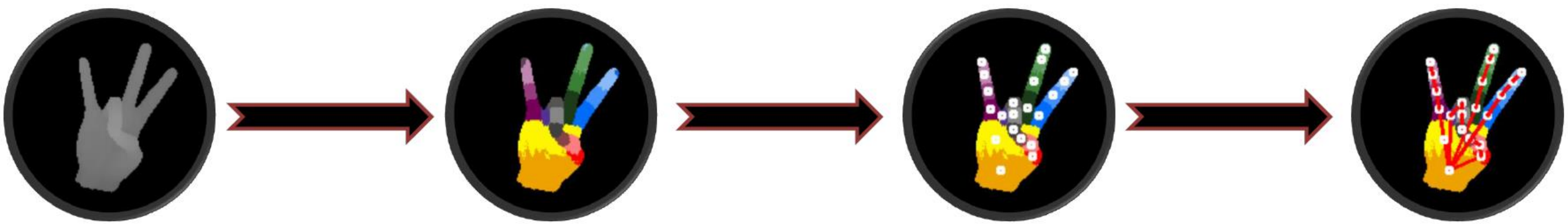
Gesture Recognition: Hand Pose Estimation

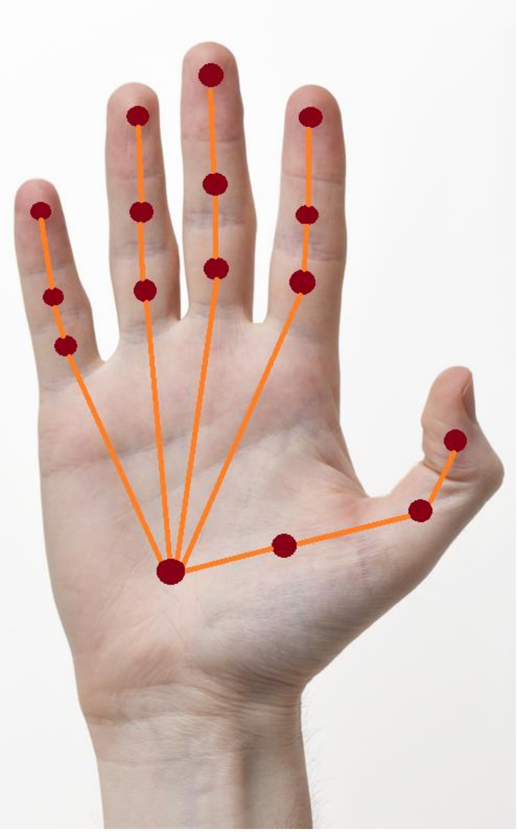


Adrian Spurr
Ubiquitous Computing Seminar FS2014
27.05.2014

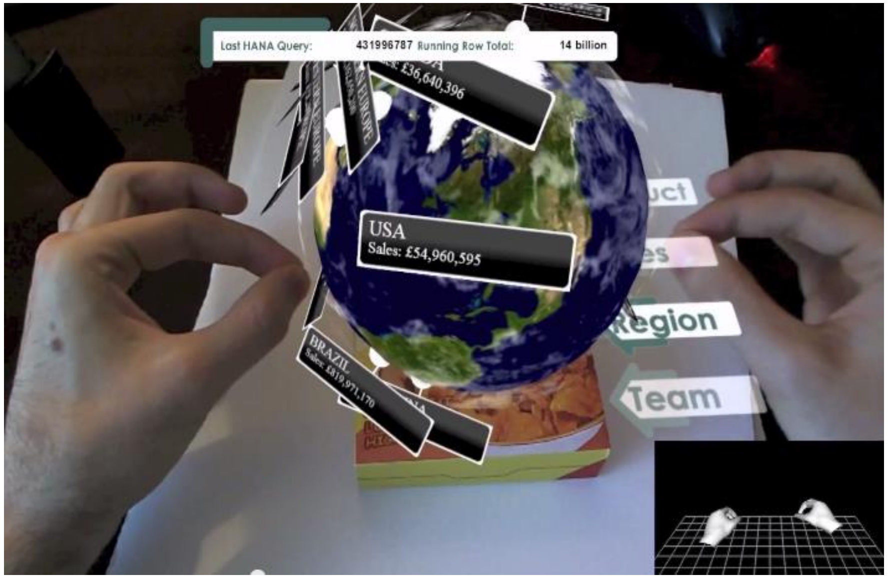
What is hand pose estimation?

Input

Computer-usable form

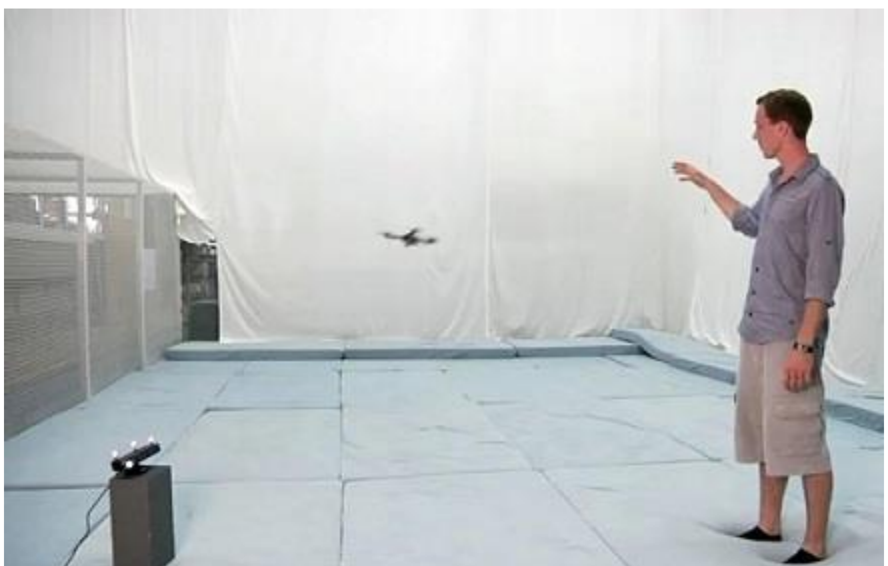


Augmented Reality



Gaming

Robot Control



PC Control

Data glove

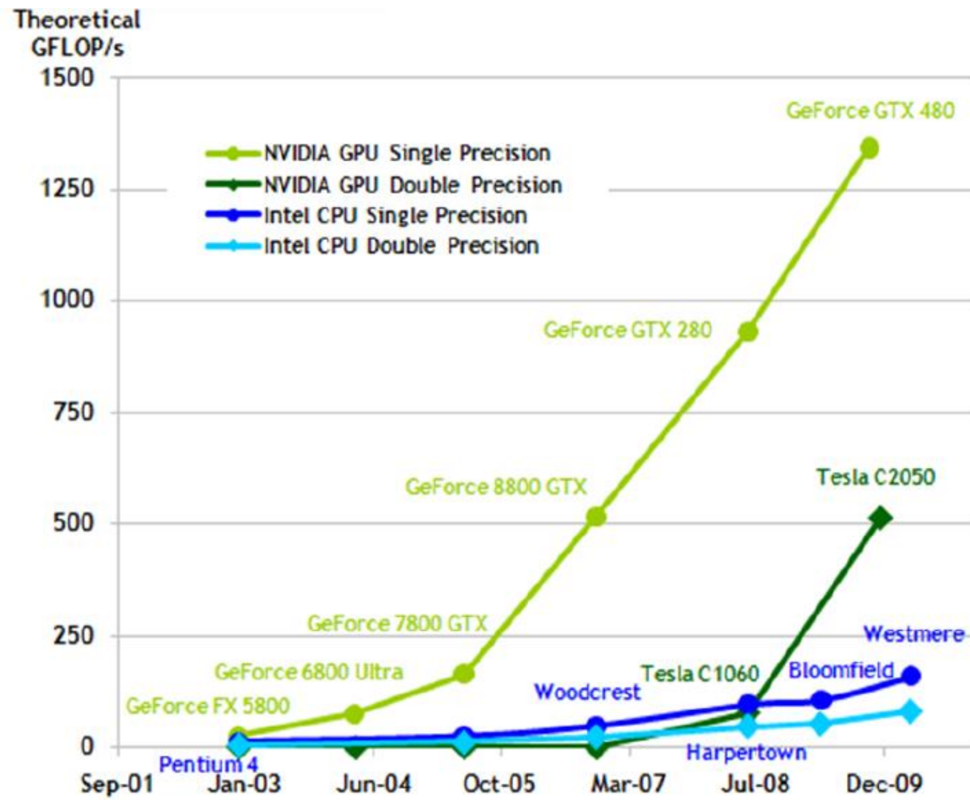
- Utilizes optical flex sensors to measure finger bending.
- Advantage: High accuracy, can provide haptic feedback.
- Disadvantages: invasive, long calibration time, unnatural feeling, heavily instrumented.

Thanks to cheap depth cameras...

Depth Camera

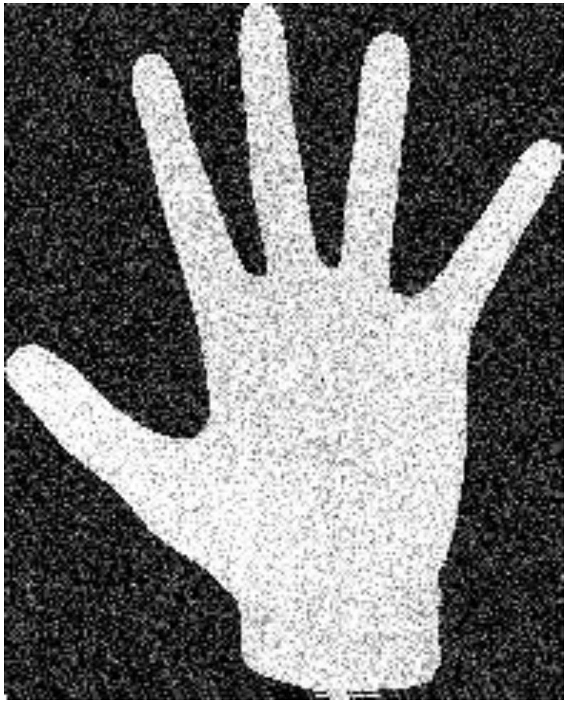
RGB Camera

...and increase in GPU Power



Problems occuring

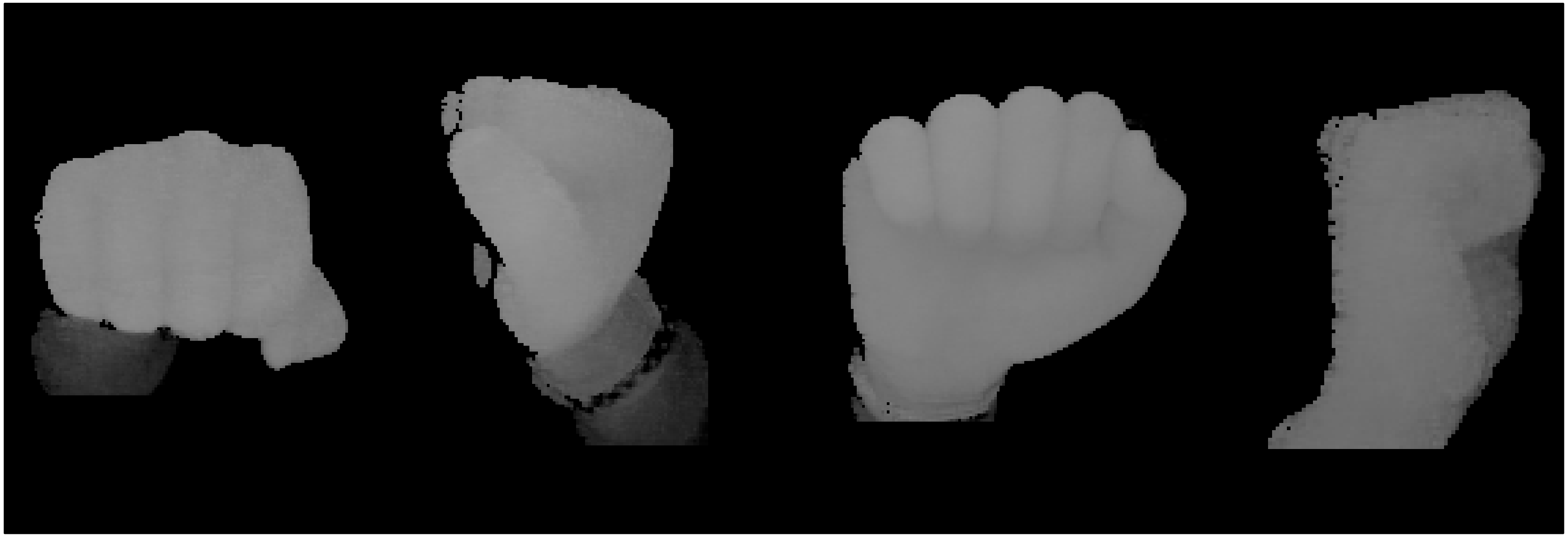
- Noisy data



- Segmentation

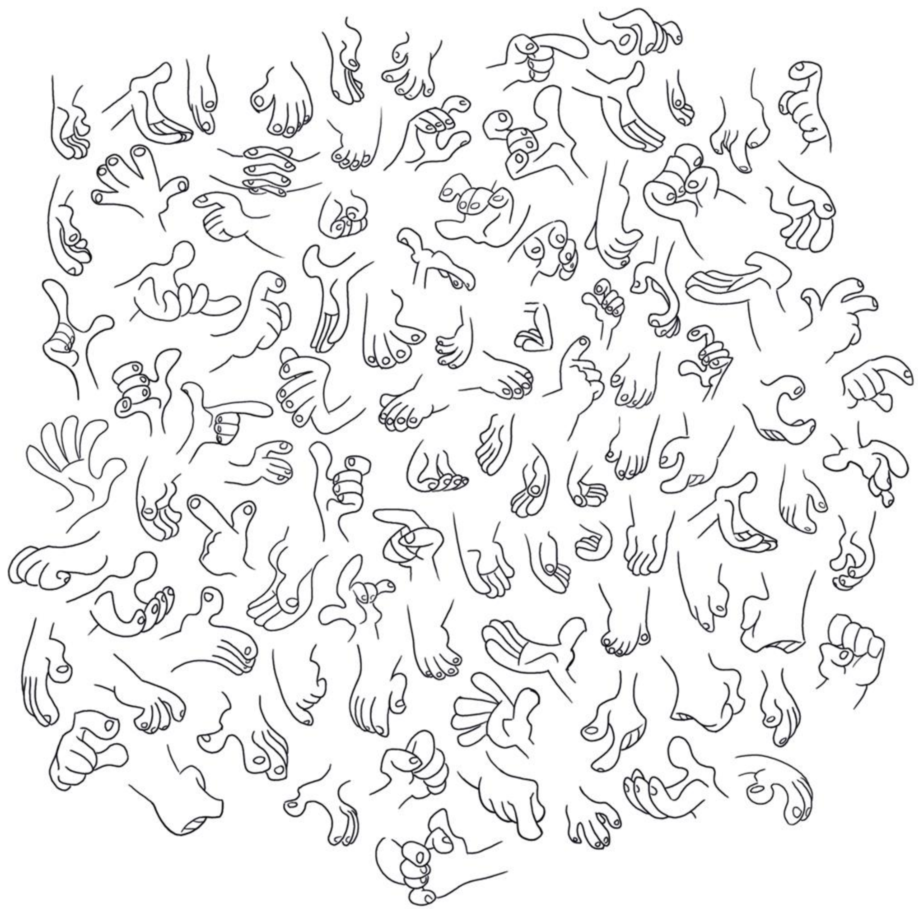
Problems occuring

- Self-occlusion and viewpoint change:



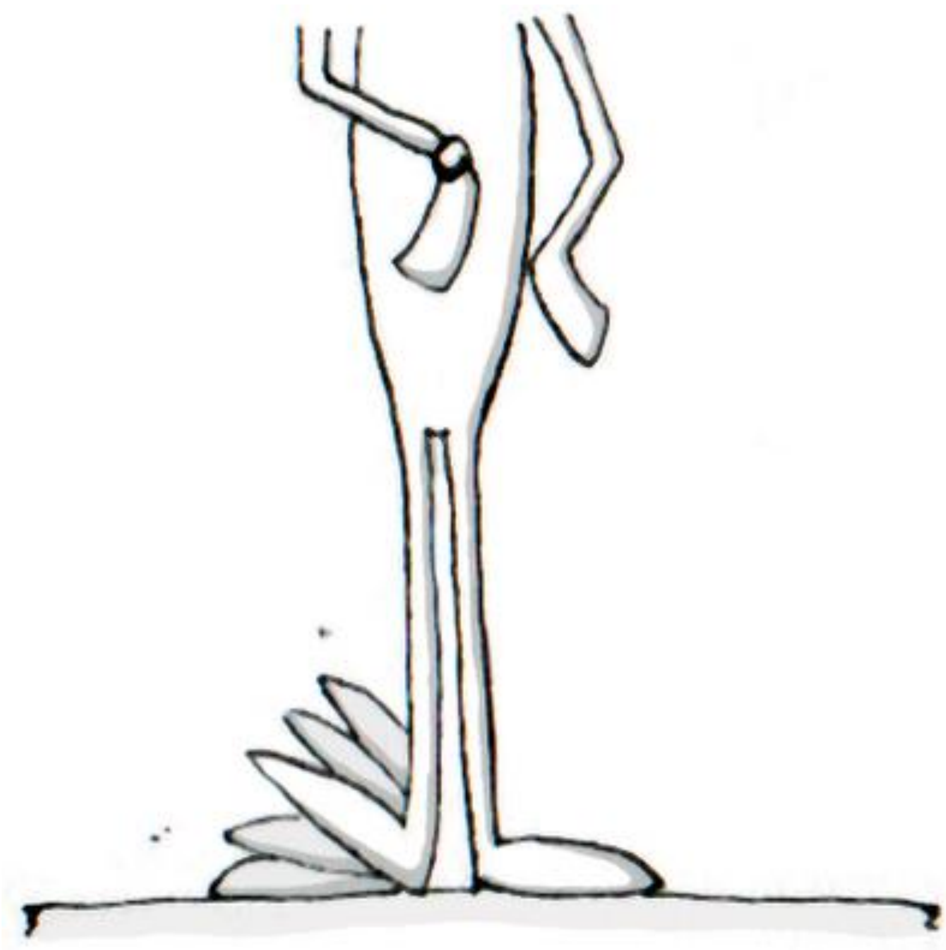
Problems occuring

- 27 Degrees of freedom per hand -> 280 trillion hand poses:

