
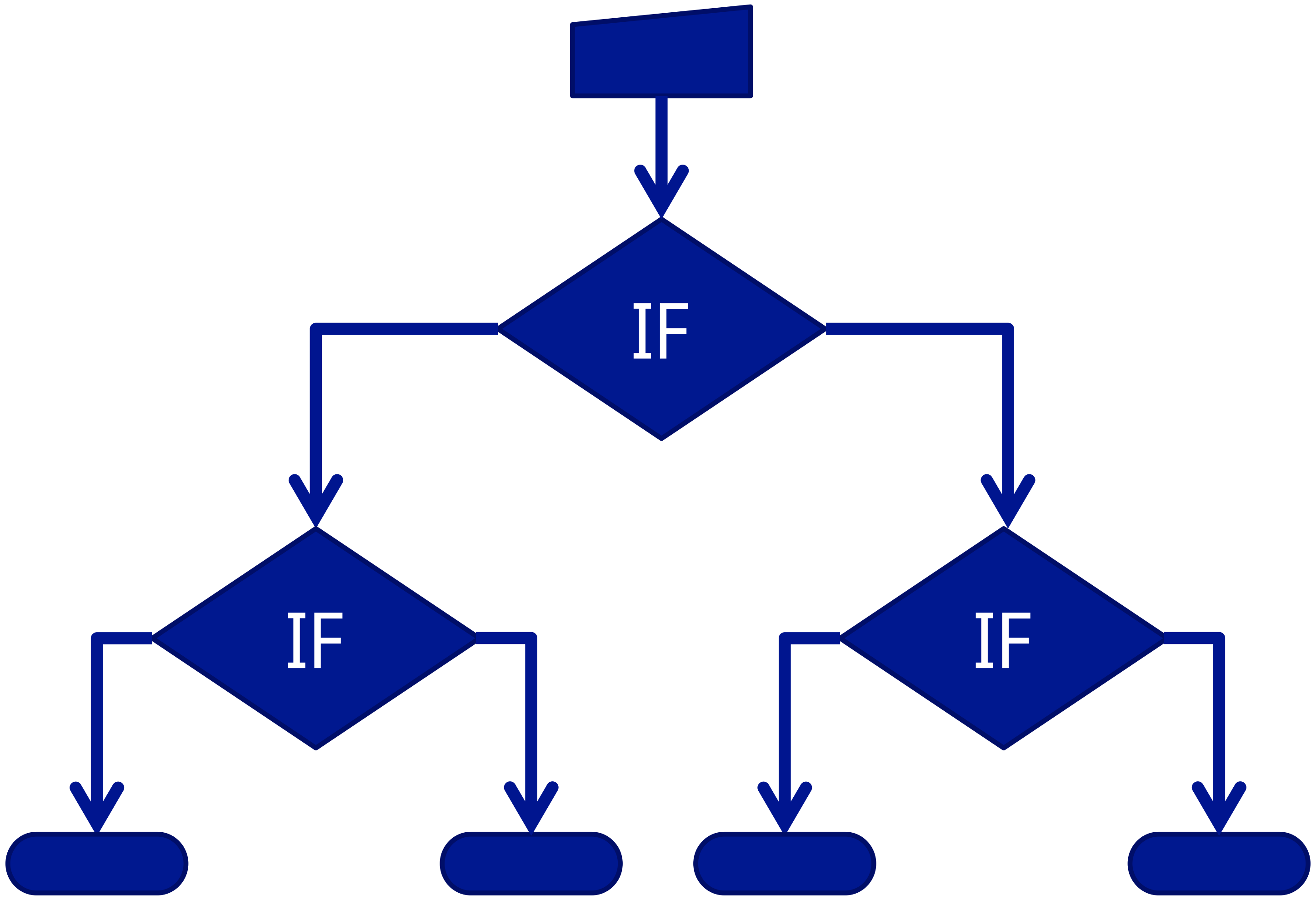


Figure 3: Very different frontiers can yield similar class predictions. (+ and – are training examples of two classes.)

Warning

The following “theory-related” section will be presented intentionally in a fast pace…

A decision tree



ID3: Iterative Dichotomizer 3

For unused features

- Choose feature of maximum information gain
- Make node splitting data based on test against that feature

End conditions

- Tree depth, lack of data, lack of features

Marine animal data

Survive underwater?	Has flippers?	Fish?
Y	Y	Y
Y	Y	Y
Y	N	N
N	Y	N
N	Y	N

Source: book Machine Learning in Action

Which feature to test first?

Information as function of probability for outcomes

$$I(x_i) = \log_2 p(x_i)$$

Shannon's entropy

$$H = -\sum_{i=1}^{n} p(x_i) I(x_i)$$

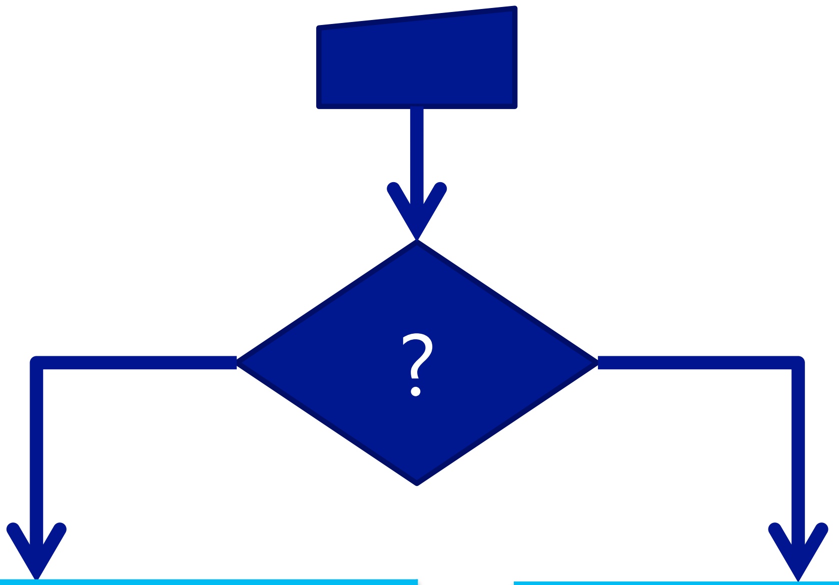
Calculate entropy

Under?	Flipper?	Fish?	Prob.	Log2(Prob.)	Entropy
Y	Y	Y			
Y	Y	Y			
Y	N	N			
N	Y	N			
N	Y	N			

Applying formulas

Under?	Flipper?	Fish?	Prob.	Log2(Prob.)	Entropy
Y	Y	Y	0.4	-1.32	0.53
Y	Y	Y			
Y	N	N	0.6	-0.74	0.44
N	Y	N			
N	Y	N			
					0.97

"Flippers?"

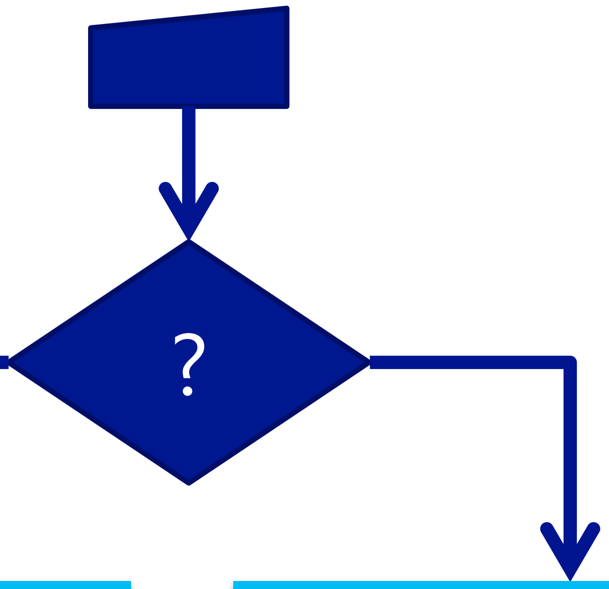


Under?	Flipper?	Fish?	P.	Log2(P.)	Entropy
Y	Y	Y	0.5	-1.00	0.50
Y	Y	Y			
N	Y	N	0.5	-1.00	0.50
N	Y	N			
					1.00

Under?	Flipper?	Fish?	P.	Log2(P.)	Entropy
Y	N	N	1	0.00	0.00

H(Flippers?) = H(Y)*4/5 + H(N)*1/5 = 1.0*4/5 + 0*1/5 = 0.8

“Underwater?”

