Imperial College London

2015

Randomised Decision Forests and Tree-Structured Algorithms in Computer Vision

Tae-Kyun Kim
Imperial College London
http://www.iis.ee.ic.ac.uk/icvl/

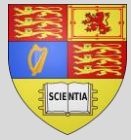
Imperial College London

Structure

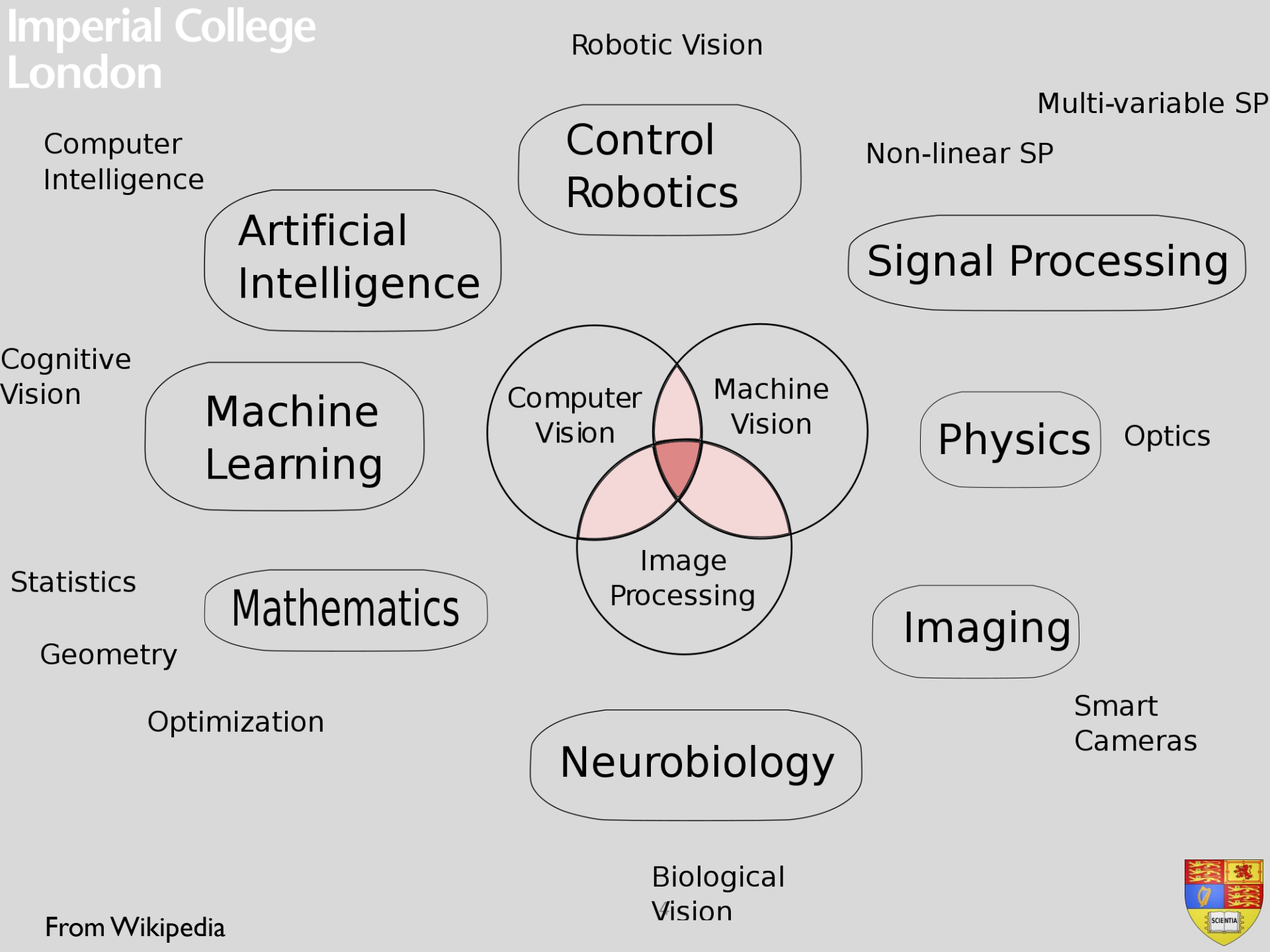
- Part I. **Boosting** (80 mins)
- Part II. **Decision Forests**
  - **Basics** (40 mins)
  - **Applications** (40 mins)
- Part III. **Hands-On Decision Forests** (90 mins)
- Part IV. **Hands-On Boosting** (90 mins)