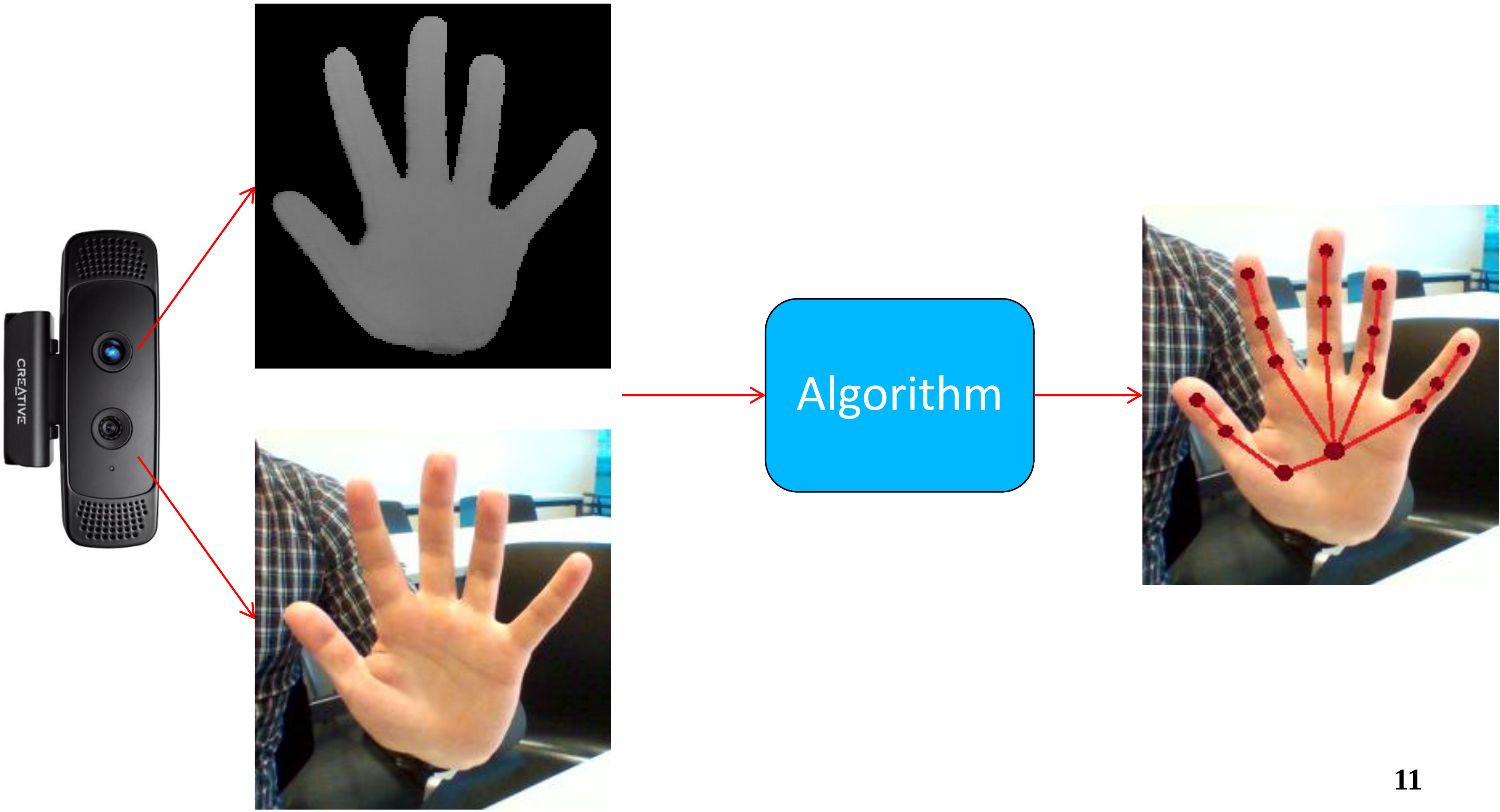


Problems occuring

- Performance: For practical use, must be real time.

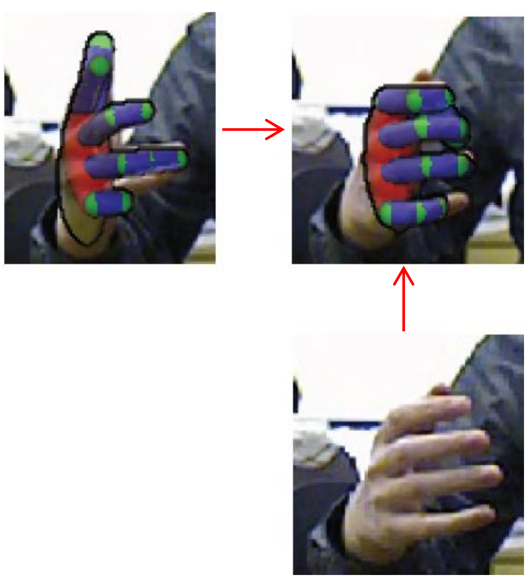


Principle of operation

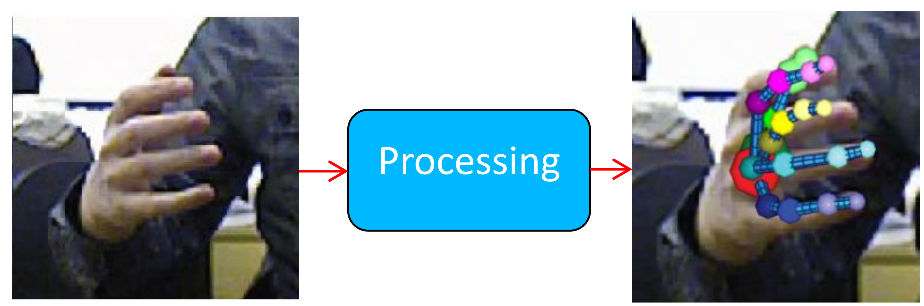


Existing schools of thought

- Model-based:
  - Keeps internally track of current pose.
  - Updates pose according to current pose and observation.

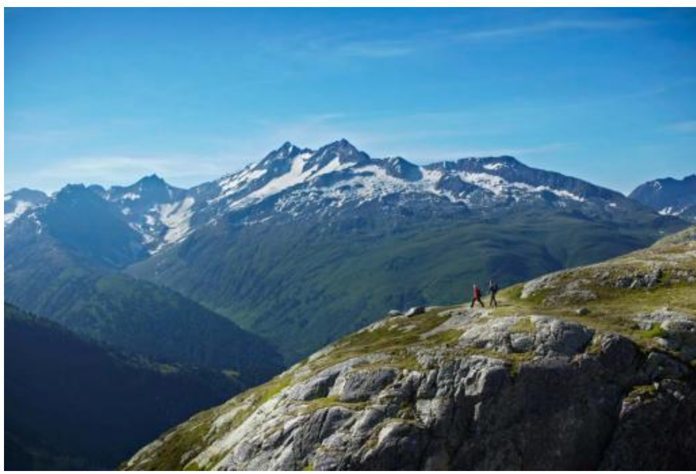


- Discriminative:
  - Maps directly from observation to pose.
  - “Learn” from training data and apply knowledge to unseen data.

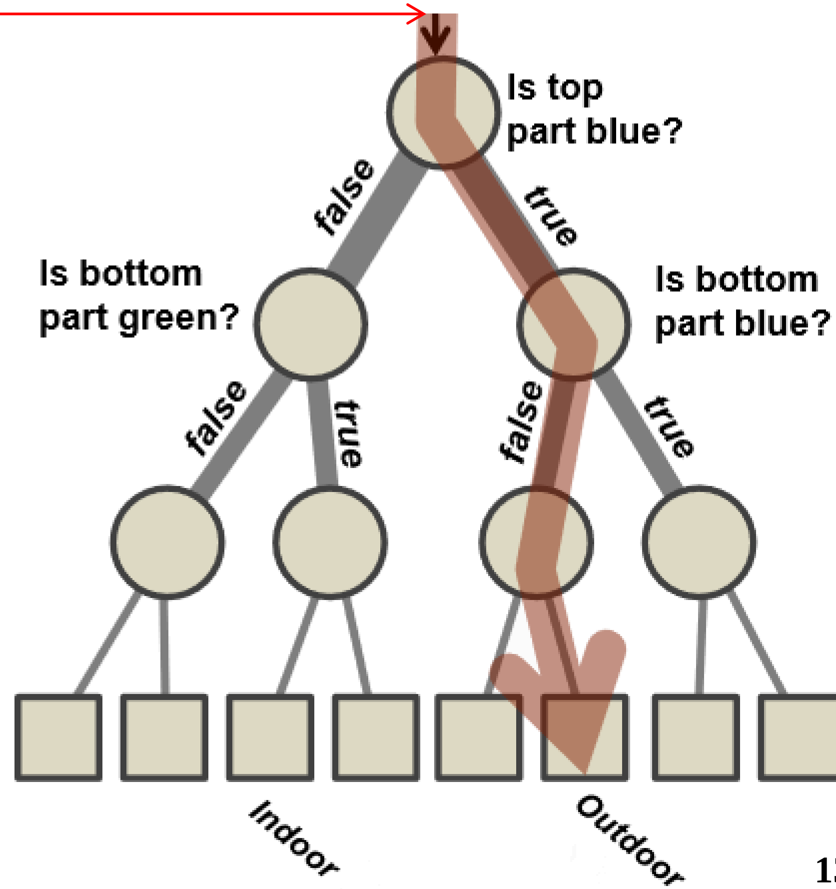


Short intro to Random Forests

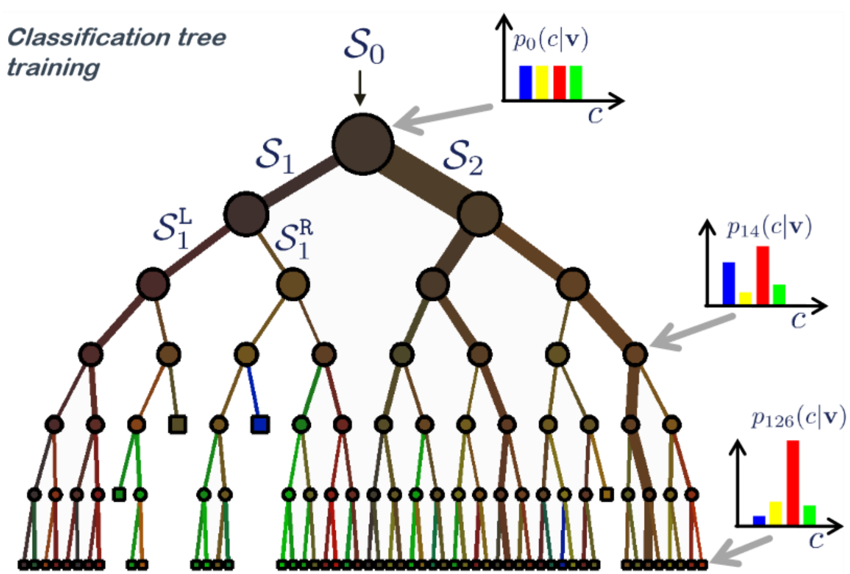
- Ensemble learning
- Classification and Regression
- Consists of decision trees

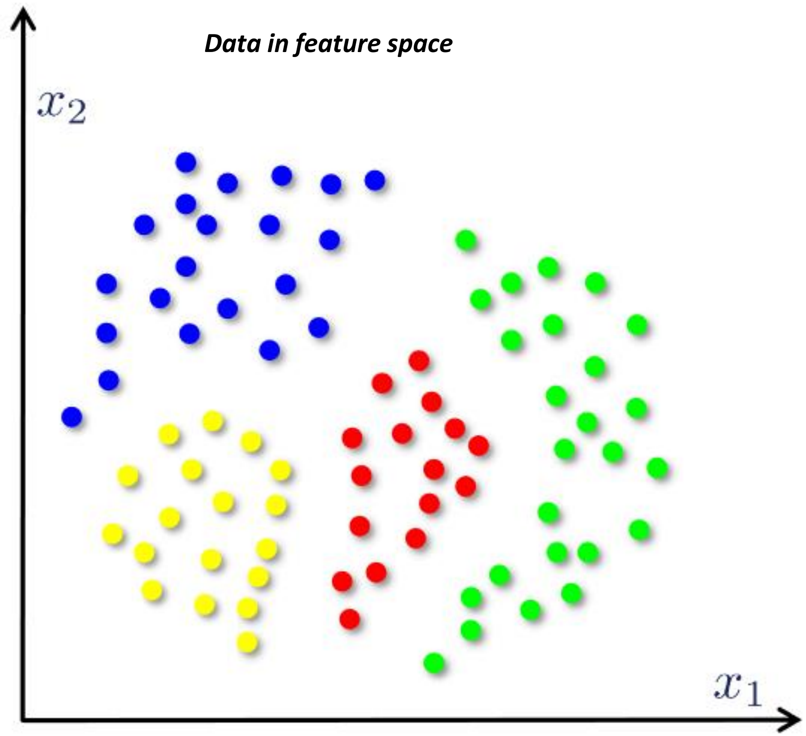


A decision tree:



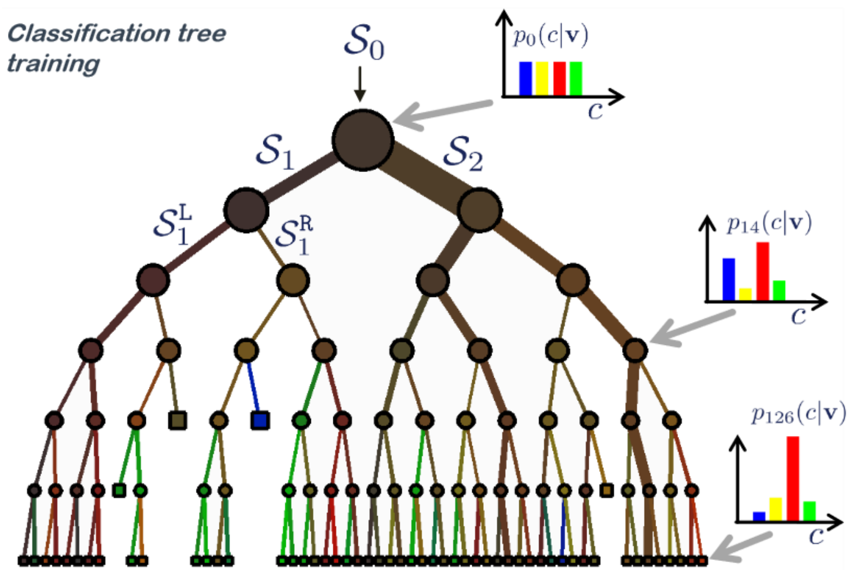
Short intro to Random Forests

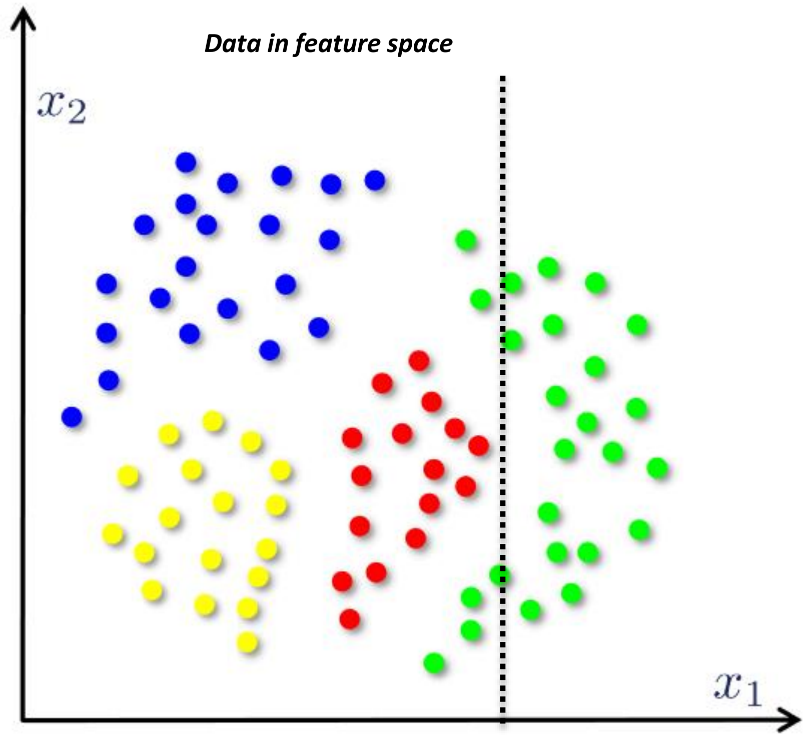




Features = «Properties» of data

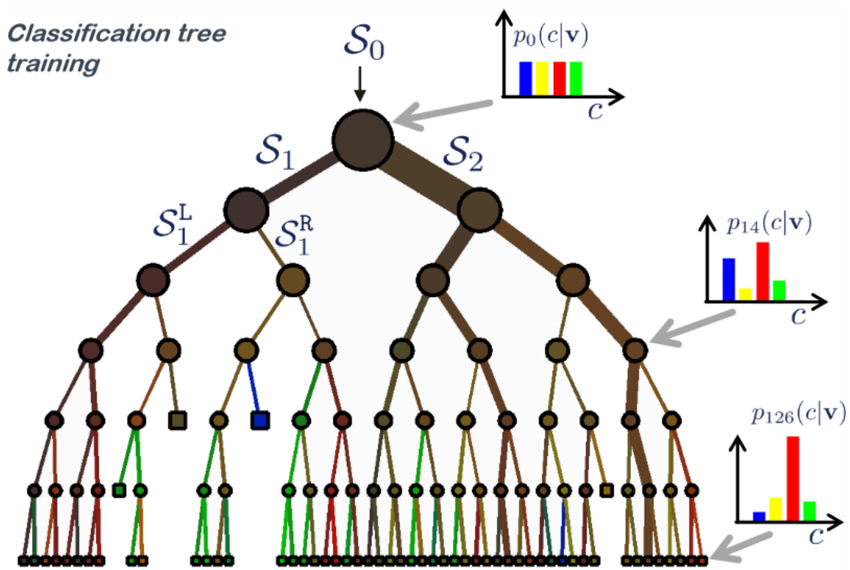
Short intro to Random Forests

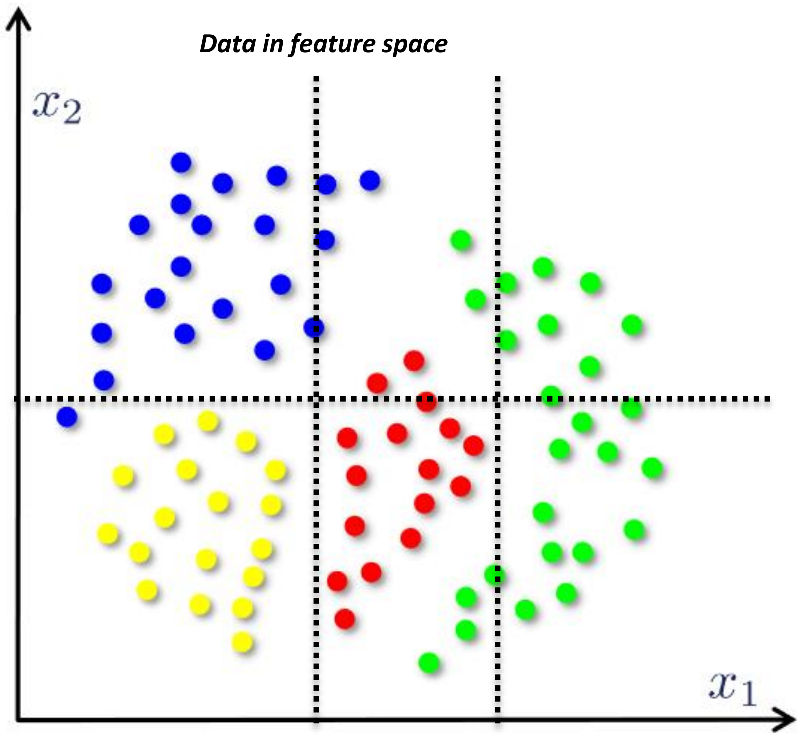




Features = «Properties» of data

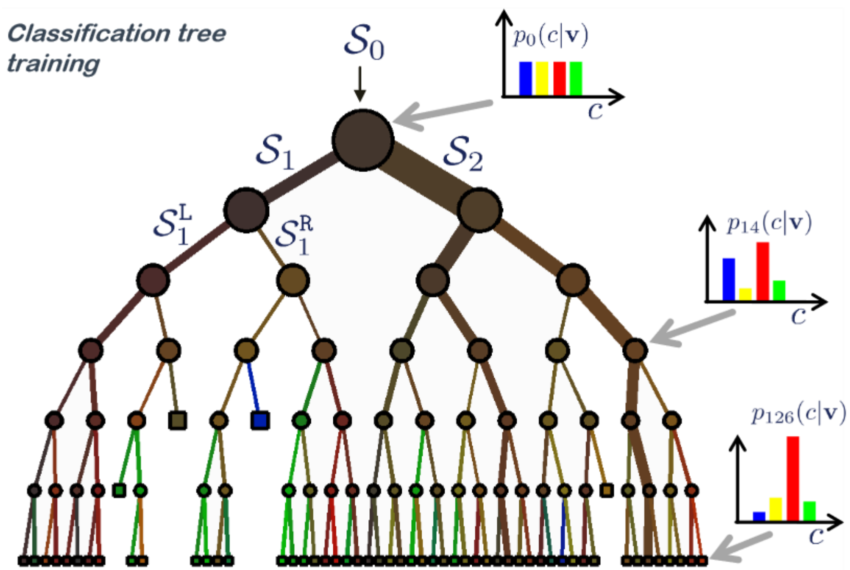
Short intro to Random Forests

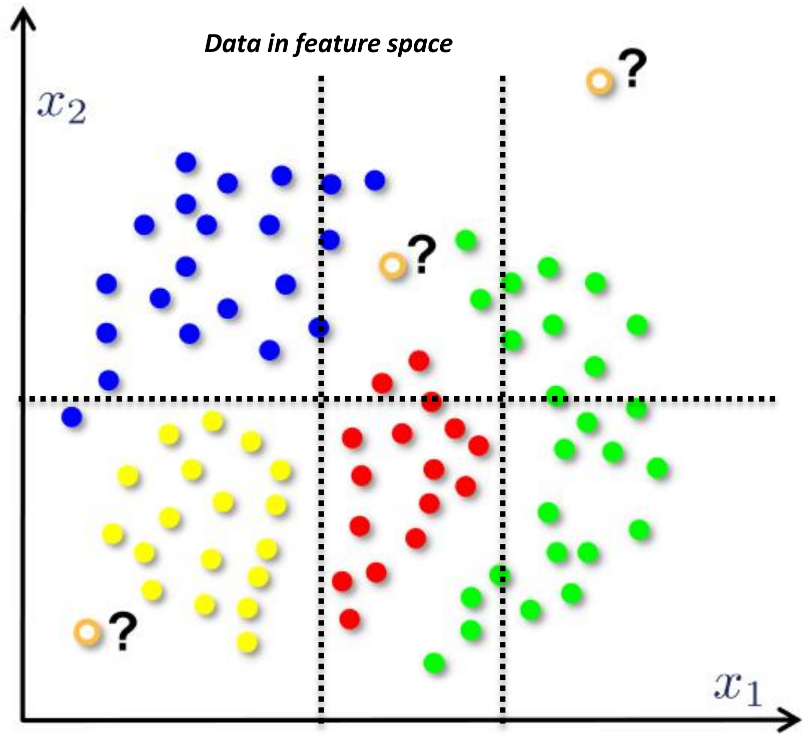




Features = «Properties» of data

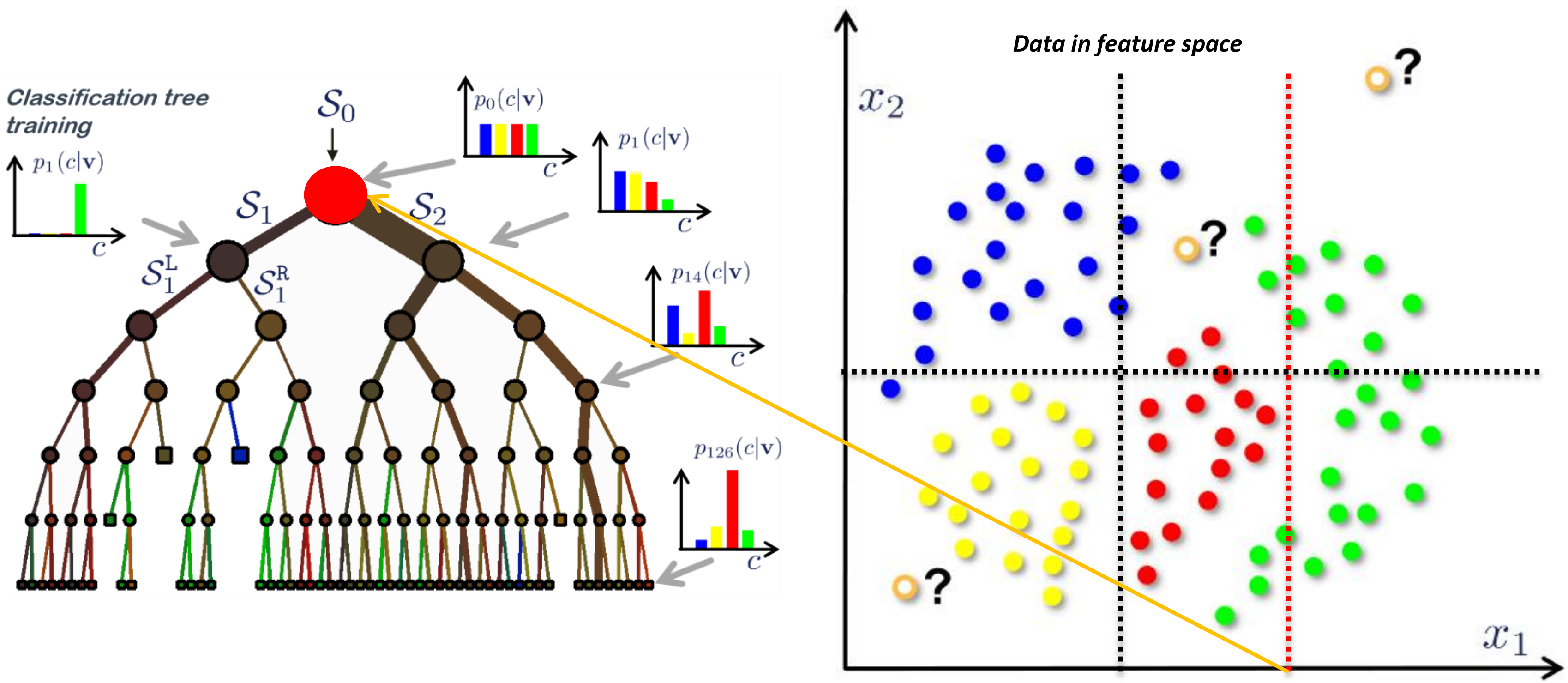
Short intro to Random Forests





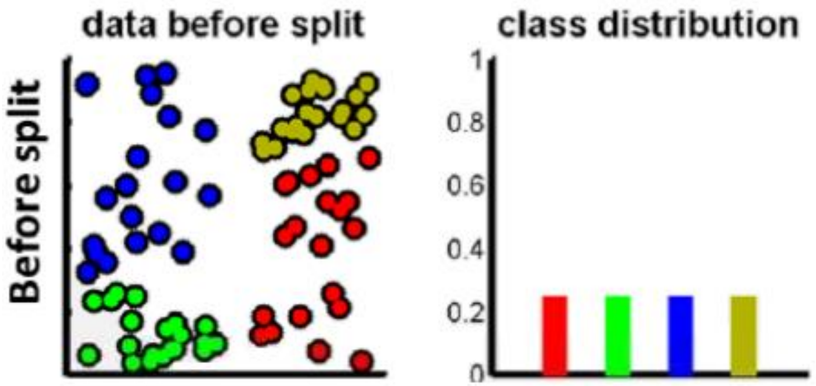
Features = «Properties» of data

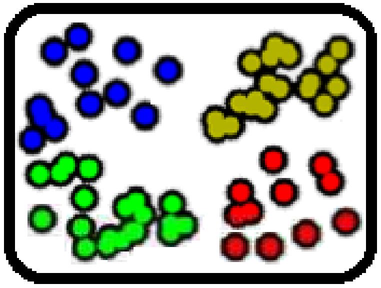
Short intro to Random Forests



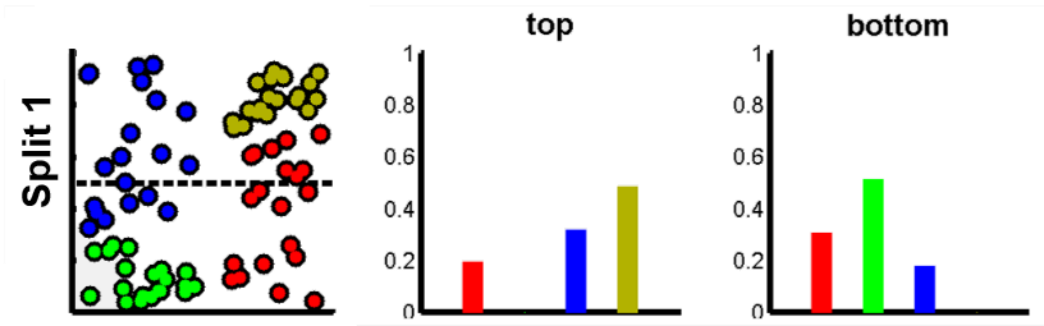
Features = «Properties» of data

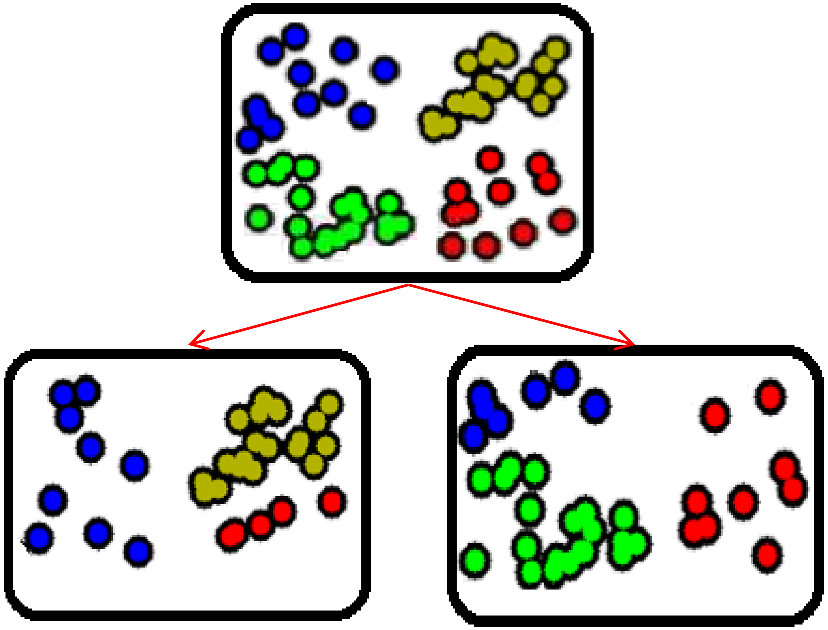
Building a classification tree



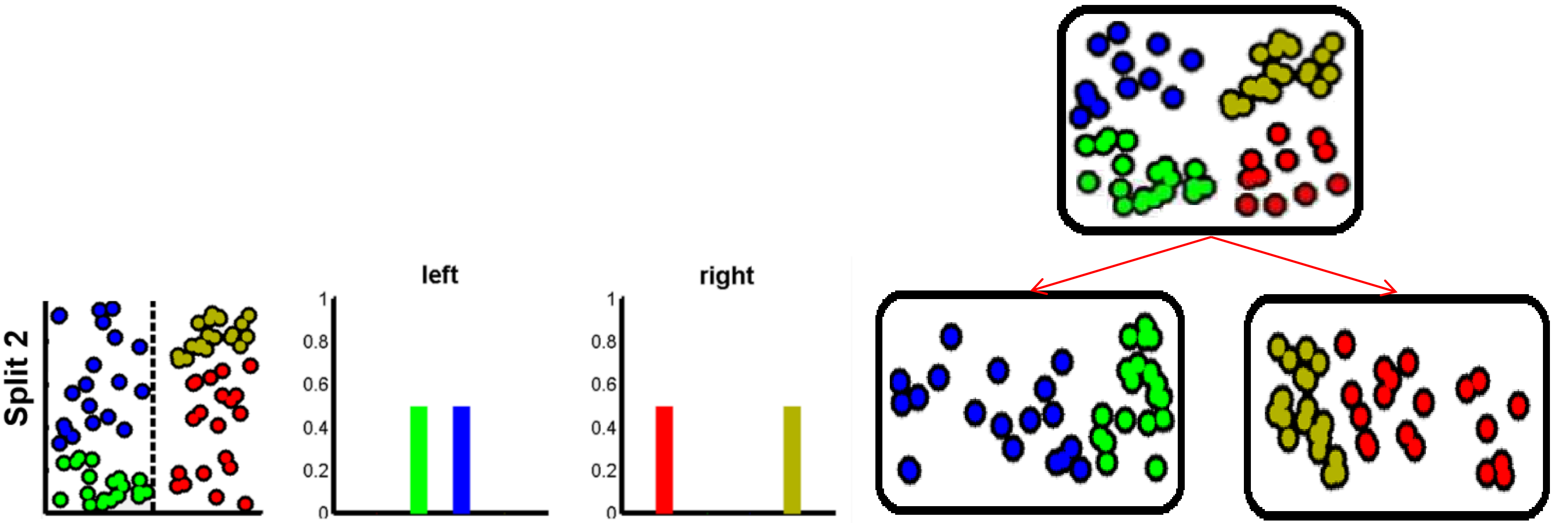


Building a classification tree





Building a classification tree



Random feature sampling

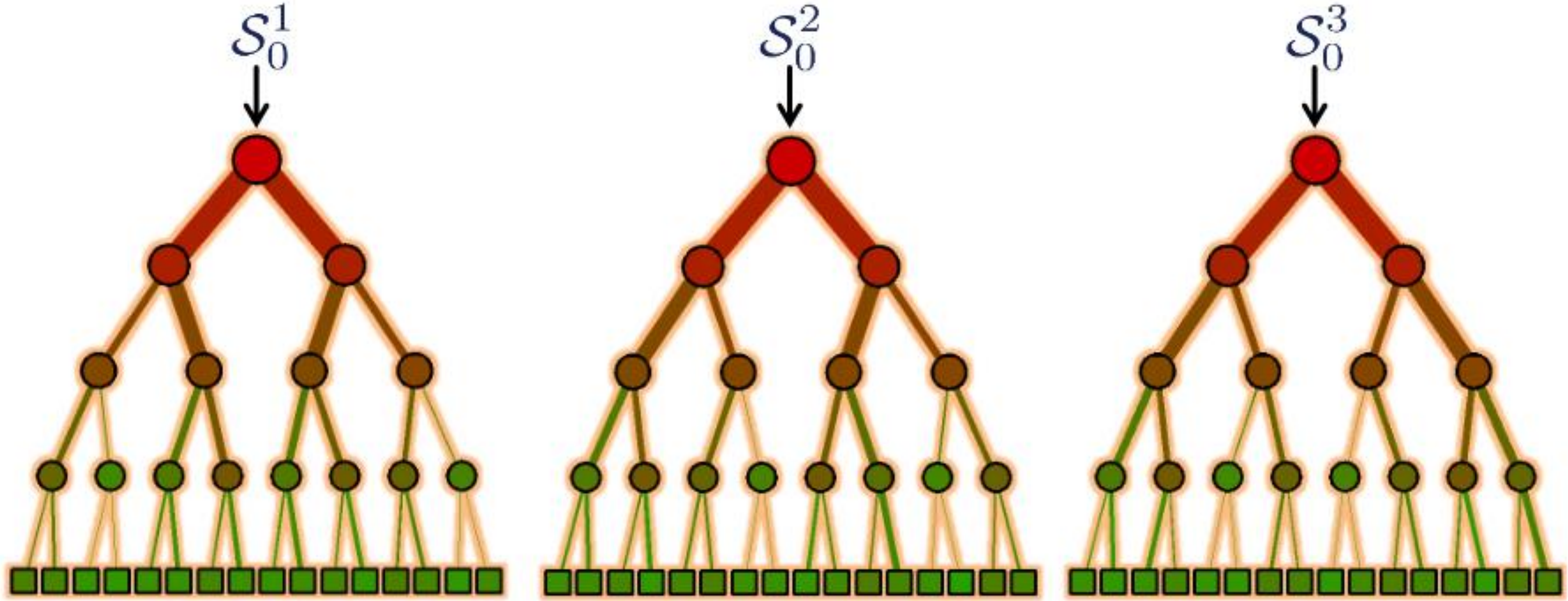
$\mathcal{T}$	The full set of all possible node test parameters
$\mathcal{T}_j \subset \mathcal{T}$	For each node the set of randomly sampled features
$\rho = \lvert\mathcal{T}_j\rvert$	Randomness control parameter. For $\rho = \lvert\tau\rvert$ no randomness and maximum tree correlation For $\rho = 1$ max randomness and minimum tree correlation

Choose T_j which splits the data with maximum information gain.