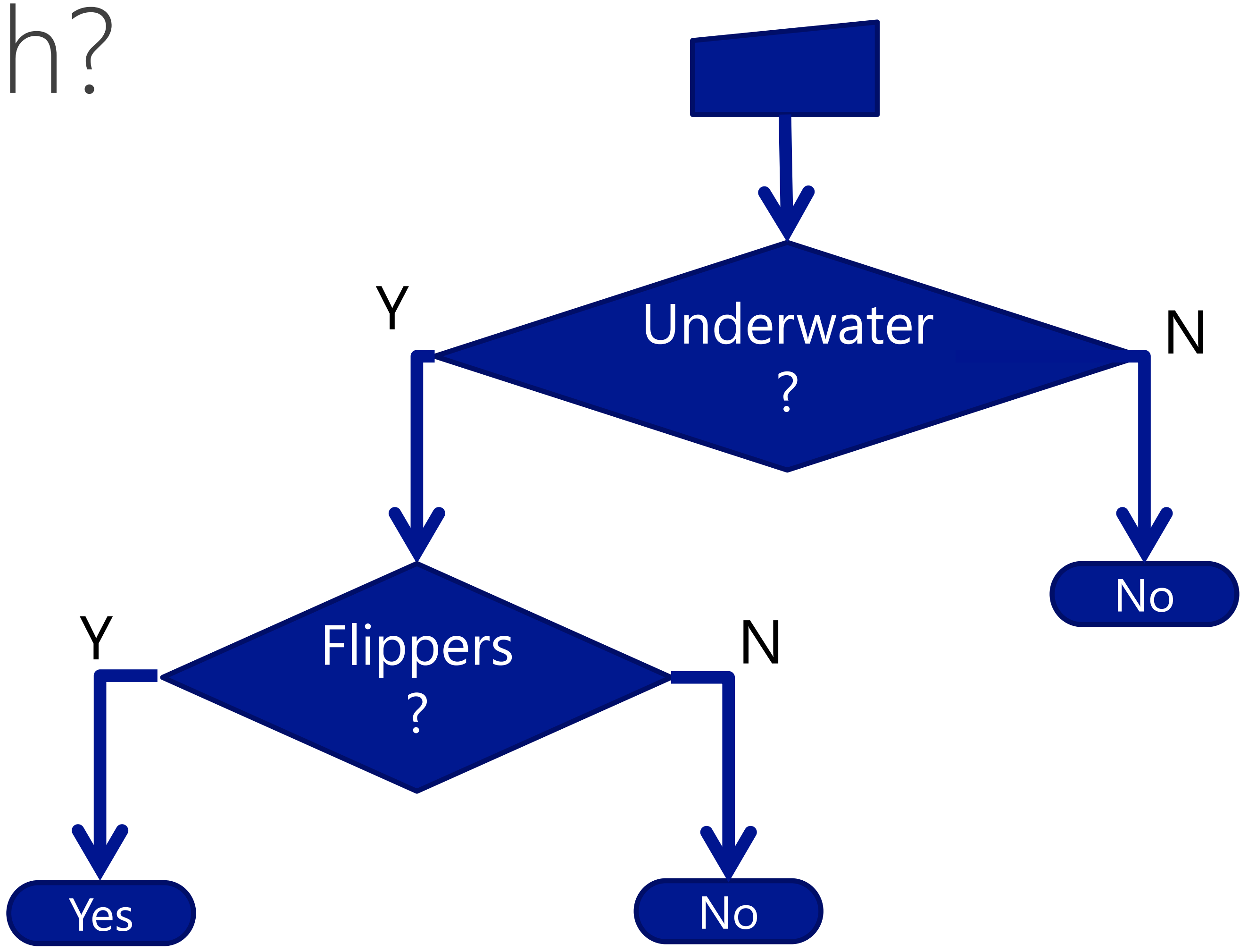


Under?	Flipper?	Fish?	P.	Log2(P.)	H
Y	Y	Y	0.67	-0.58	0.39
Y	Y	Y			
Y	N	N	0.33	-1.58	0.53
					0.92

Under?	Flipper?	Fish?	P.	Log2(P.)	Entropy
N	Y	N	1	0.00	0.00
N	Y	N			

H(Underwater?) = H(Y)*3/5 + H(N)*2/5 = 0.92*3/5 + 0*2/5 = 0.552

Is Fish?