Part I. Boosting

Motivations, Theories, Extensions



Imperial College London



From Wikipedia

Image Recognition Pipeline

- A typical structure of image recognition pipeline is shown below:

  Image ***I*** ➔

  Image *Representation* ***x*** (input) ➔

  Machine *Learning* ➔

  Semantic Labels ***t*** (output)

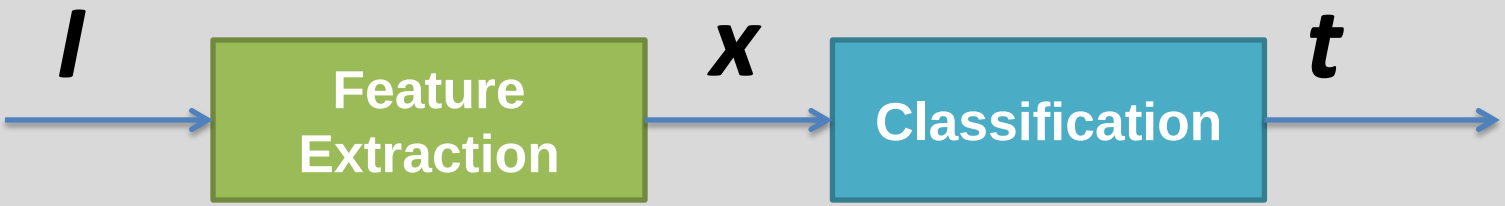
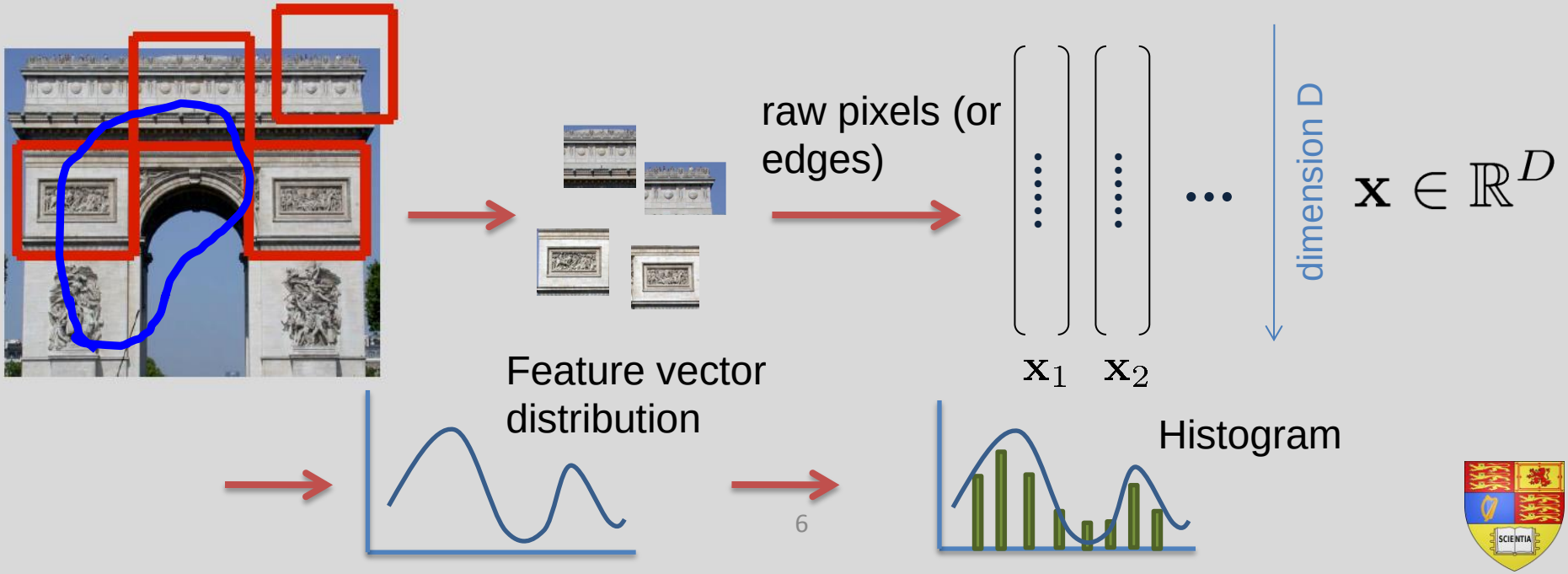