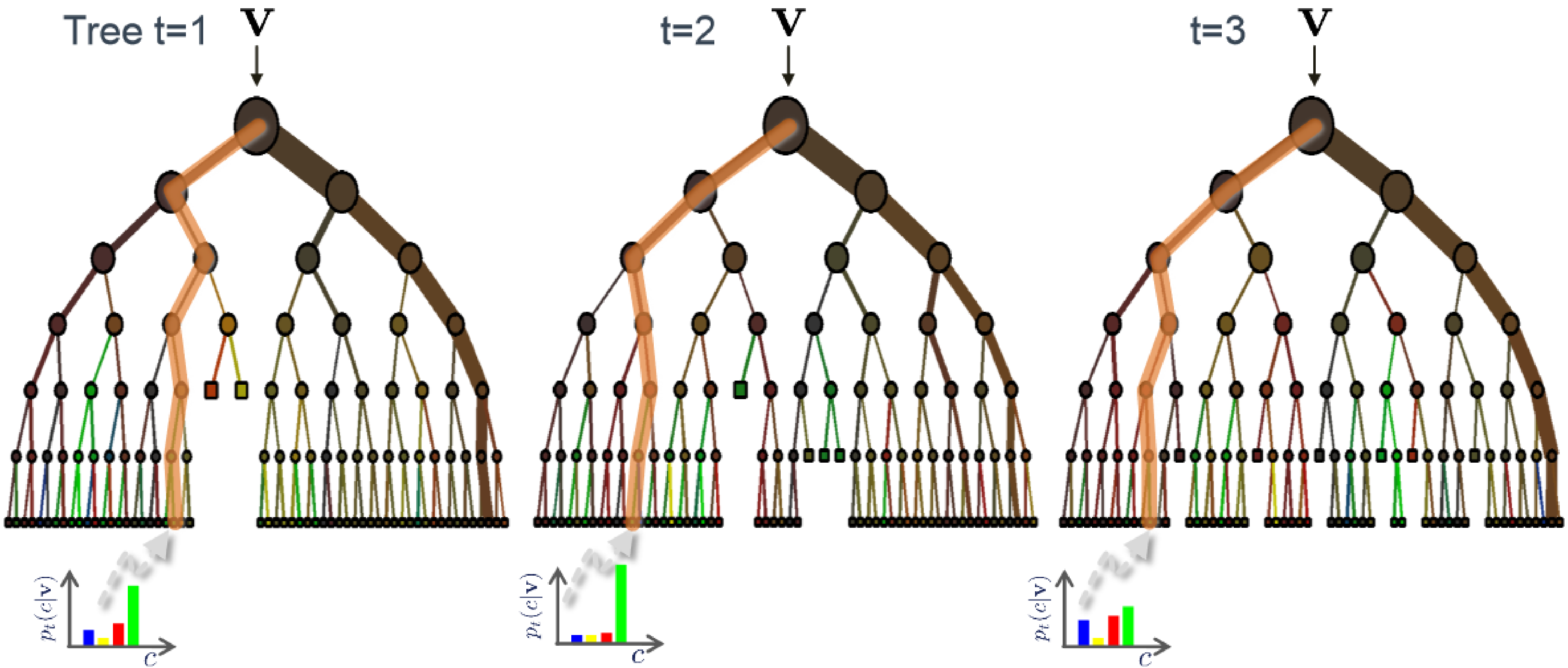
Bagging

S_0 Full training set

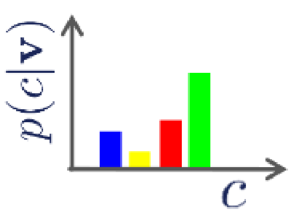
$S_0^t \subset S_0$ Randomly sampled subset



Prediction



Forest output probability $$p(c|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(c|\mathbf{v})$$

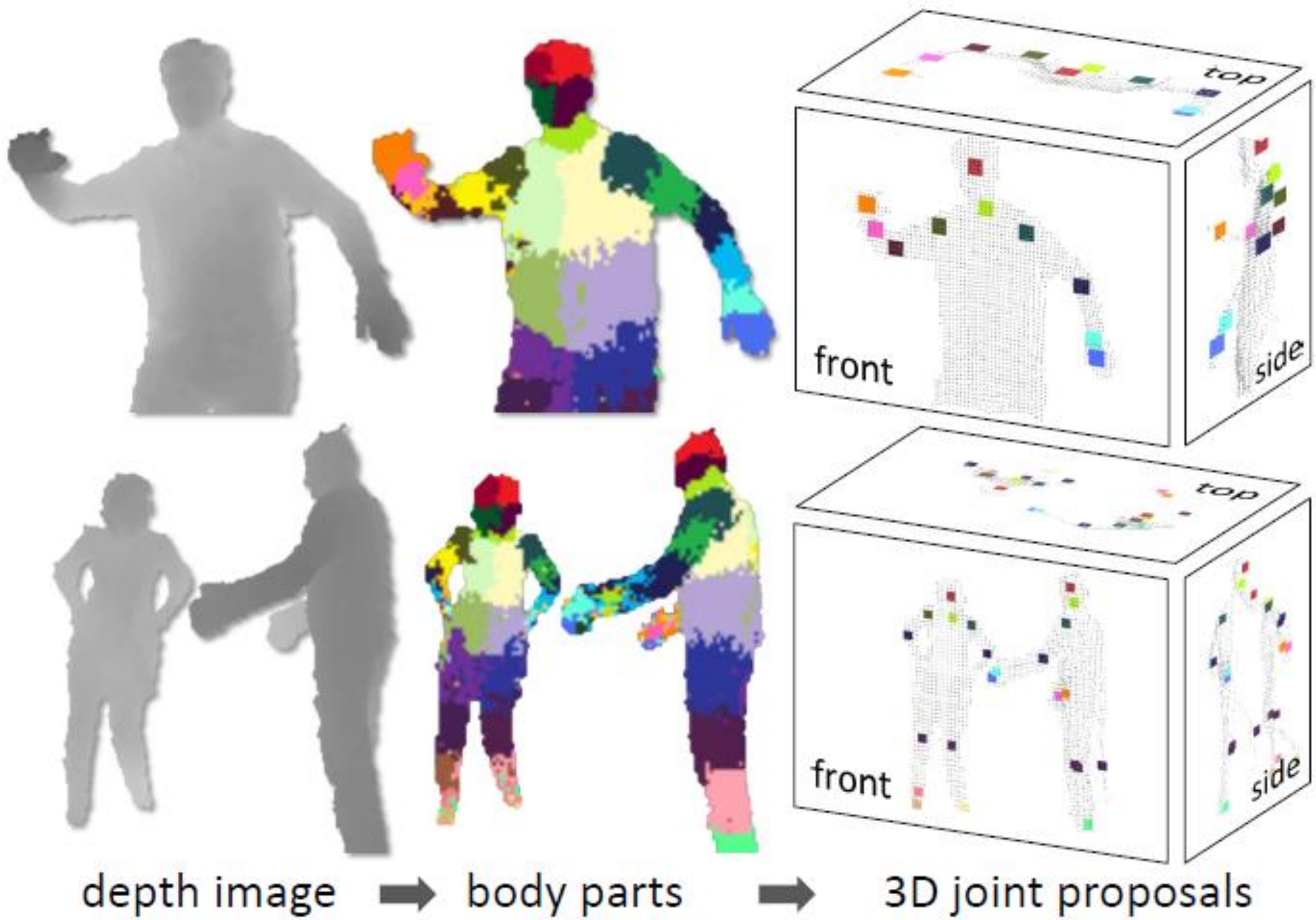


RF for pose estimation

Why Random Forests?
- Robust
- Fast
- Thorougly studied

How should we use them?
- Must choose what to split on.
- What should the labels be?

Advanced body pose recognition



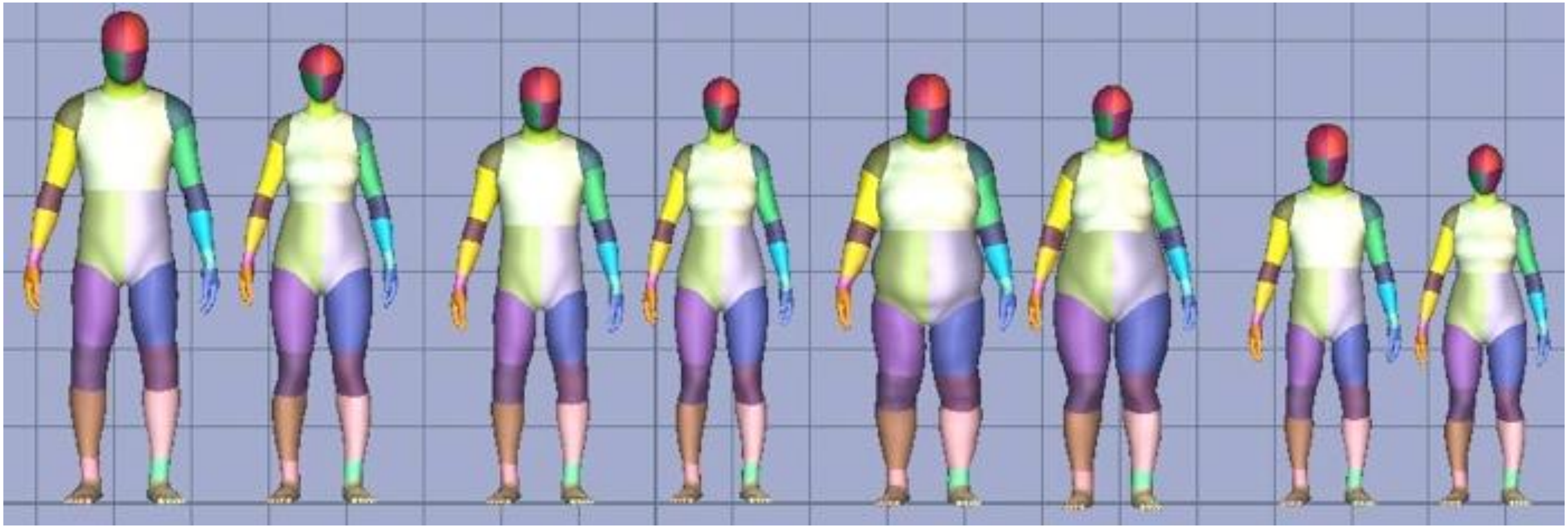
[Shotton2011]

Advanced body pose recognition

- Discriminative approach.
- Used in the Kinect.
- First paper to use synthetic training data.
- Basis for many future papers.

[Shotton2011]

Creating synthetic data



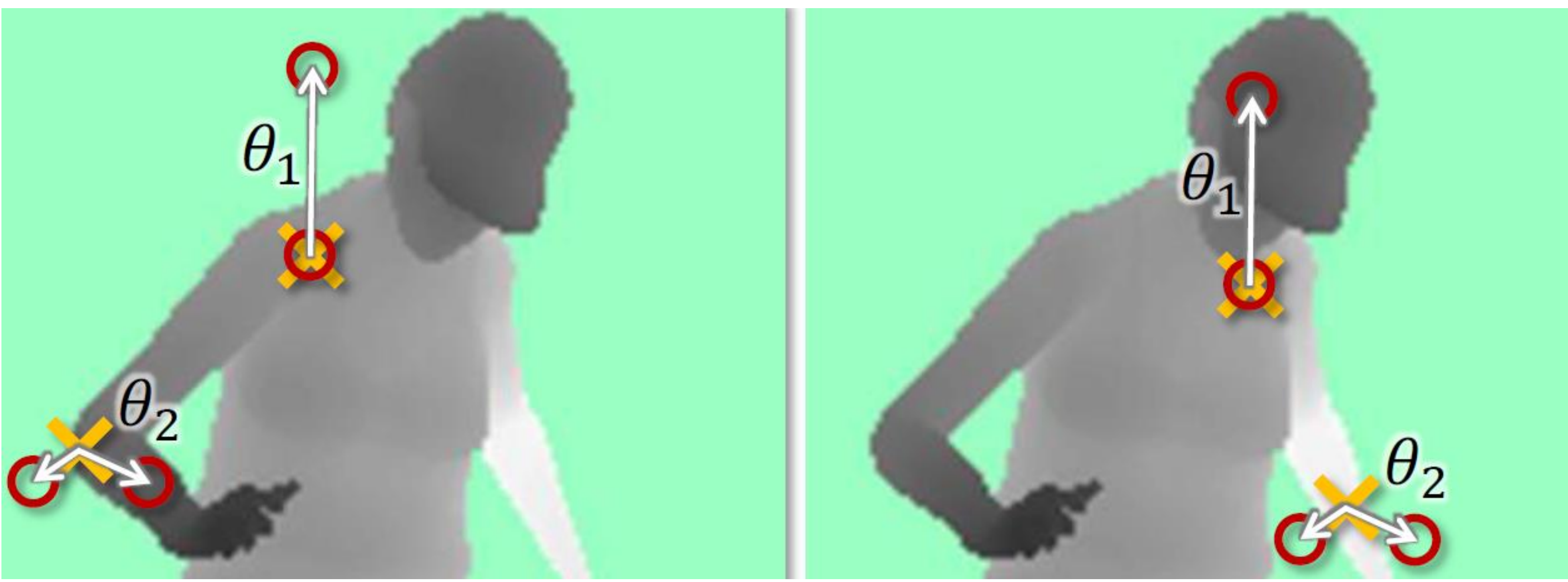
[Shotton2011]

Split funtion

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

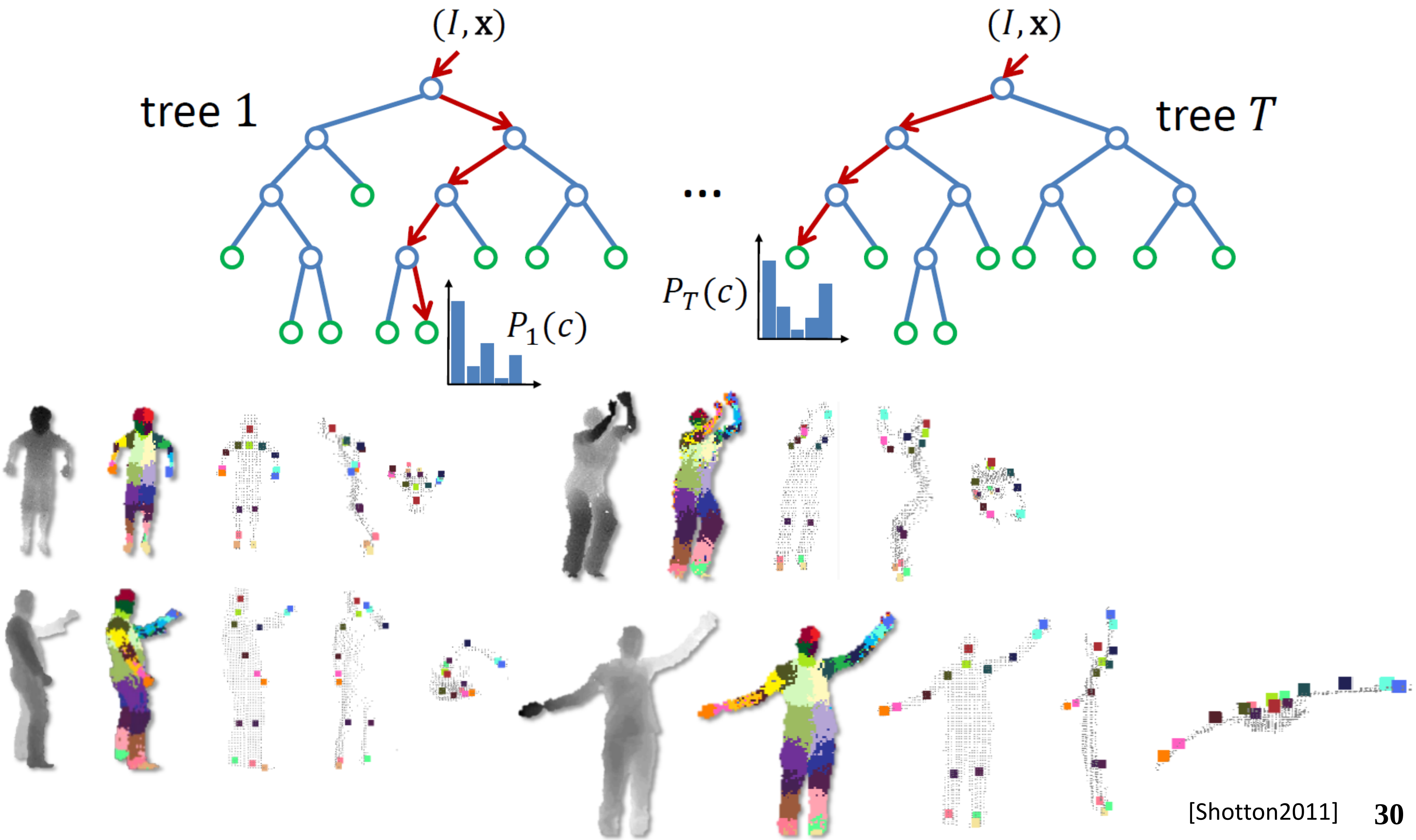
$d_I(\mathbf{x})$: Depth at position x

$$\theta = (\mathbf{u}, \mathbf{v})$$



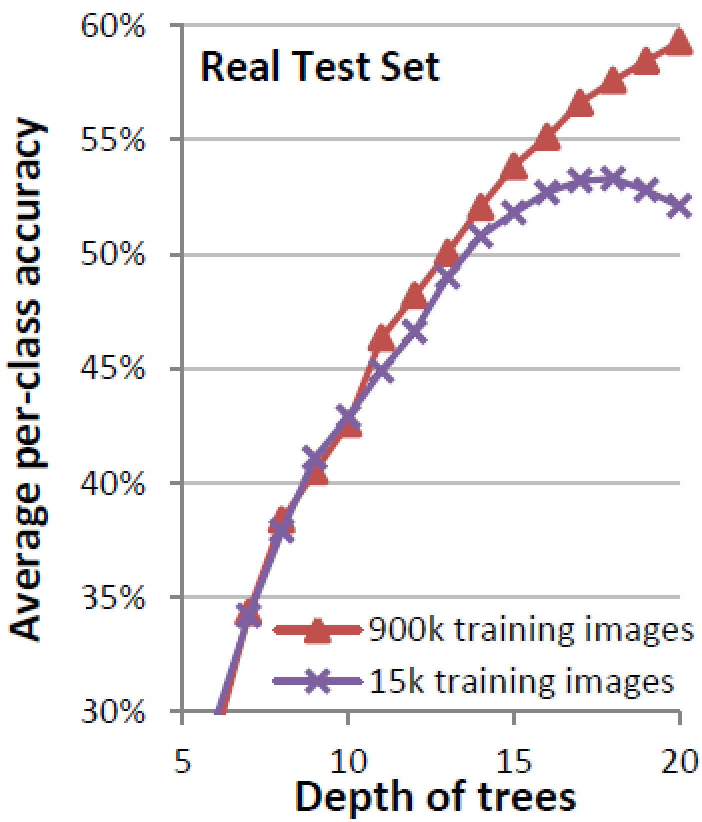
[Shotton2011]