End of fast road…

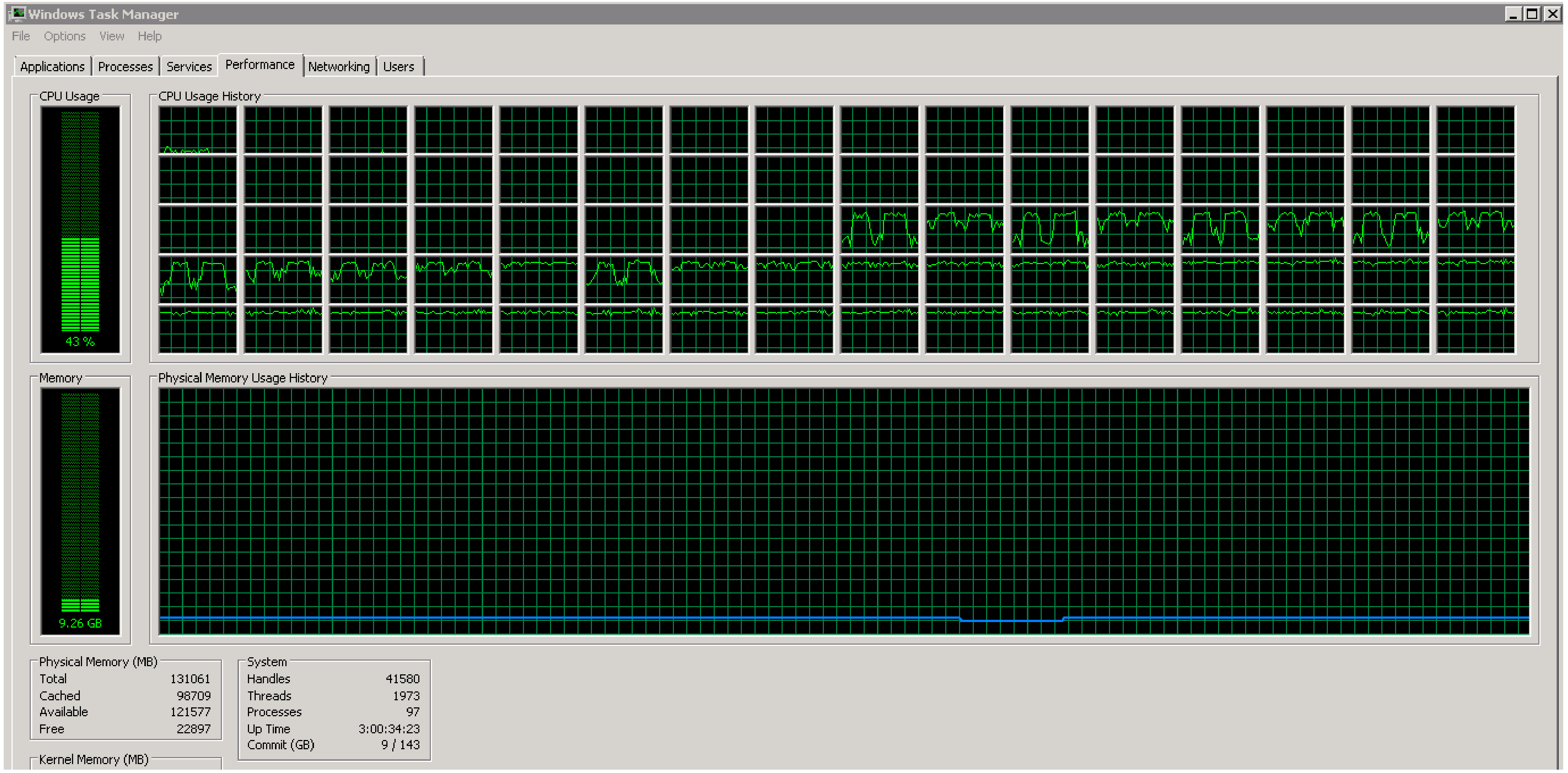
Ommitted several details

- Labels may not be boolean
- “Ground truth” can have wrong labels
- Trees need testing, pruning, etc.

…

- Your resulting model may not predict anything!
  - Because of the data…
  - Because of the model…

Training is costly...


Windows Task Manager
File Options View Help
Applications Processes Services Performance Networking Users
CPU Usage
43 %
CPU Usage History
Memory
9.26 GB
Physical Memory Usage History

Physical Memory (MB)	
Total	131061
Cached	98709
Available	121577
Free	22897

Kernel Memory (MB)

System	
Handles	41580
Threads	1973
Processes	97
Up Time	3:00:34:23
Commit (GB)	9 / 143

Hand state demo

What needs to be done (v2)

- ~~Interface with other components~~
- ~~Hand stabilization~~
- ~~Data collection and tagging~~
- ~~Machine learning training code~~
- Performance and accuracy tools

Why a ML classifier may fail (in production)?

- Wrong features
- “Hidden/unknown” features
- Wrong implementation
  - Debugging ML-based systems is hard
- Wrong testing…

Which surgical team do you prefer?

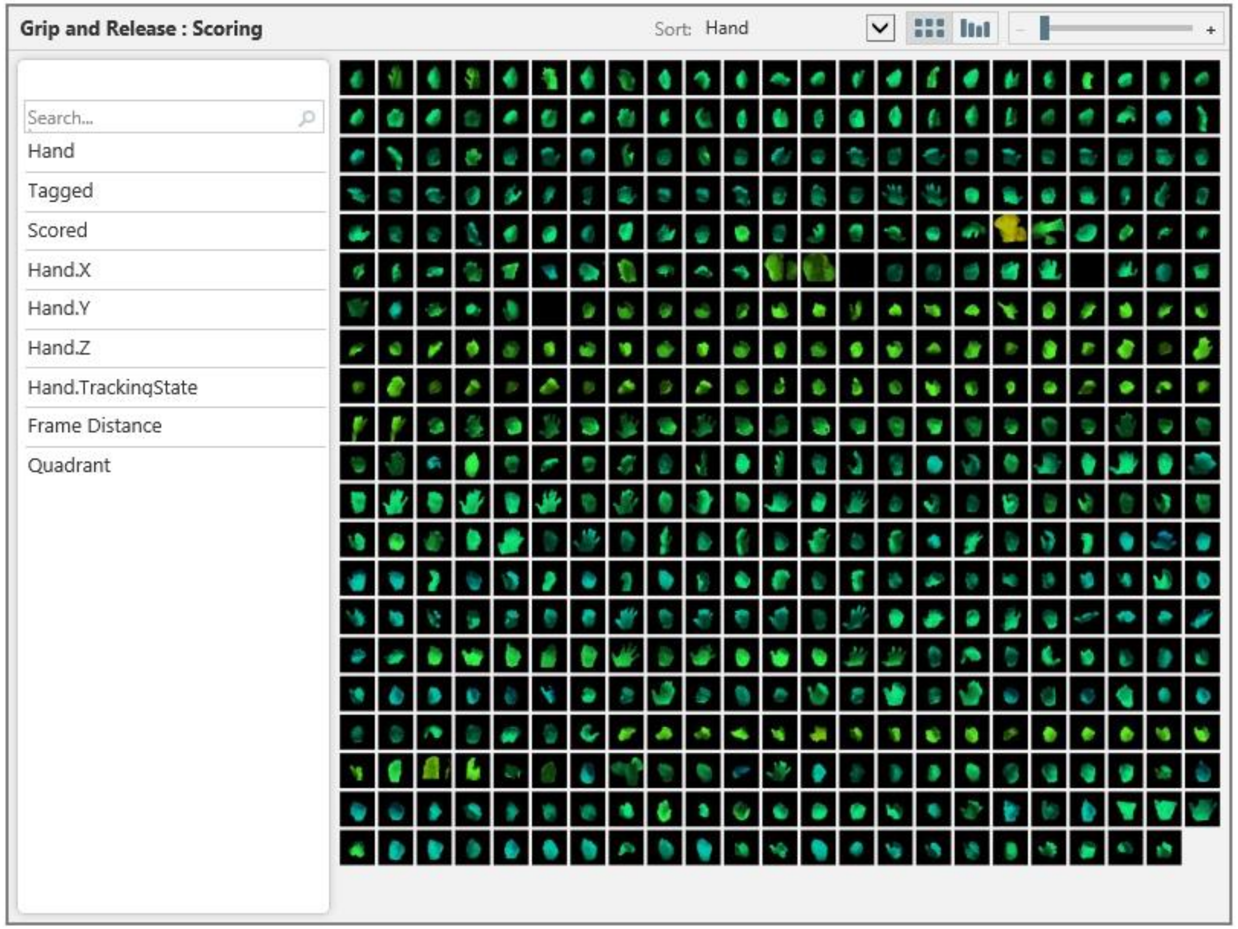
Team 1	Team 2
Nurse	Nurse
Surgeon	Surgeon
Assistant surgeon	Assistant surgeon
Surgeon	Cardiologist
Surgeon	Perfusionist
Surgeon	Anesthesiologist
Nurse	Scrub tech