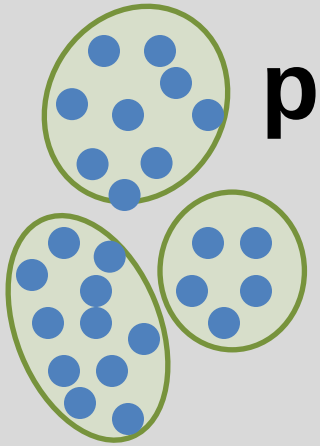
Image representation

- A whole image or an image region, is represented by a set of image patches around feature points (e.g. corners).
- Image patches are represented by feature vectors formed by concatenating raw pixel values (or edges, SIFT).
- An image is then given as distribution of features (BoW).



Machine Learning

- Unsupervised Learning



Density estimation

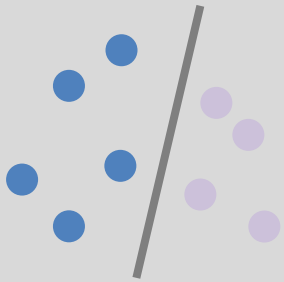
- Clustering
- GMM (EM)
- PCA, manifold

- Supervised Learning

(x,t) **f:x → t**

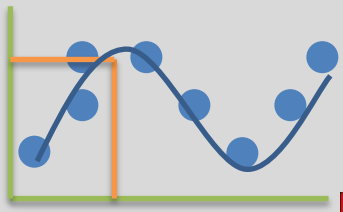
Classification, t={1,…,n}

- Boosting
- SVM



Regression, t: conti. variable

- Polynomial curve fit
- Gaussian Process



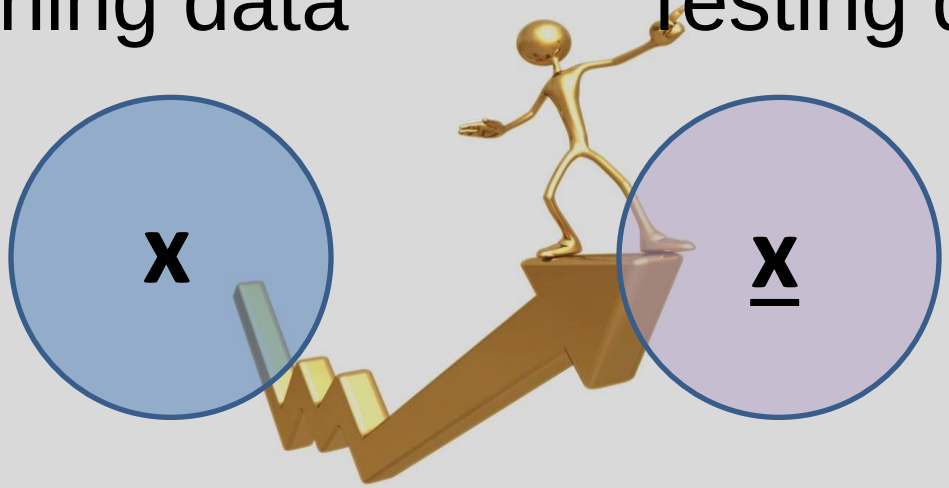
Generalisation to unseen data

Image space is huge, practically assumption is that we see a small fraction of instances in training.

A fundamental issue, what we learnt is well generalised to unseen data?