Joint prediction



[Shotton2011]

Per-class accuracy vs. tree depth

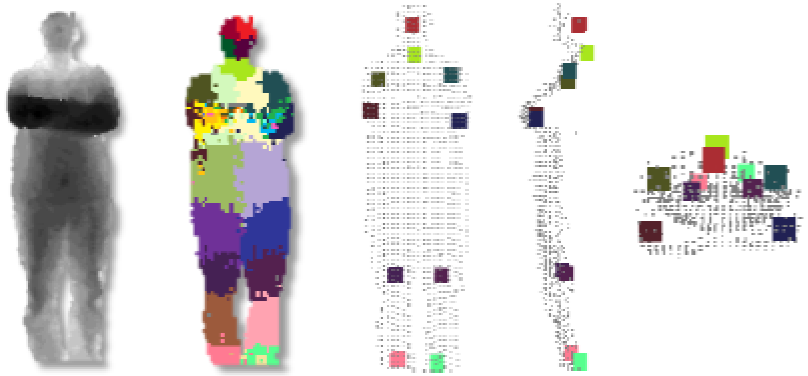


- Accuracy increases as depth of tree increases.
- Overfitting occurs for 15k training images.
- More training images leads to higher accuracy and less overfitting.

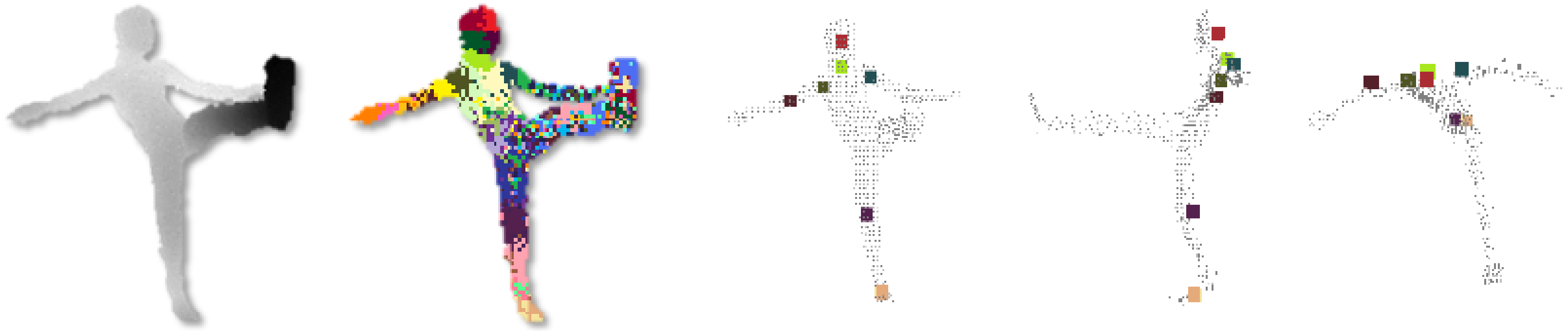
[Shotton2011]

Negative Results

- Failure due to self-occlusion:



- Failure due to unseen pose:



[Shotton2011]

Unresolved issues

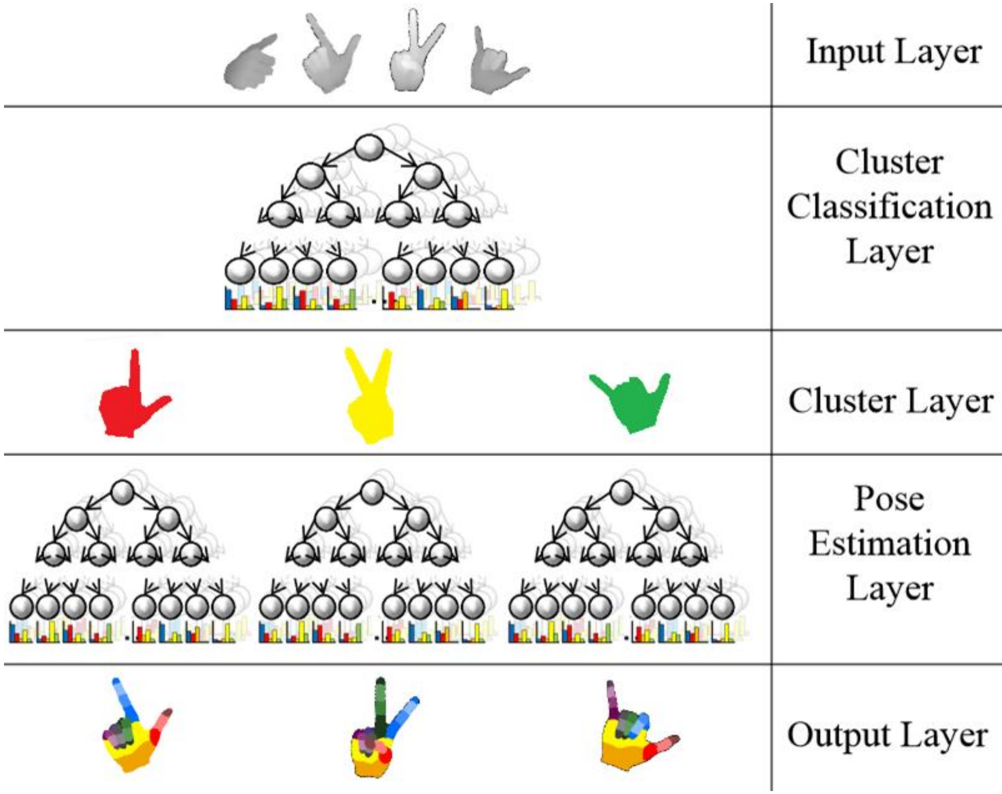
- To capture all possible poses, need to generate huge amount of training data.
- Training RF on big training set means more trees and deeper trees.
- Big amount of memory needed.

Unresolved issues

- To capture all possible poses, need to generate huge amount of training data.
- Training RF on big training set means more trees and deeper trees.
- Big amount of memory needed.
- Solution: Divide training data into sub-sets and solve classification for each set separately.

Multi-layered Random Forest

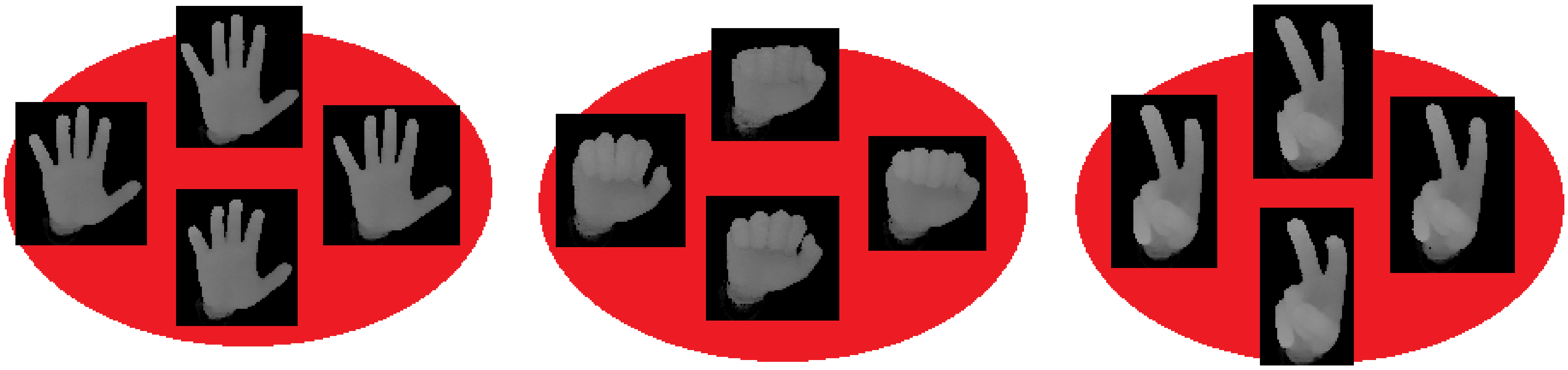
- Cluster training data based on similarity.
- Train RF on and for each cluster.
- First layer assigns input to proper cluster.
- Second layer gives the final hand part label distribution.



[Keskin2012]

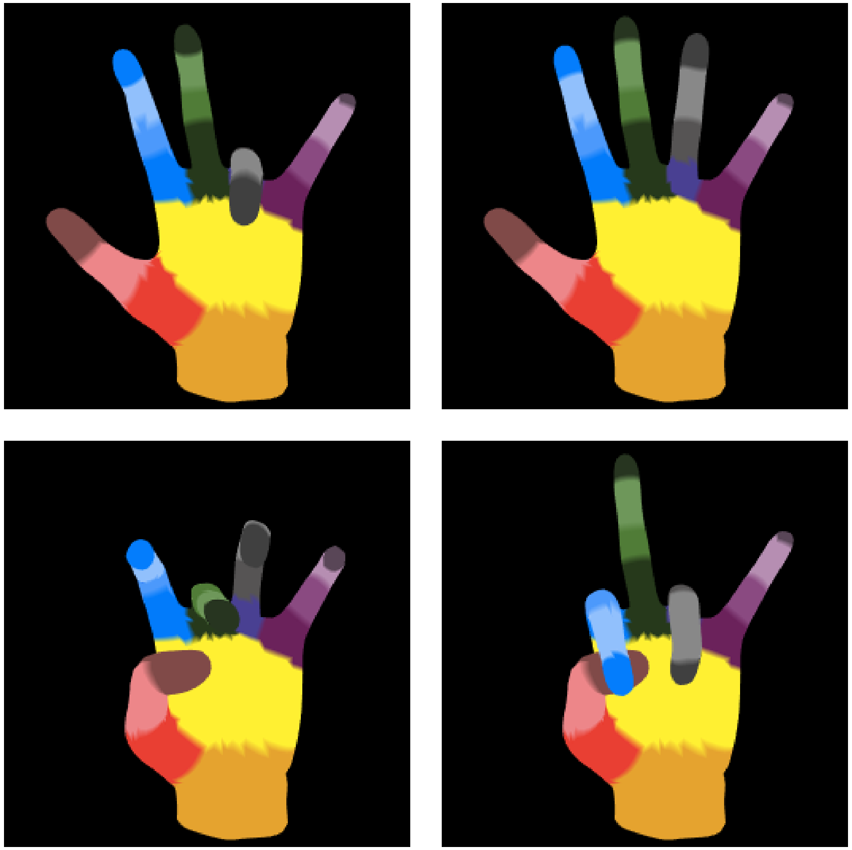
Clustering training data

- Cluster based on weighted differences.
- Penalize differences of viewpoint, finger positions.
- Label each cluster, labels refer to hand shape.
- Train Random Forest on clusters.



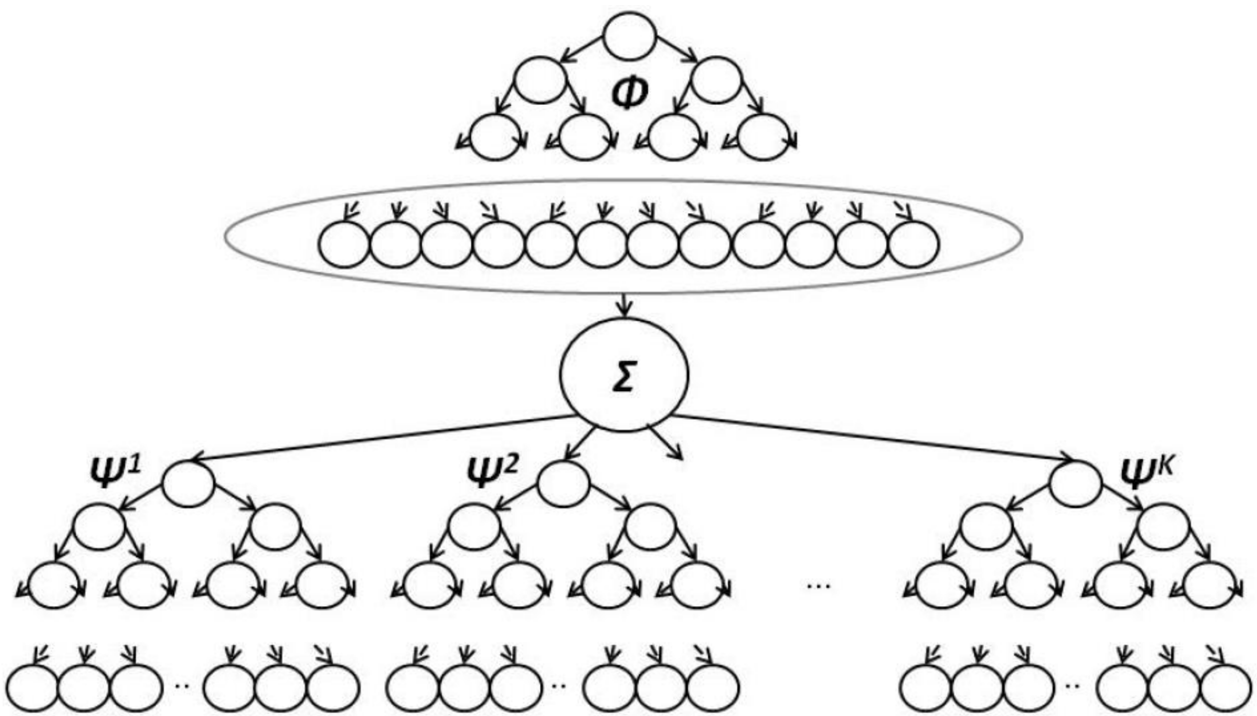
Experts

- Use hand part labels.
- Train for each cluster a separate Random Forest.
- Each forest is called Expert.



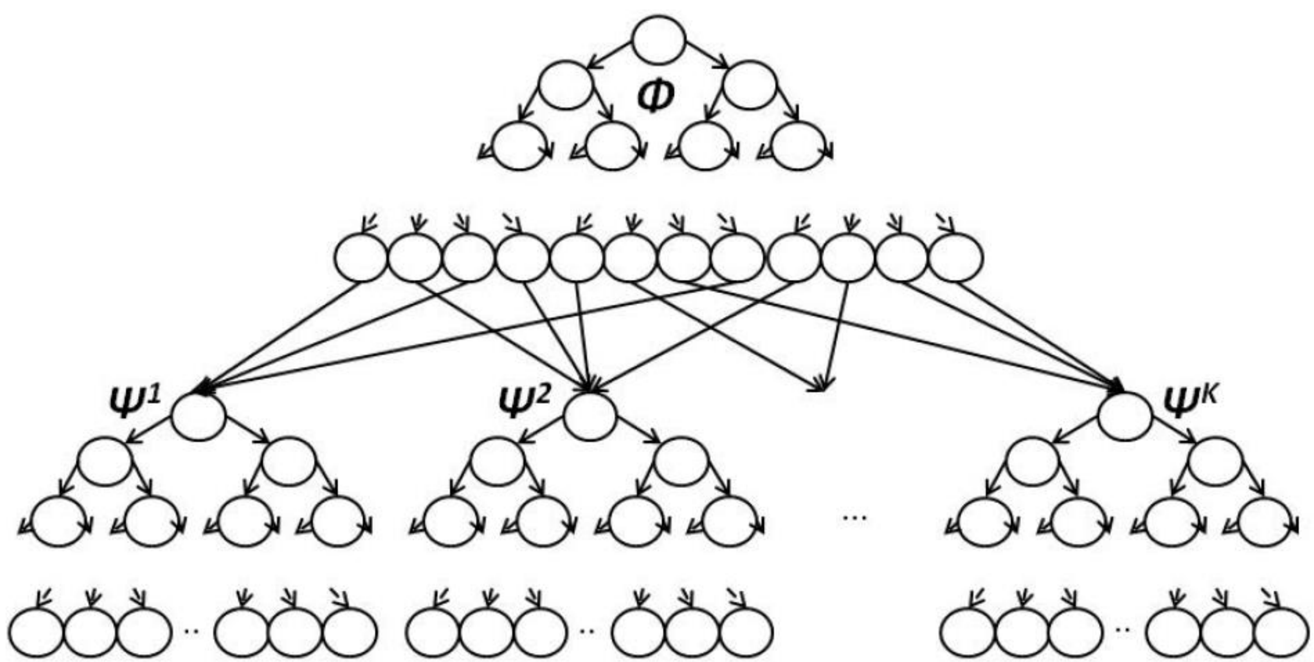
Two prediction methods

- Global Expert Network:
  - Feed input to first layer of Random Forest, average input, get hand shape label.
  - Feed input to corresponding expert, get hand part distribution.



Two prediction methods

- Local Expert Network
  - Feed input to first layer of Random Forest, get hand shape label for each pixel.
  - Feed each pixel to its corresponding expert, get hand part distribution.