Test data and scenarios…

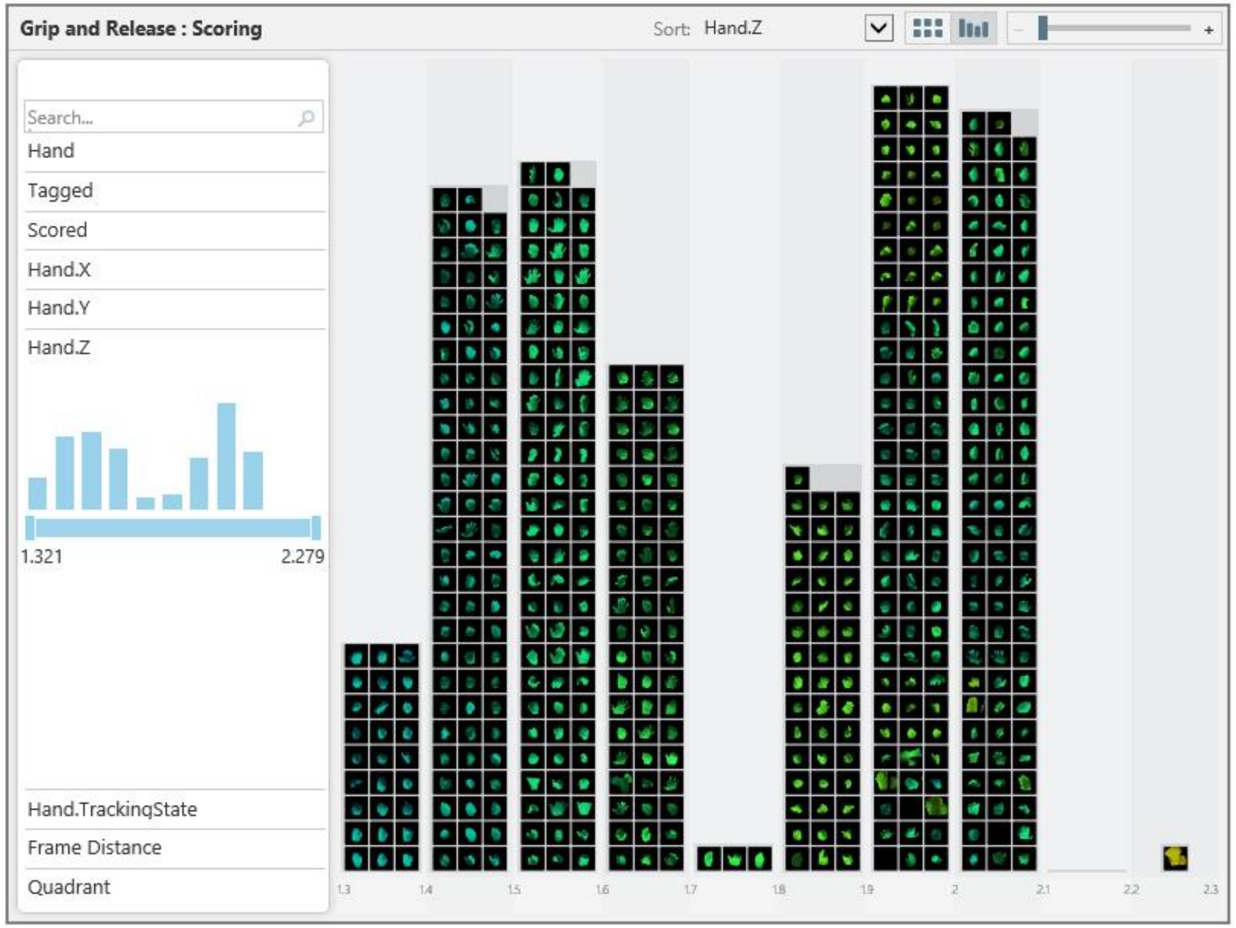
May not represent “population”