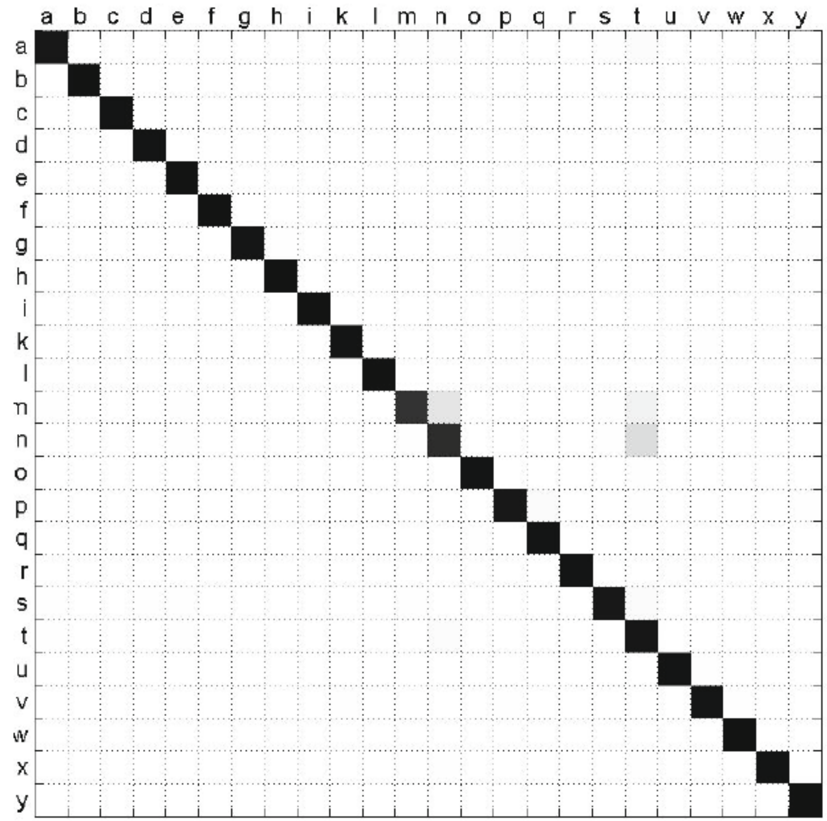
Parts distribution to pose

- RDF returns the hand part distribution.
- Get centre of each distribution by utilizing mean shift.

American Sign Language

First layer accuracy on ASL

- 2-fold cross-validation: 97.8%

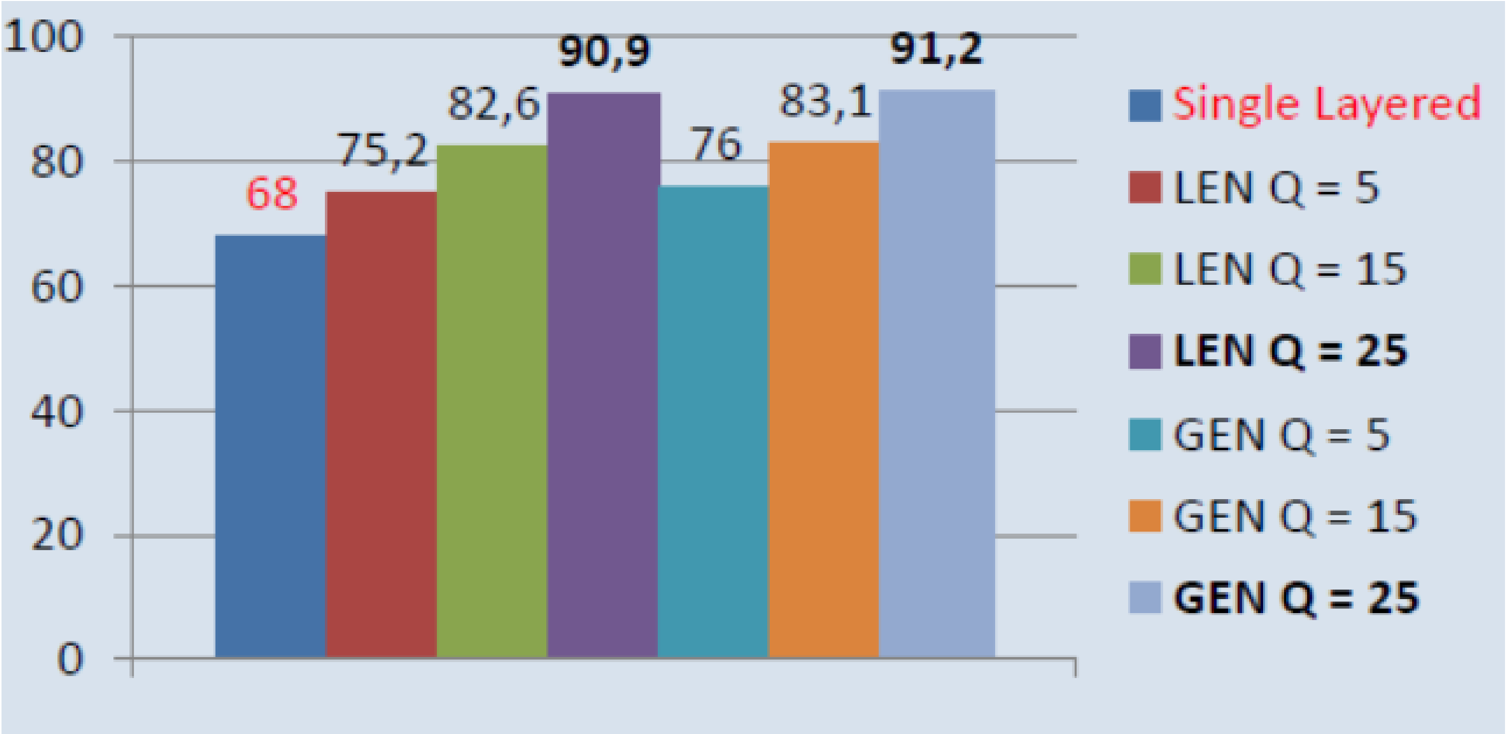


- Confusion occurs for (m,n), (m,t) and (n,t)

Confusions

- Confusion occurs for (m,n), (m,t) and (n,t)

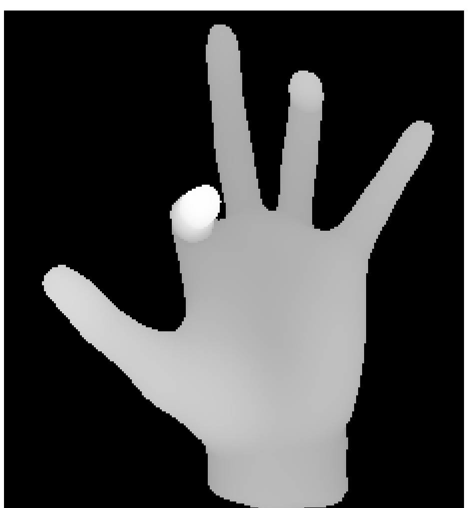
Second layer accuracy



Q = Number of clusters

Problems

- Not feasible to capture all possible variations of hand with synthetic data.
- Methods using only synthetic data suffer from synthetic-realistic discrepancies.
- But: Using realistic training data expensive, due to manually labelling them.



Synthetic

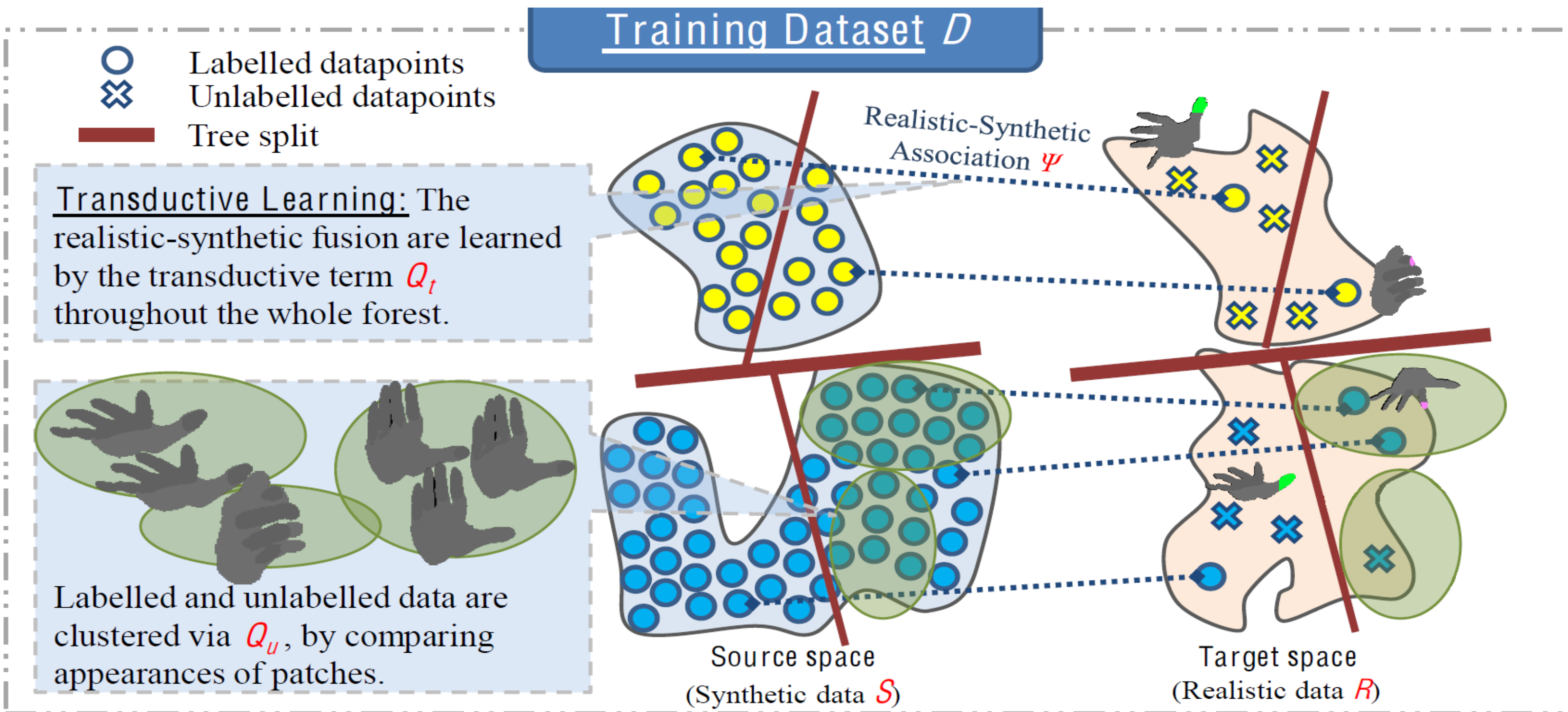
Real

Problems

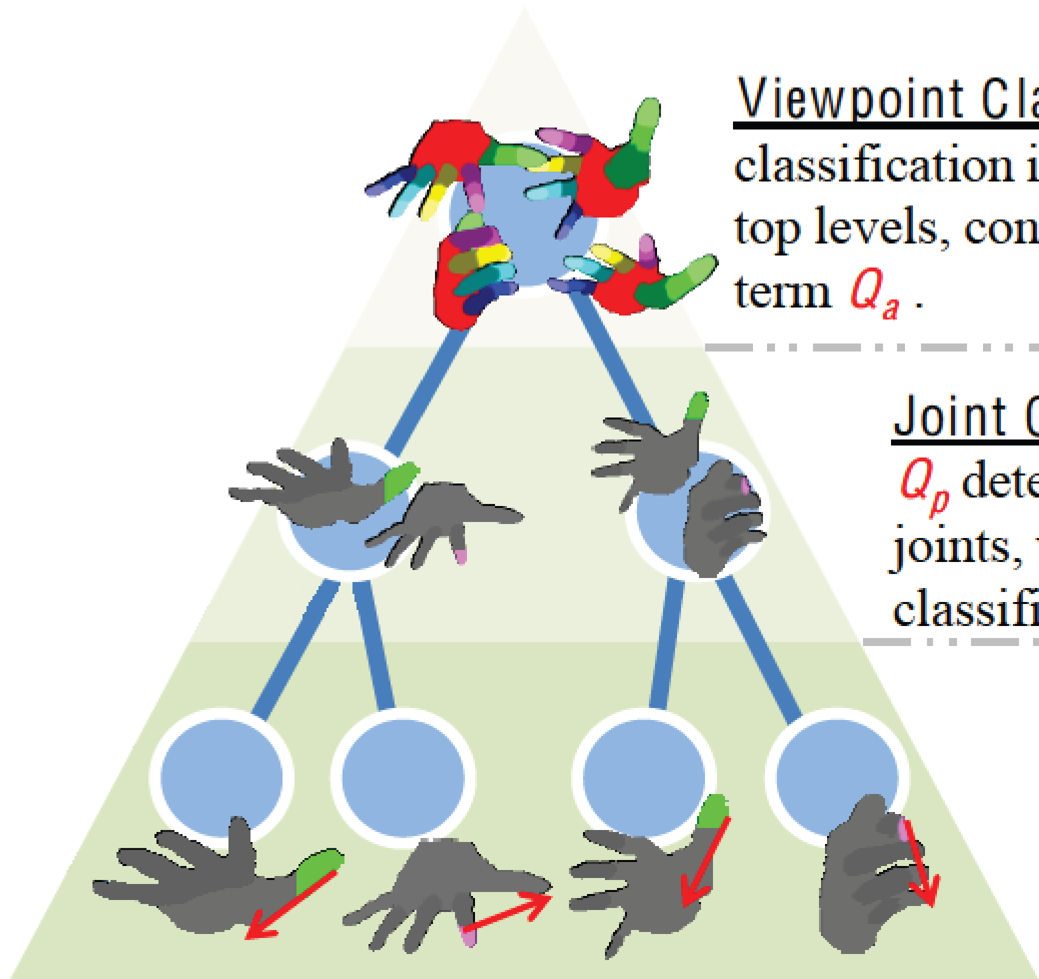
- Not feasible to capture all possible variations of hand with synthetic data.
- Methods using only synthetic data suffer from synthetic-realistic discrepancies.
- But: Using realistic training data expensive, due to manually labelling them.
- Solution: Transductive Learning.

Transductive Random Forest

- Transductive learning: learn from labelled data, apply knowledge transform to related unlabelled data
- Estimate pose based on knowledge gained from both labelled and unlabelled data.



Overview



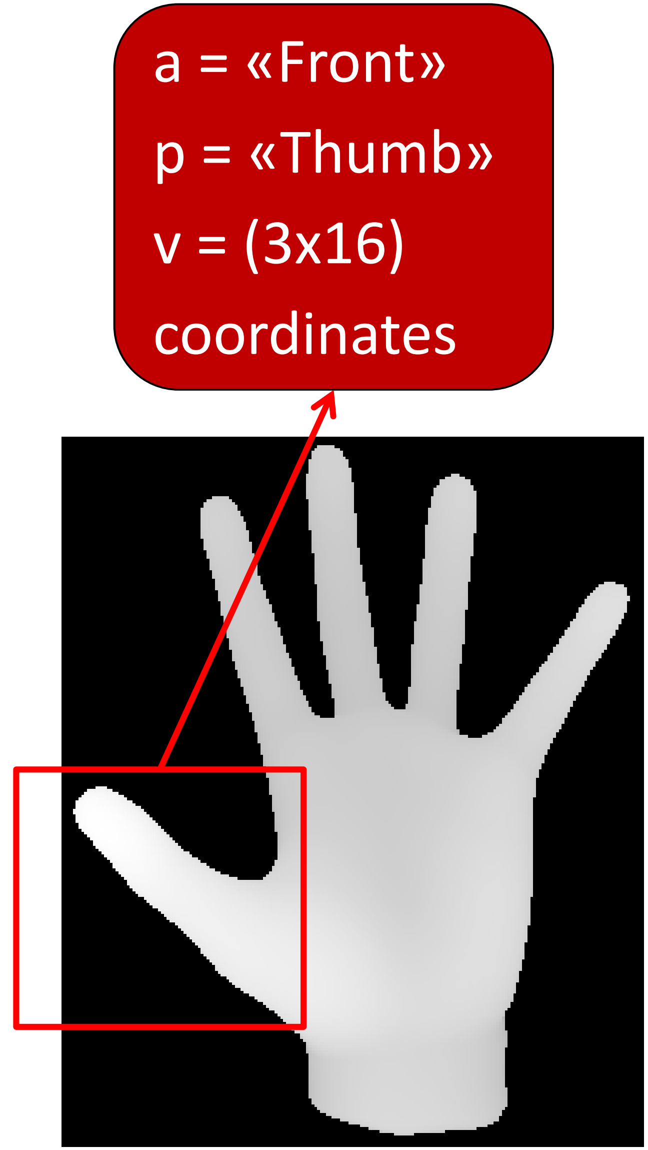
<u>Viewpoint Classification:</u> Viewpoint classification is first perfromed at he top levels, controlled by the viewpoint term Q_a .

<u>Joint Classification:</u> At mid levels, Q_p determines classification of joints, when most viewpoints are classified.

<u>Regression:</u> To describe the distribution of realistic data, nodes are optimised for data compactness via Q_v and Q_u towards the bottom levels.