May not be large enough

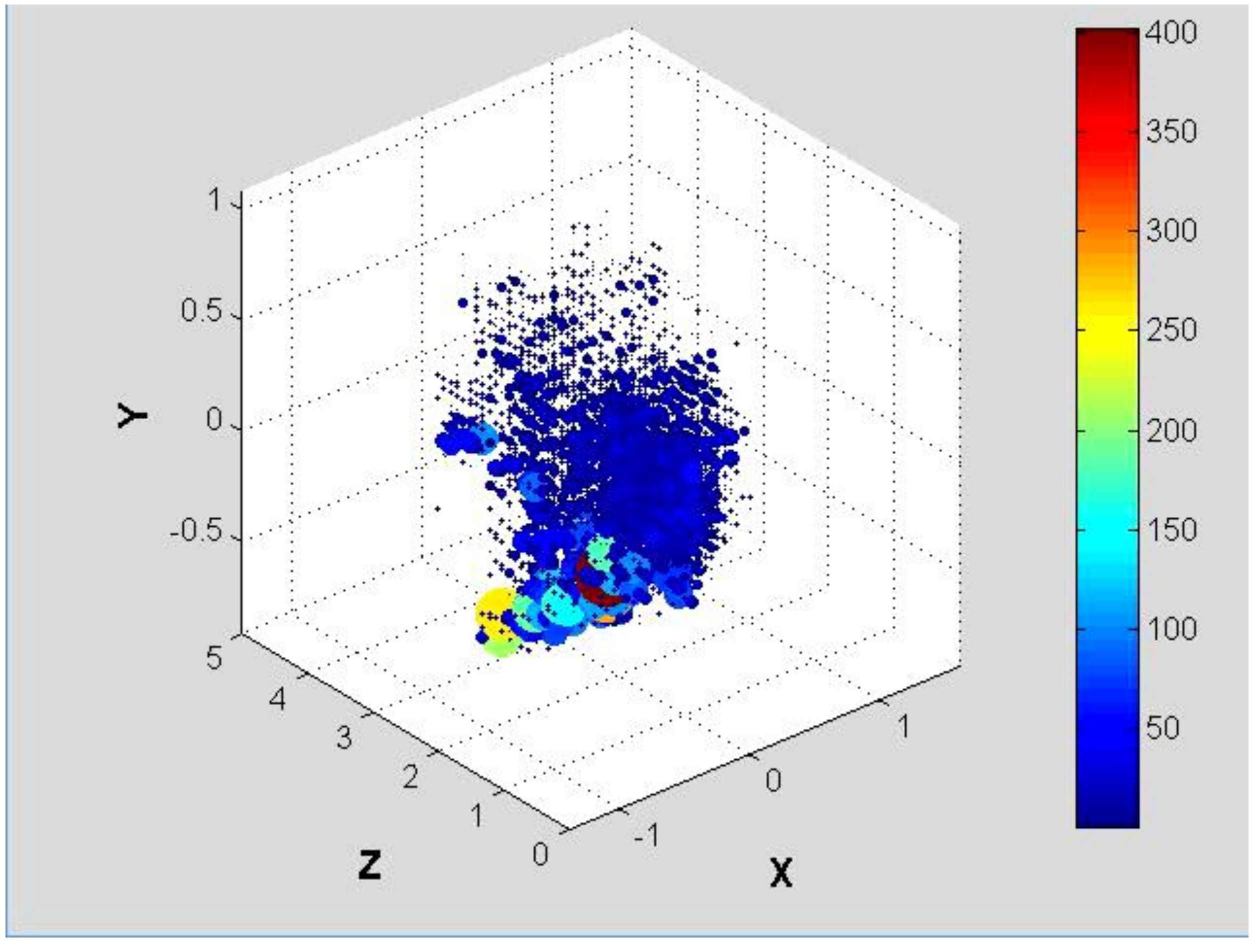
Data analysis for an experiment



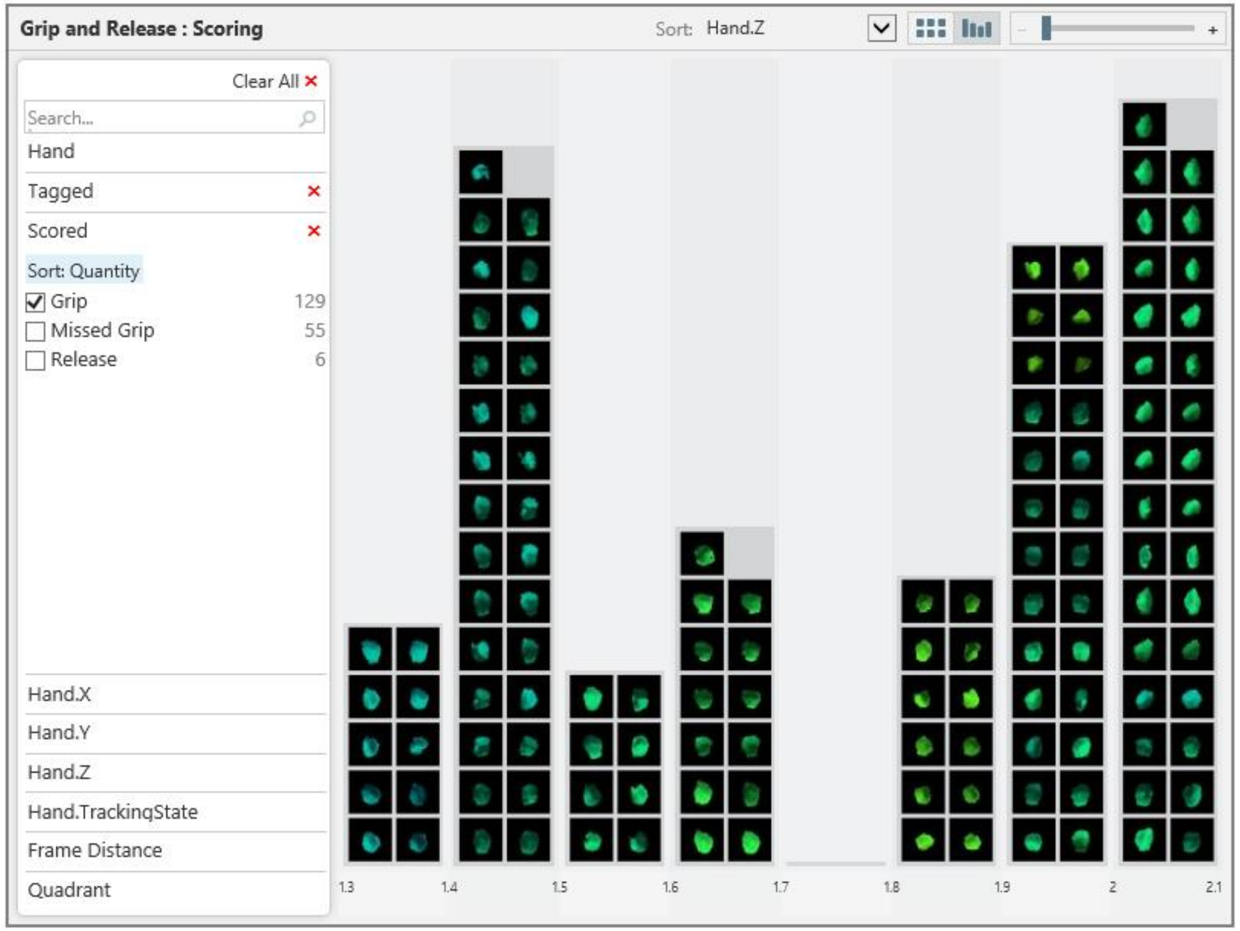
Sort by distance



Data in 3-D space



Grip true positives


Grip and Release : Scoring

Sort: Hand.Z

Clear All ✕

Search...

Hand

Tagged ✕

Scored ✕

Sort: Quantity

☑ Grip	129
☐ Missed Grip	55
☐ Release	6

Hand.X

Hand.Y

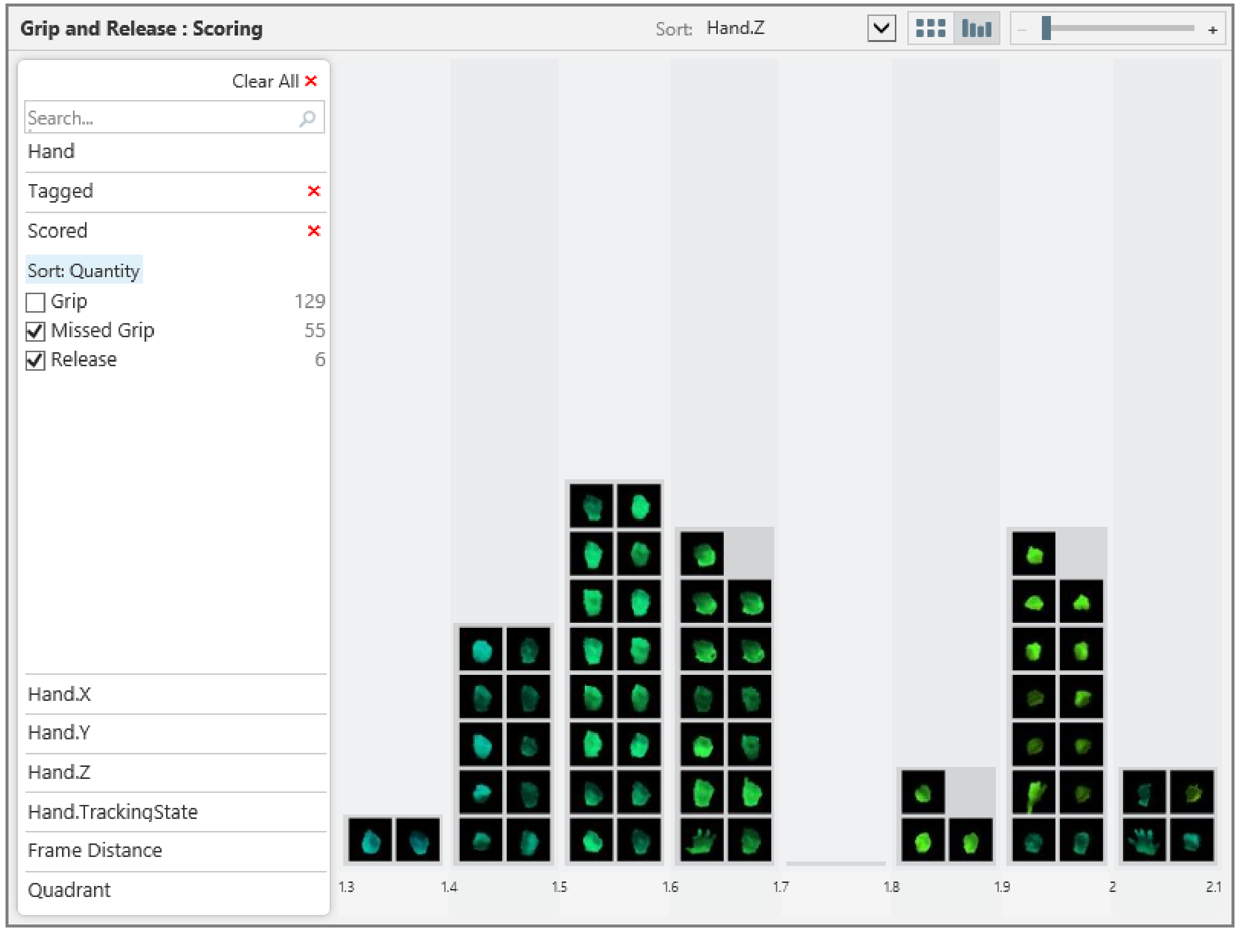
Hand.Z

Hand.TrackingState

Frame Distance

Quadrant

Grip false negatives



Grip and Release : Scoring

Sort: Hand.Z

Clear All ×

Search...