Training data

- Training data consists of labelled real data and synthetic data, and unlabelled real data
- Labelled elements are image patches, not pixels
- Label consists of tuple (a,p,v):
  - a = Viewpoint
  - p = Label of the closest joint
  - v = Vector containing all positions of joint



Quality Function

- Randomly choose between the two:

$$\begin{cases} Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_p + (1-\alpha)(1-\beta) Q_v \\ Q_{tss} = Q_t^{\omega} Q_u \end{cases}$$

Transductive Term

Classification-Regression Term

Quality Function

$$Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_p + (1-\alpha)(1-\beta) Q_v$$

- Q_a : Measures quality of split with respect to viewpoint a
- Q_p : Measures quality of split with respect to joint label p
- Q_v : Measures compactness of vote vector v

Quality Function Parameter

Measures the “purity” of the node with respect to either the viewpoint a, or the joint label p

$$\alpha = \begin{cases} 1 \text{ if } \Delta_{\mathbf{a}}(\mathcal{L}) < t_\alpha \\ 0 \text{ otherwise} \end{cases} \qquad \beta = \begin{cases} 1 \text{ if } \Delta_{p}(\mathcal{L}) < t_\beta \\ 0 \text{ otherwise} \end{cases}$$

Quality Function

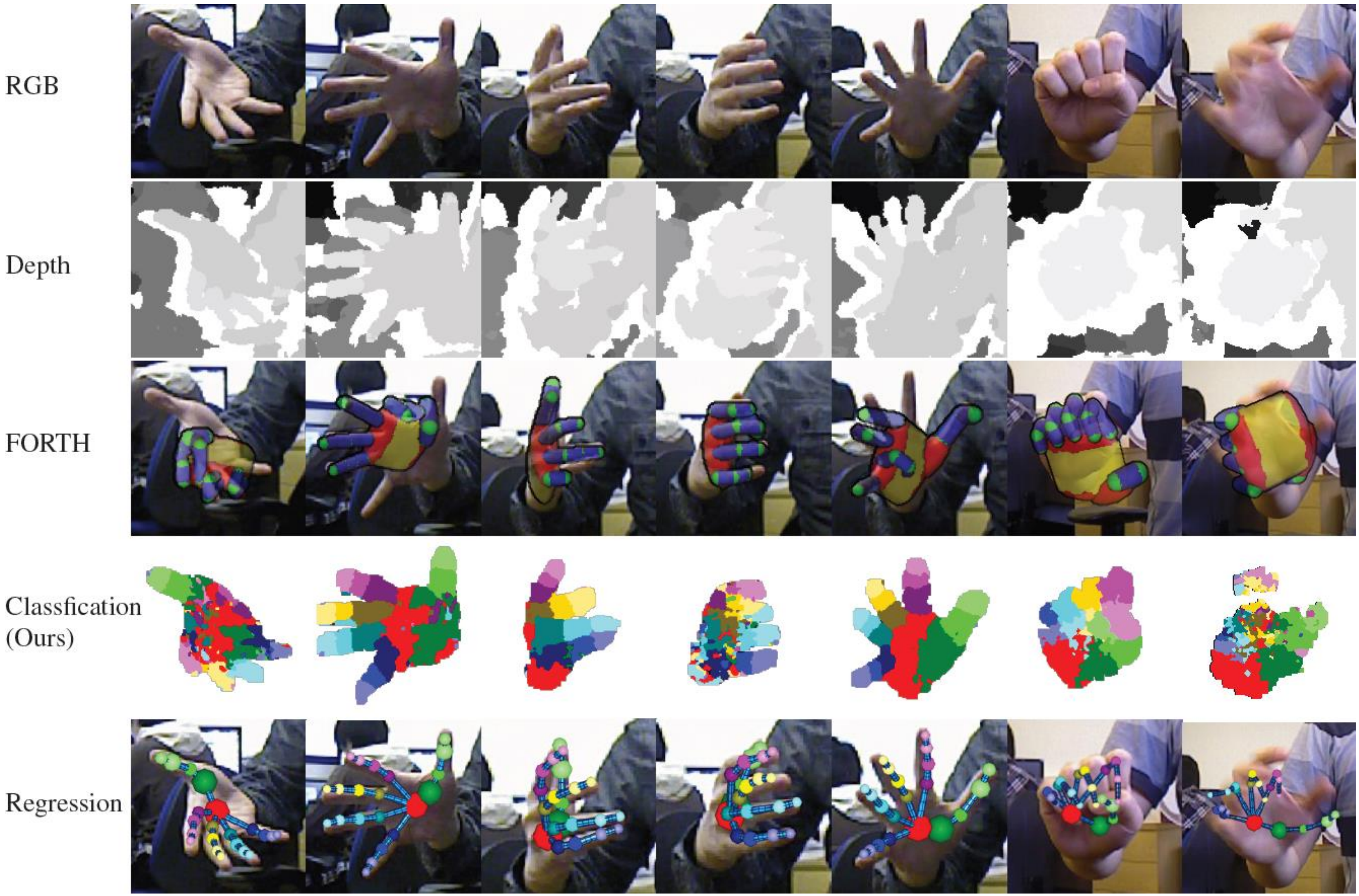
$$Q_{tss} = Q_t^{\omega} Q_u$$

- Q_t : Measures image similarity between real data patches
- Q_u : Measures purity based on the association between the labelled and unlabelled data

Kinematic Refinement

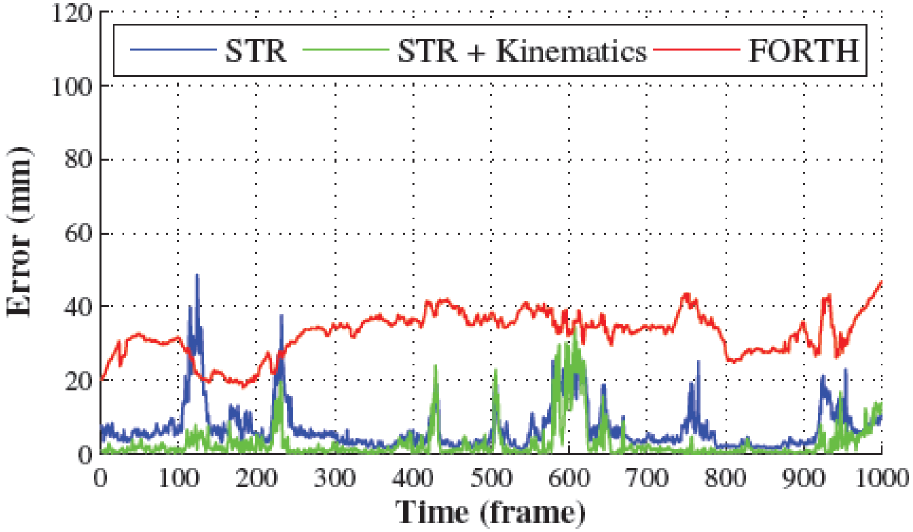
- Hands are biomechanically constrained on the poses it can do.
- Use this for our advantage.
- Utilize kinematic refinement to enforce these constraints.

Some results

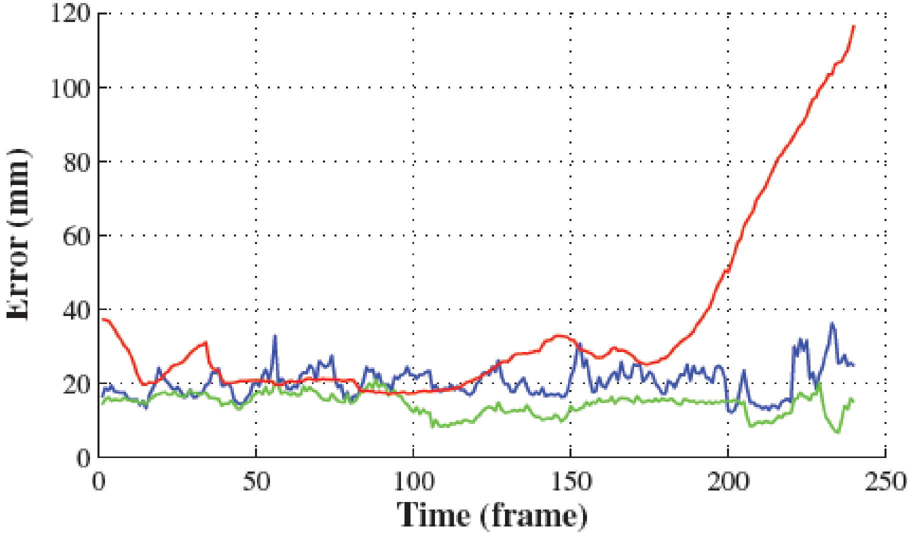


Joint prediction accuracy

Quantitative results of the multi-view experiment



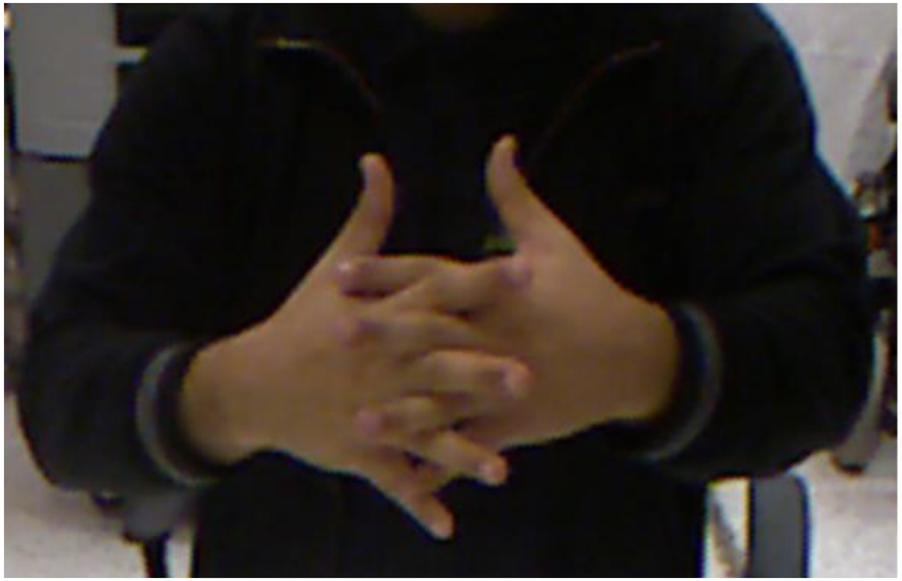
Test sequence B (Average error)



Test sequence C (Average error)

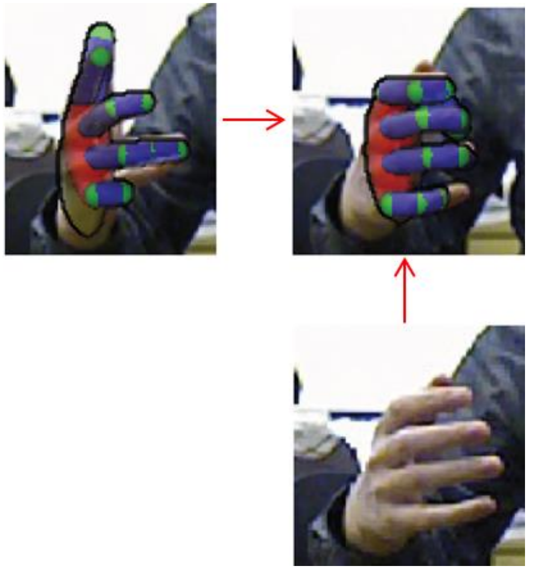
Estimating pose of two hands?

- Just apply single hand pose estimator twice?
- What if both hands are strongly interacting?
- Additional occlusion must be accounted for.

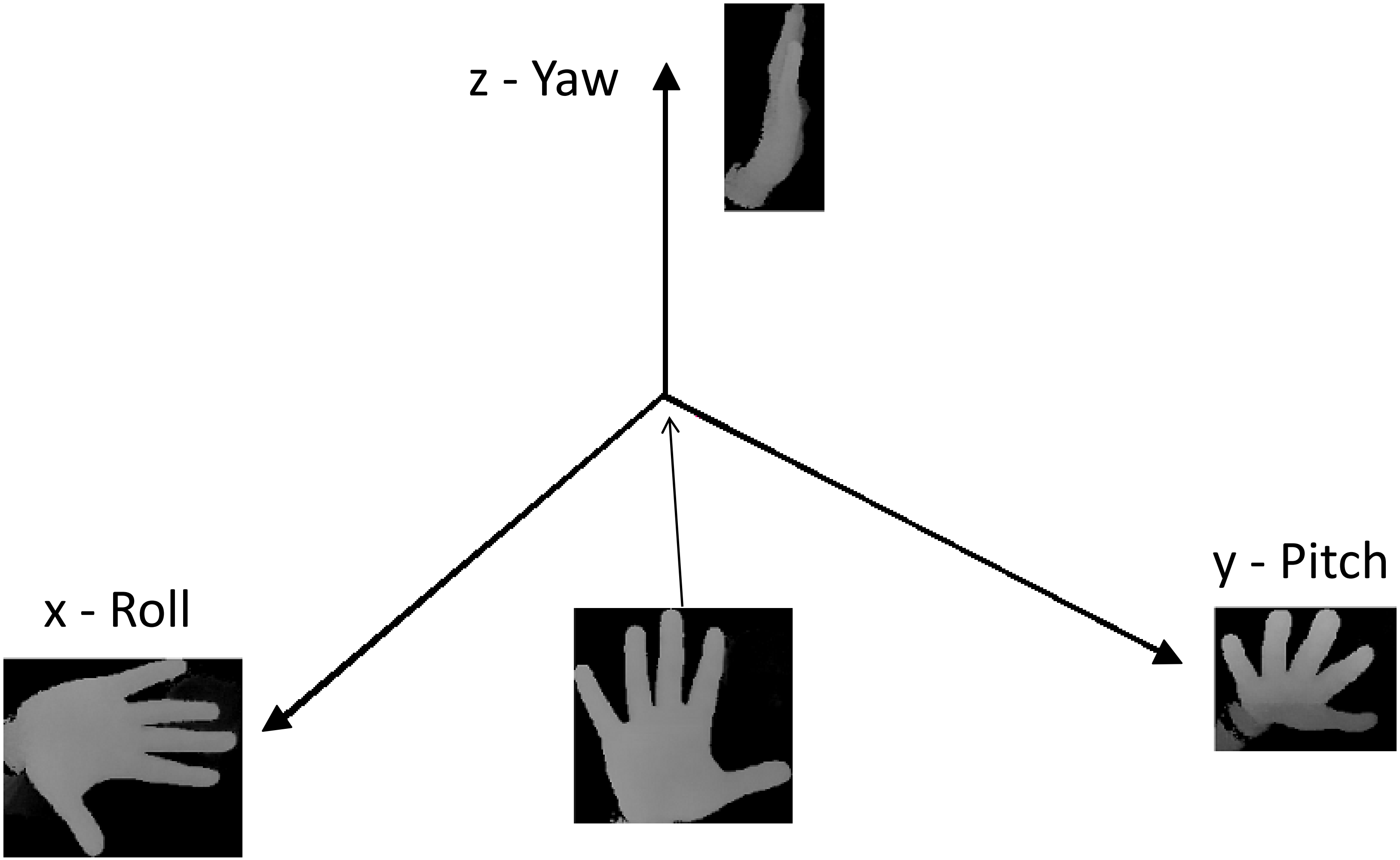



Dual hand pose estimation

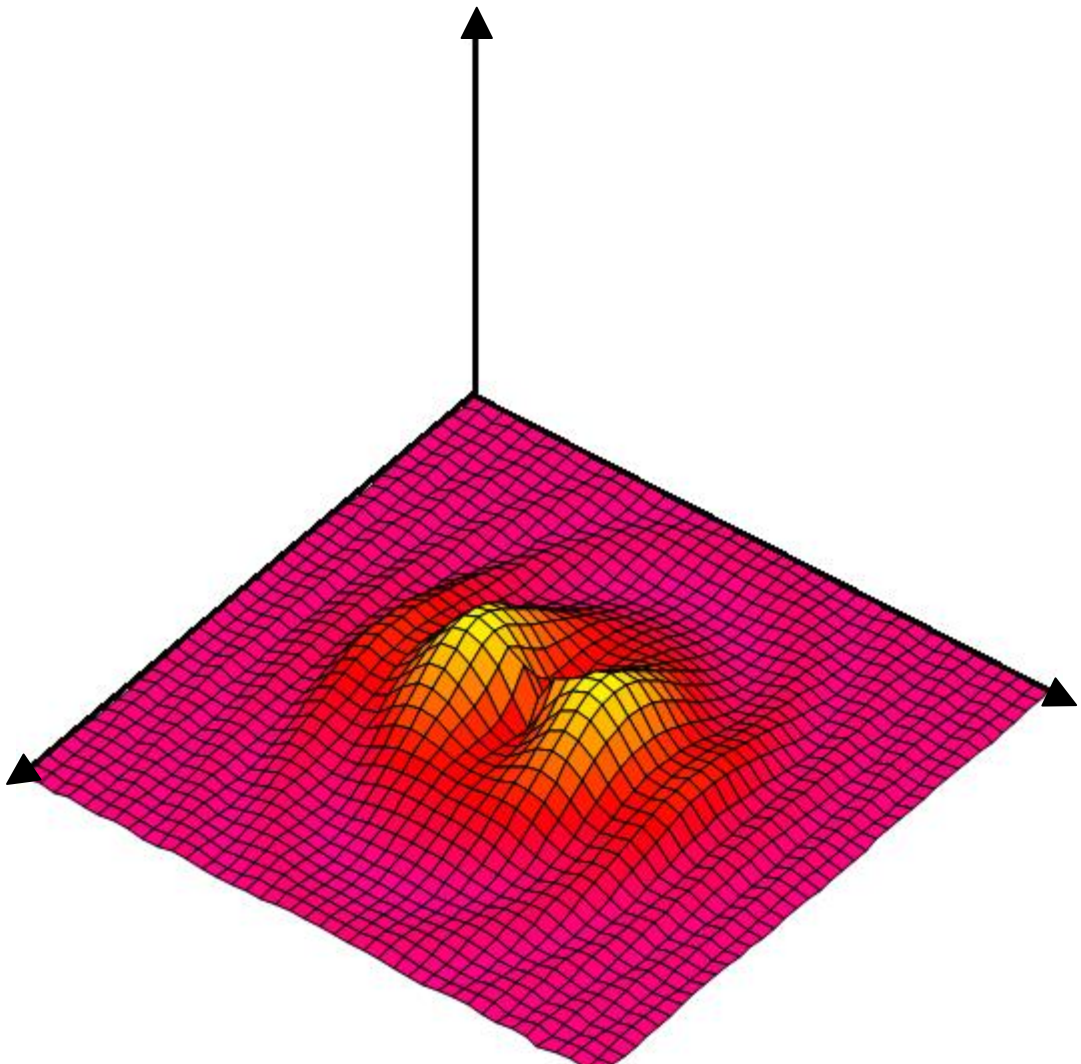
- Model-based approach.
- Set up parameter space representing all degrees of freedom for both hands.
- Employ PSO to find best parameters suiting observation and current configuration with respect to a cost function.