Hand

Tagged ×

Scored ×

Sort: Quantity

☐ Grip	129
☑ Missed Grip	55
☑ Release	6

Hand.X

Hand.Y

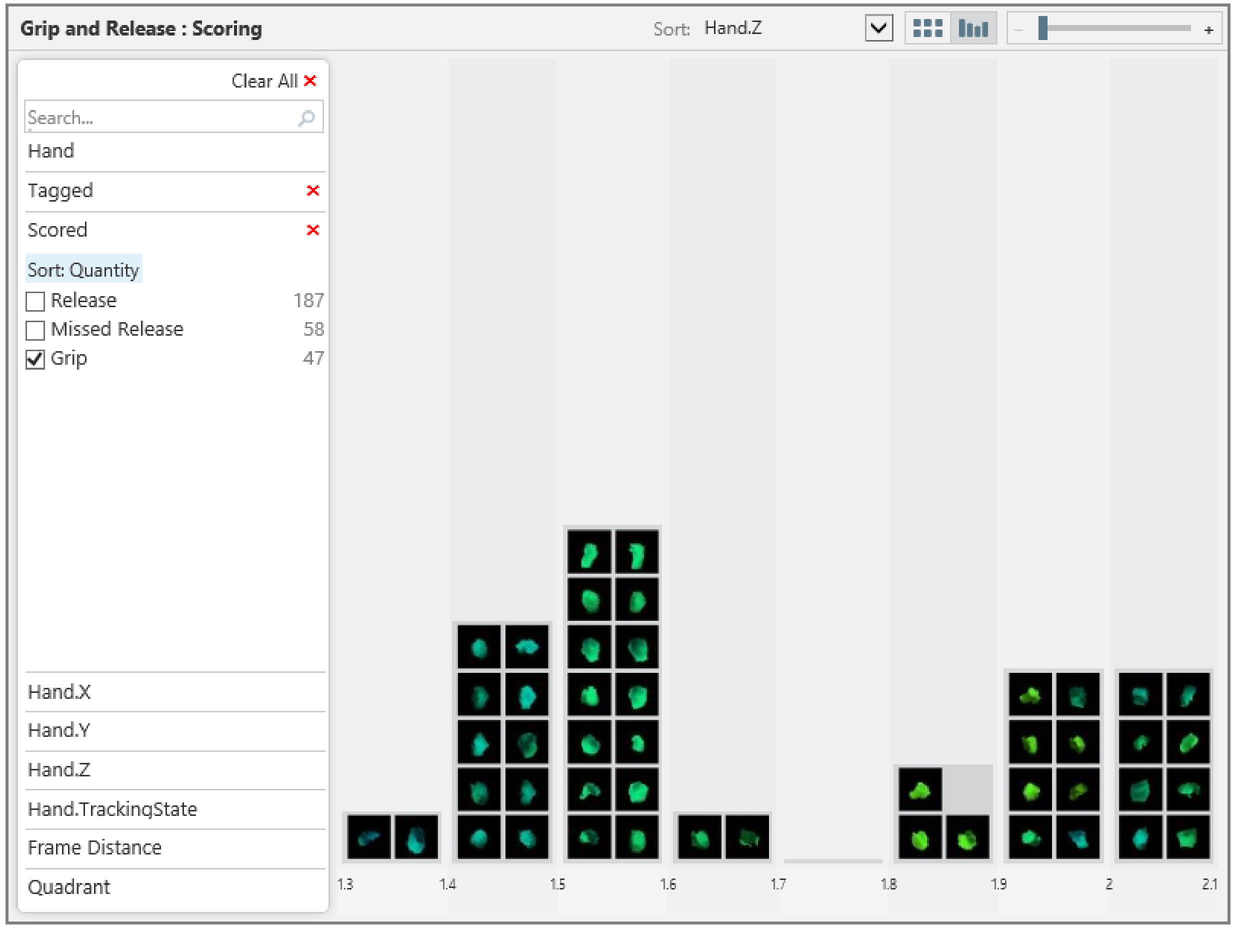
Hand.Z

Hand.TrackingState

Frame Distance

Quadrant

Grip false positives



Grip and Release : Scoring

Sort: Hand.Z

Clear All ✕

Search...

Hand

Tagged ✕

Scored ✕

Sort: Quantity

☐ Release	187
☐ Missed Release	58
☑ Grip	47

Hand.X

Hand.Y

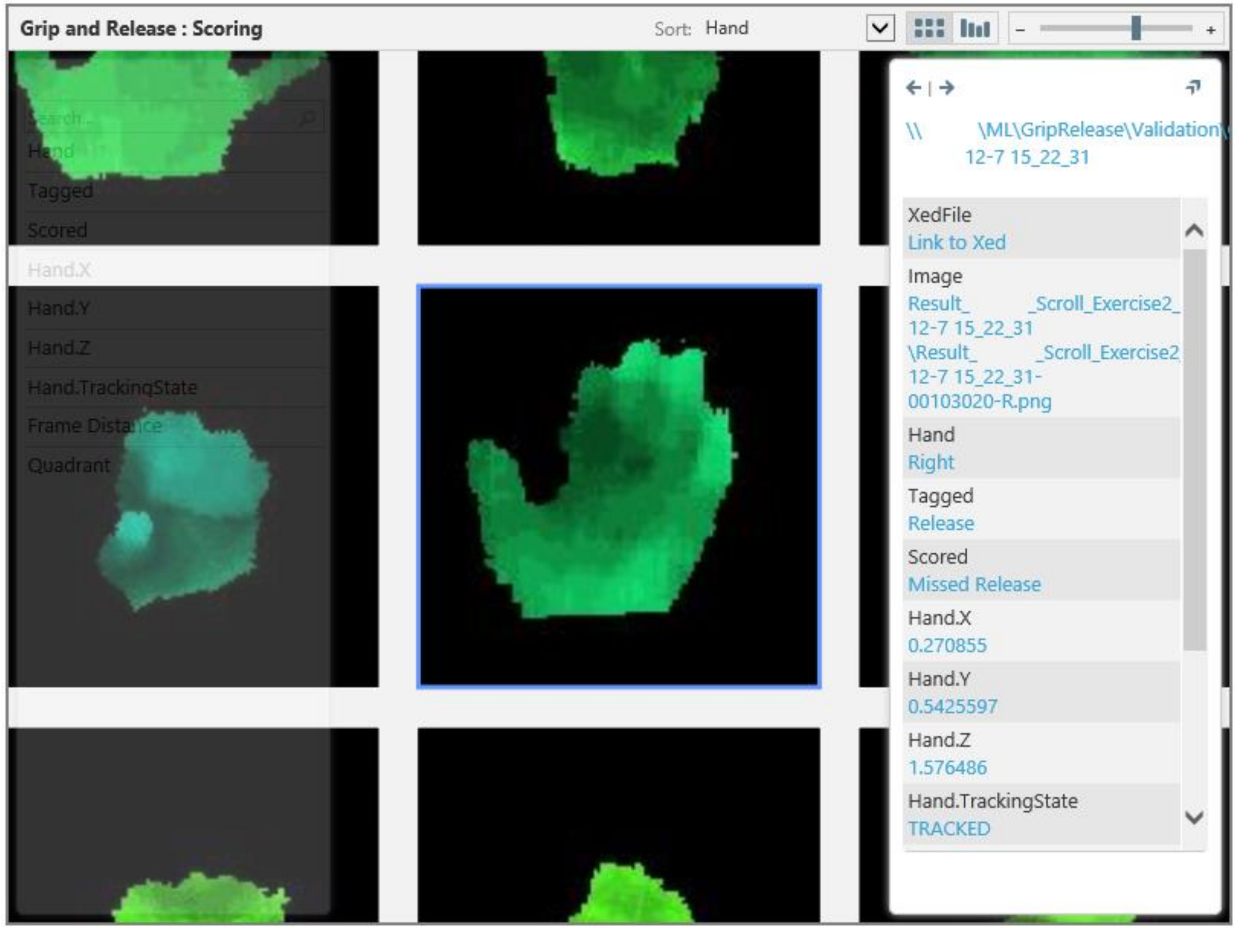
Hand.Z

Hand.TrackingState

Frame Distance

Quadrant

Drill-down to details



Grip and Release : Scoring

Sort: Hand

\\ \ML\GripRelease\Validation\ 12-7 15_22_31

XedFile	Link to Xed
Image	Result_ _Scroll_Exercise2_ 12-7 15_22_31 \Result_ _Scroll_Exercise2 12-7 15_22_31-00103020-R.png
Hand	Right
Tagged	Release
Scored	Missed Release
Hand.X	0.270855
Hand.Y	0.5425597
Hand.Z	1.576486
Hand.TrackingState	TRACKED