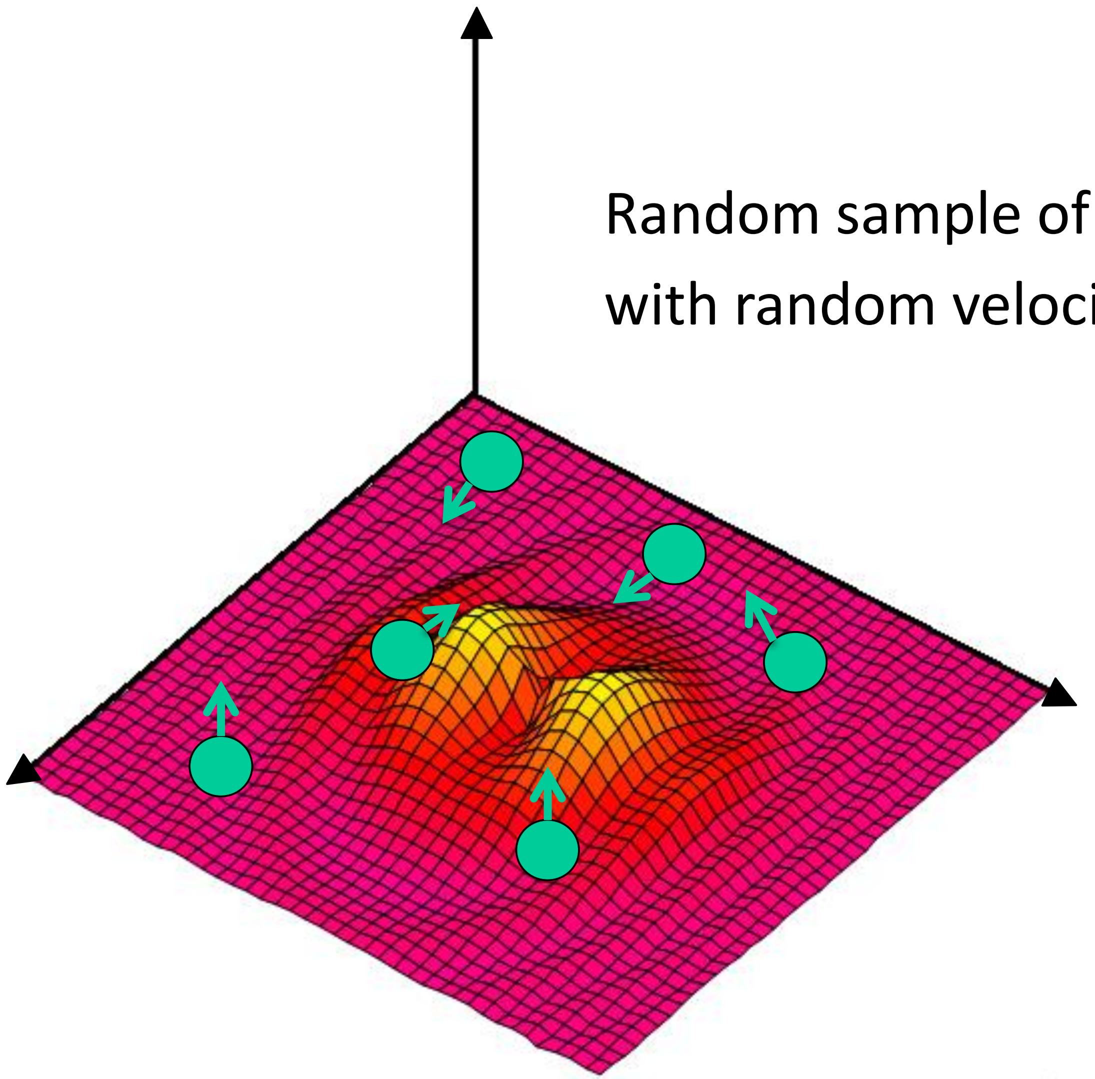
Sample parameter space



Cost function over param. space



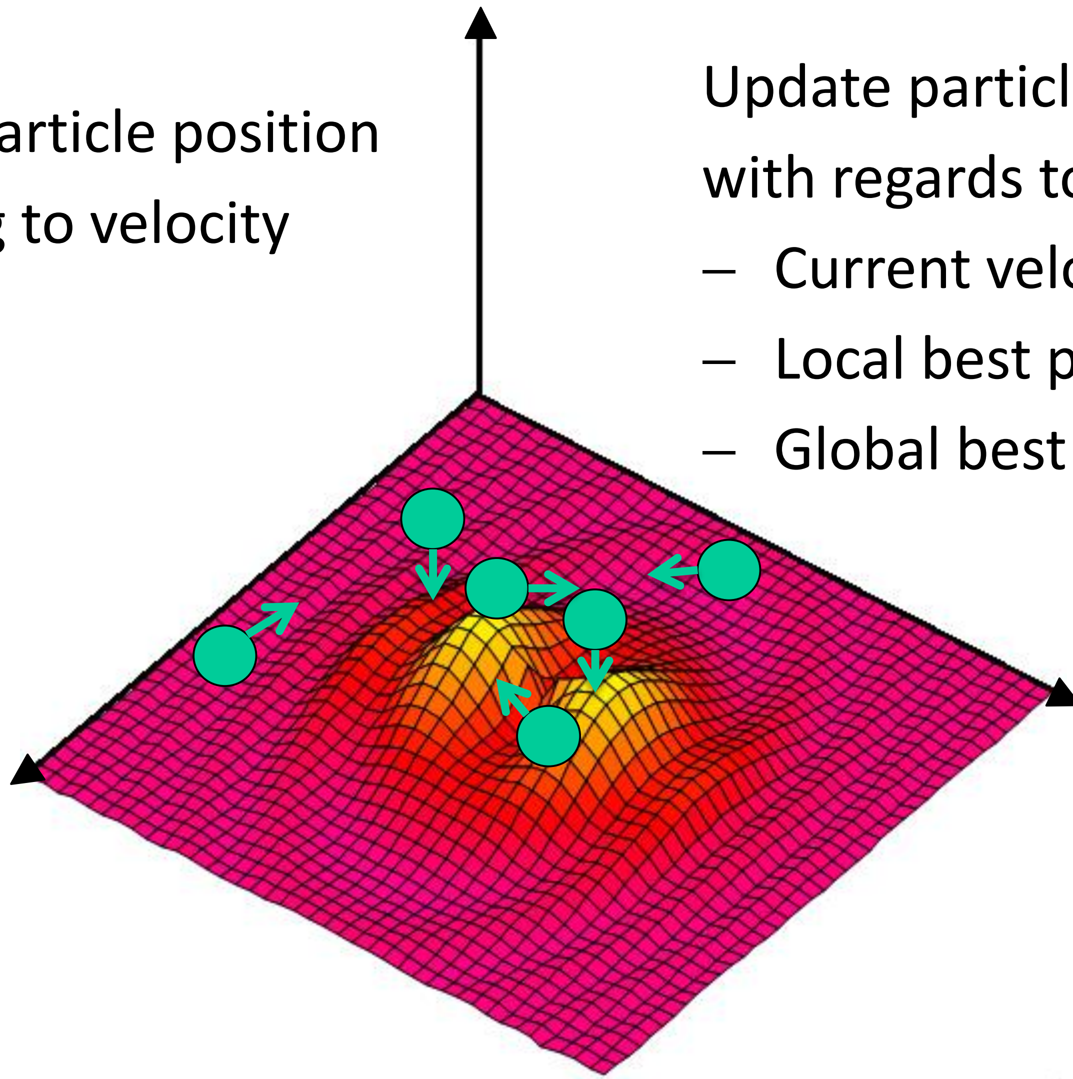
Initialization



Random sample of n particles
with random velocities.

Iterating over parameter space

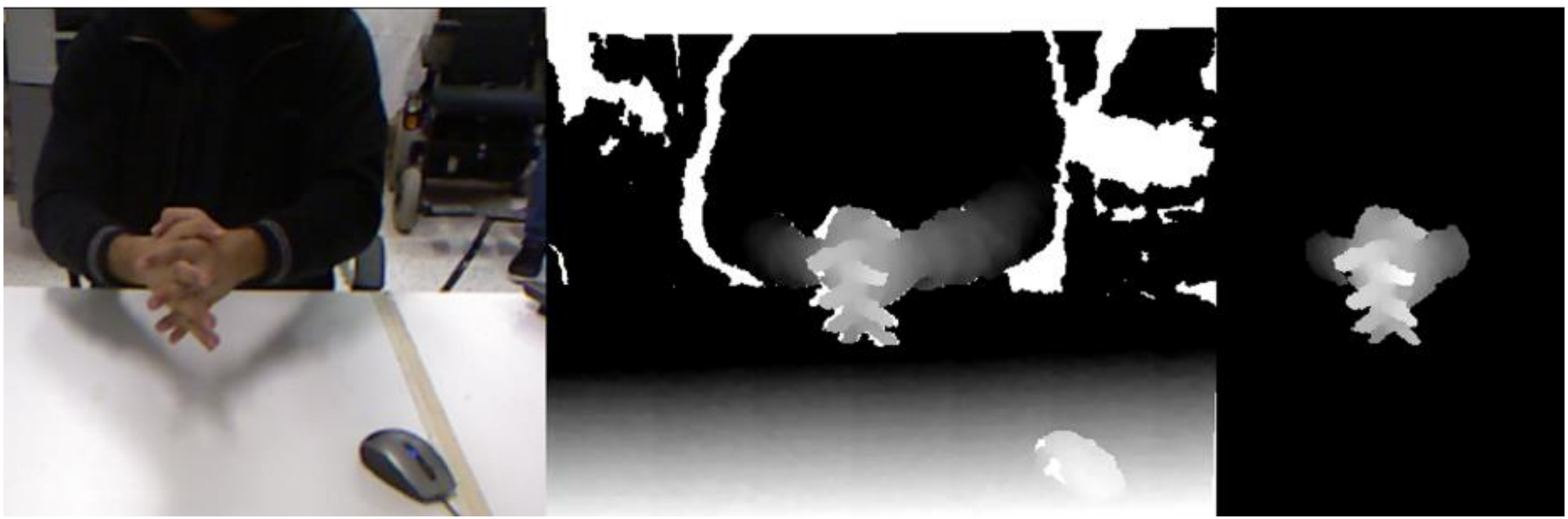
Update particle position according to velocity



Update particle velocities with regards to:
- Current velocity
- Local best position
- Global best position

Tracking

- Use RGB image to create skin map.
- Segment depth image according to skin map.



Tracking

- Cost function to optimize:

$$E(O,h,C) = P(h) + \lambda_k \cdot D(O,h,C)$$

P(h): Penalizes invalid finger positions.

D(O,h,C): Penalizes discrepancies between hypothesis h and observation O.

Applying PSO

- Change particle velocity according to:

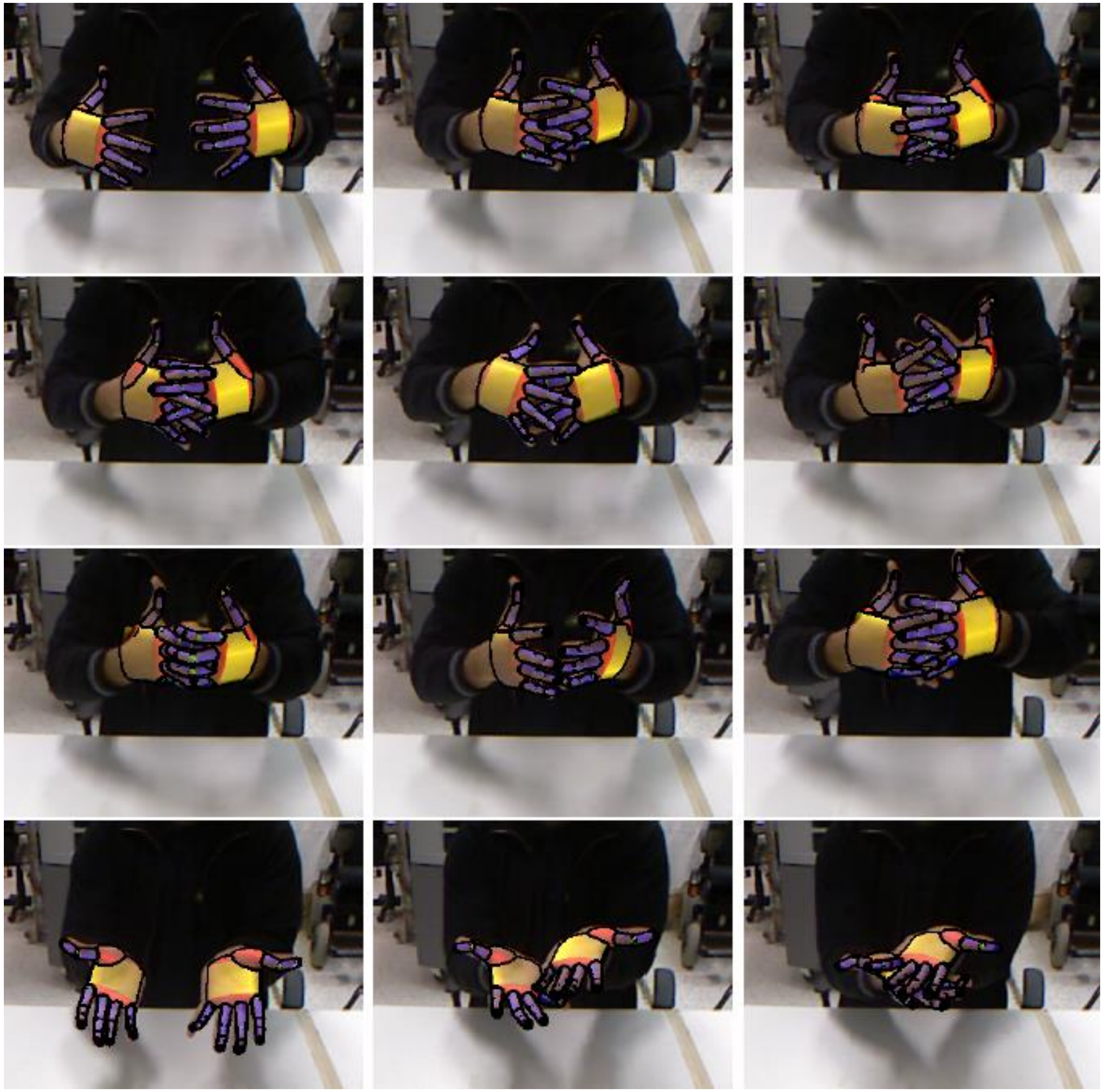
$$v_{k+1,i} = w(v_{k,i} + c_1 r_1 (P_{k,i} - x_{k,i}) + c_2 r_2 (G_k - x_{k,i}))$$

$P_{k,i}$ = Best known position of particle i in generation k.

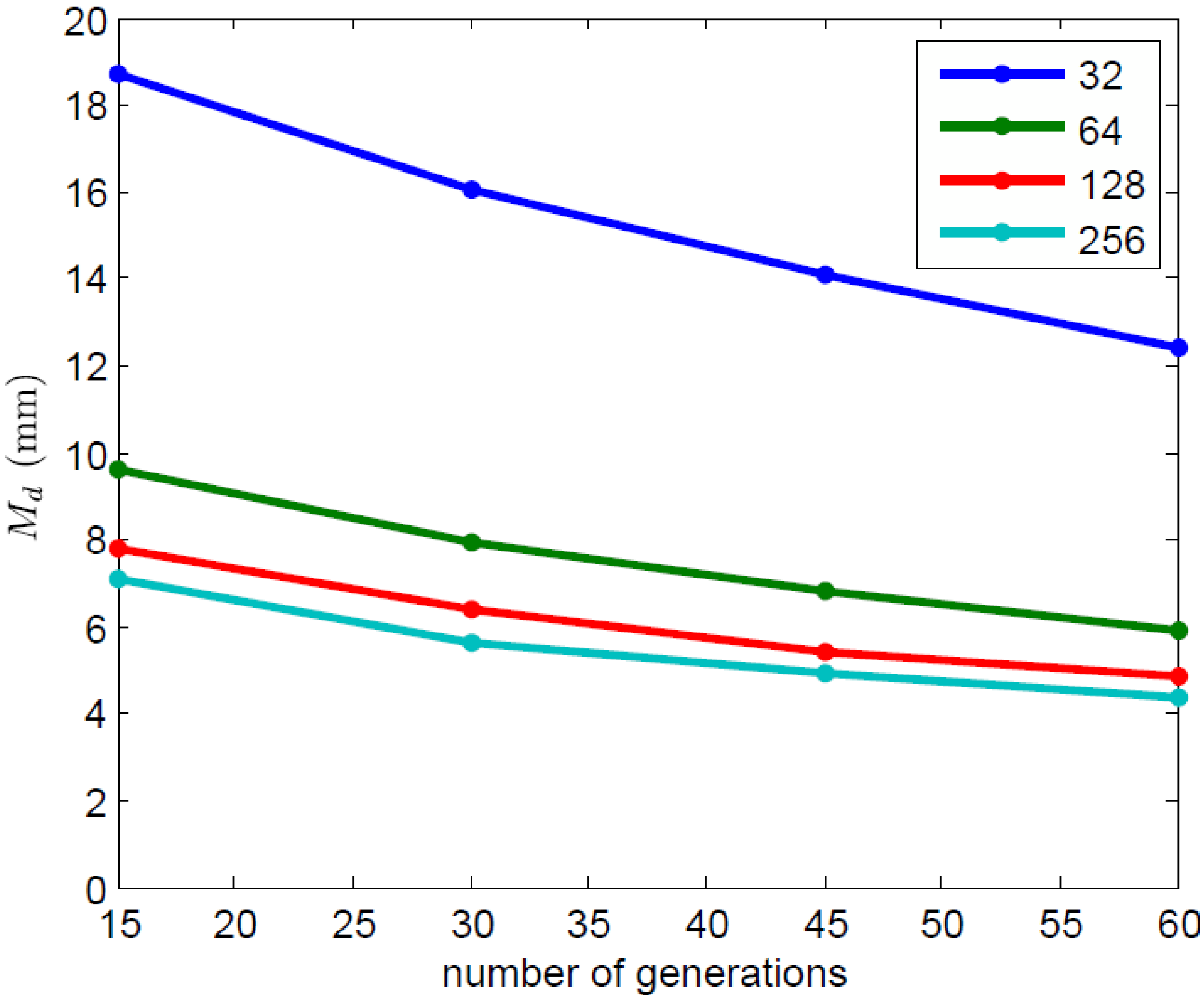
G_k = Best known position of all particles in generation k.

- Apply PSO for each observation O. Exploit temporal information by sampling particles around previous hypothesis.

Some results



Accuracy



32
64
128
256
M_d (mm)
number of generations
0
2
4
6
8
10
12
14
16
18
20
15
20
25
30
35
40
45
50
55
60

Future of Hand Pose estimation

- Academically solved
- Further research in areas of recovering more than pose, such as hand model or 3D skin models.
- Including RGB image for prediction increases accuracy.
- Use of real data reduces synthetic-realistic discrepancies.

Thank you for your attention!