Data and metrics

- True positives, false negatives, false positives
- Overwhelming true negatives and influence on metrics
    - Precision = TP / (TP + FP)
    - Recall = TP / (TP + FN)
    - Accuracy = (TP + TN) / (TP + TN + FP + FN)
    - F-Score = 2 * (Precision * Recall) / (Precision + Recall)

Testing ML requires domain knowledge

Make sure test data represents population

- “Clustering” population

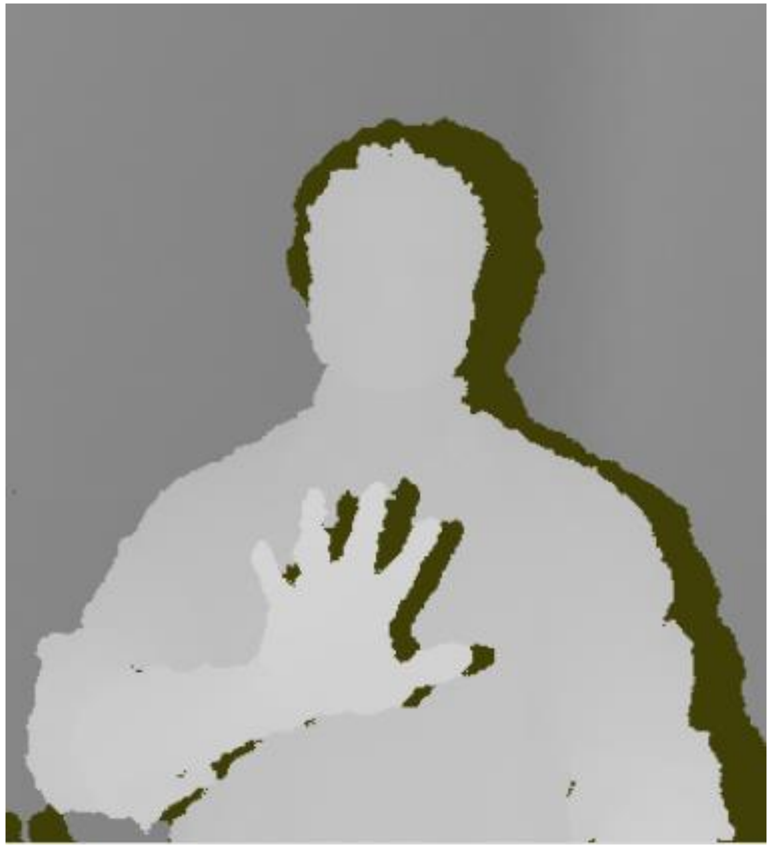
Characterizing classes of bugs properly

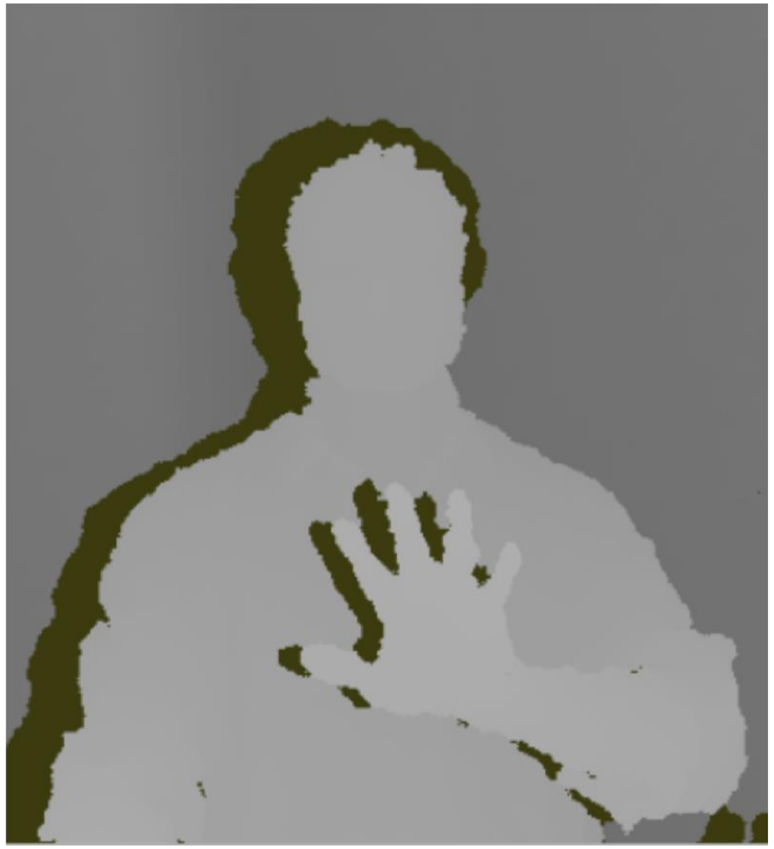
- Otherwise you have bugs in quantity, but not in quality

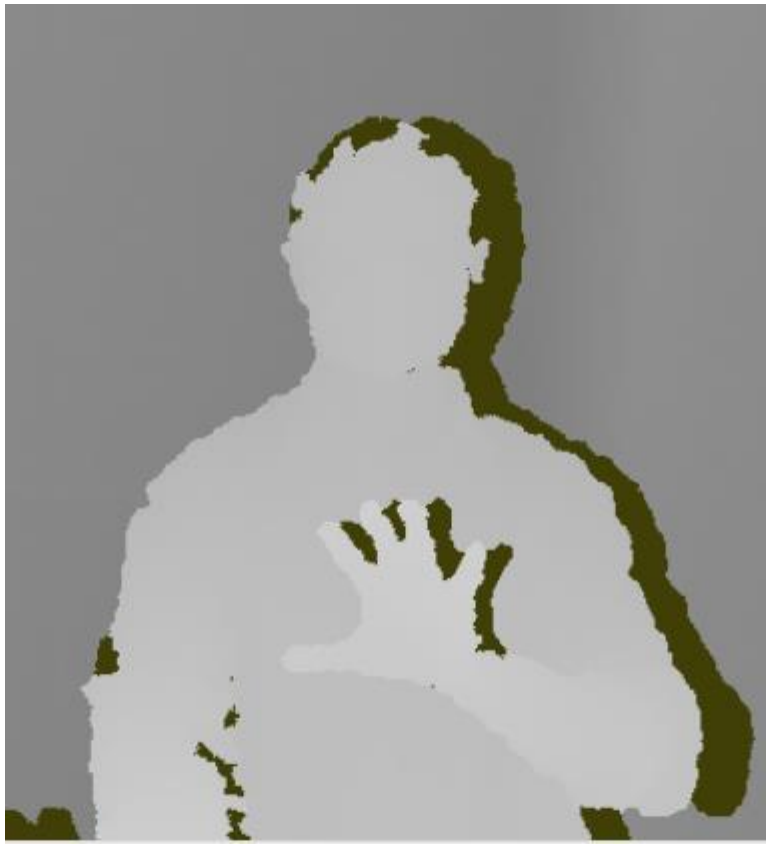
Bugs on TAP release (Dec/19/2012)

- Grip recognizer inconsistent between left and right hands
- AV when wake up machine from long time sleep
- Cursor moves downward on grip
- Grip recognition performs worse during vertical scrolling than horizontal scrolling
- Palm facing Grip Problematic
- Grip detection does not work with sweater/sleeves
- Phiz shows apparent random walk behavior for specific user position
- Release event triggered before Release.Begin Tag
- There seems to be a high latency though between while gripping and when the app responds to it, via the cursor visualizer.

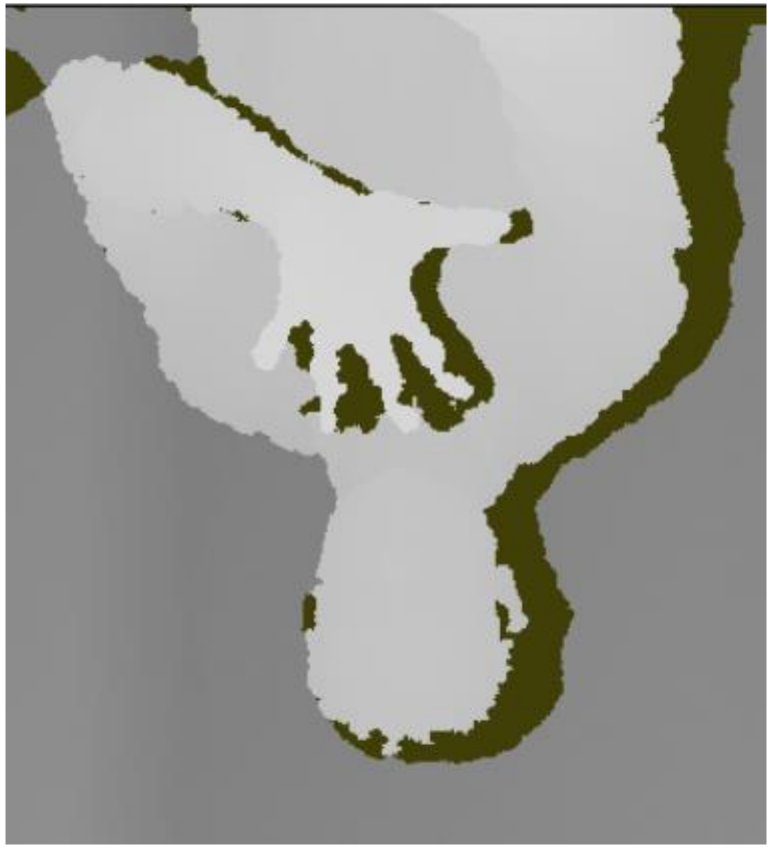
Depth data is not symmetrical

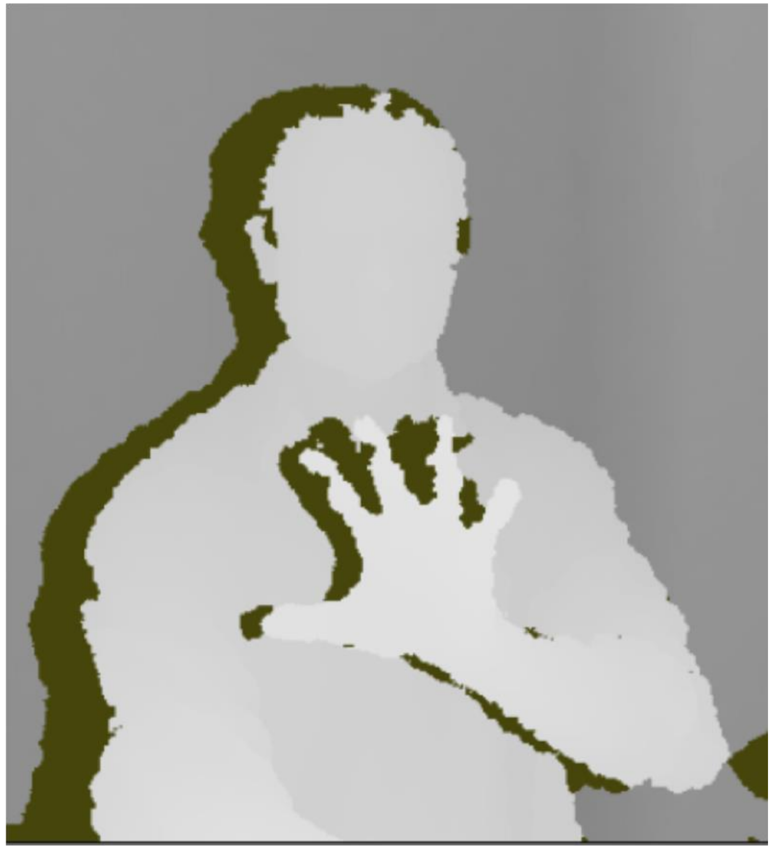

Left

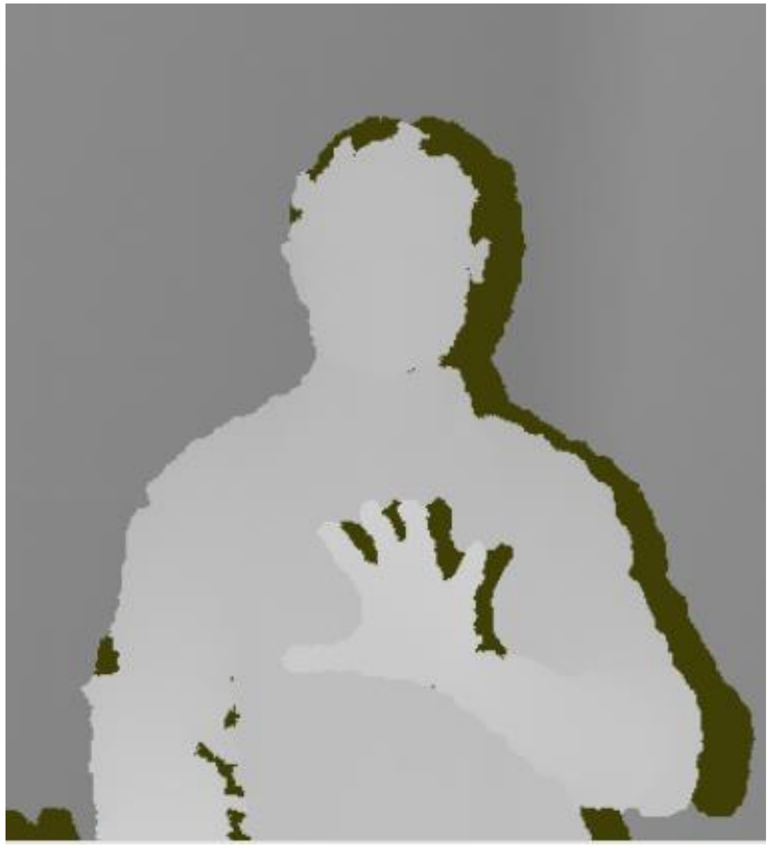

Left: Mirrored X


Right

Upside down device...


Upside down


Upside down
Mirrored X and Y


Right

Key Messages

Data isn’t the same as information

Testing ML requires domain knowledge

How bugs were resolved

More data (10x more for both training and validation)

- Final validation set: ~thousand of events x 100’s of thousands of frames
- Final accuracy (F-score): ~90%

Microsoft Research tools, knowledge, resources

- Multiplication of data (rotation)
- Speeding up execution (skipping pixels)
- Some “filters” on state change

Improvements demo

Experience that you can reuse

- Your team needs the right mix of skills and interests
- Have management buy-in

- Browse through ML training online
- Focus on the right problem
- Implement a heuristic solution before using ML
- Have a test plan for your tools and solution

Microsoft