Training data

Testing data



p(x)

p(x)

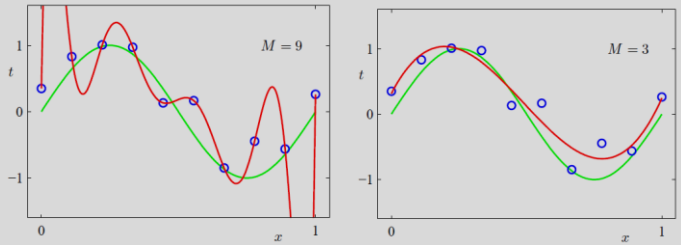
f:x → t

f:x → t

How to ensure *generalisation*

Polynomial curve fit: a regularization term

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



SVM : maximizing the margin

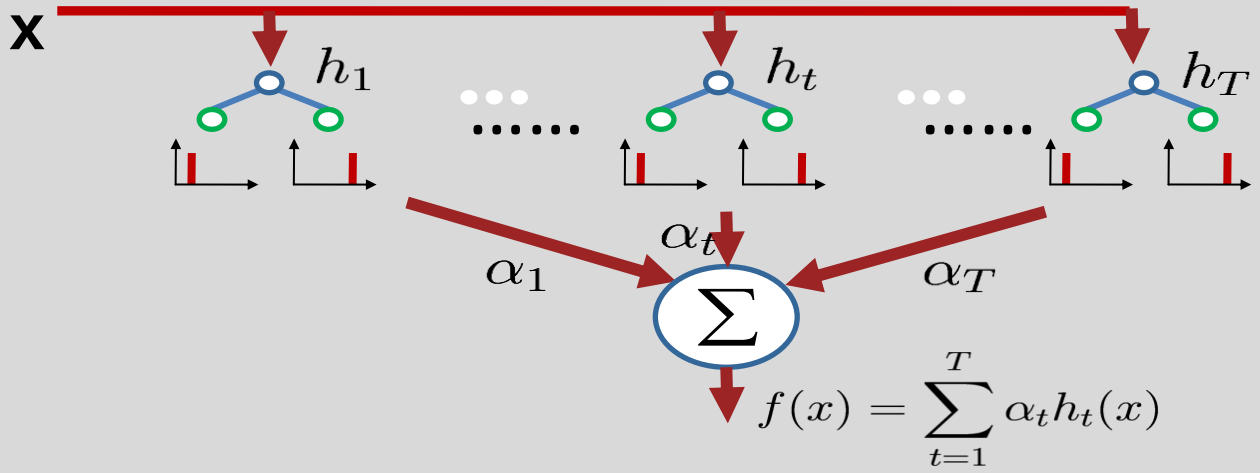
$$\arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$ subject to …, $$y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$$

GP : prior distribution

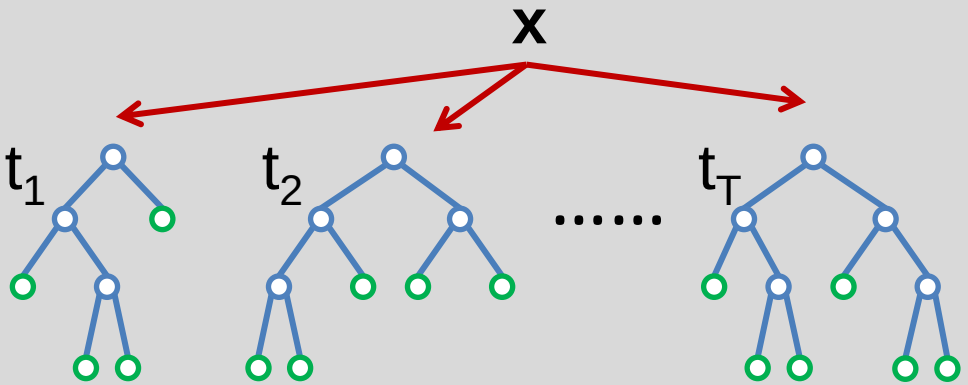
$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$ $$y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$$

How to ensure *generalisation*

Boosting: a flat structure by a linear weighted sum



Decision Forests: randomisation on data/features



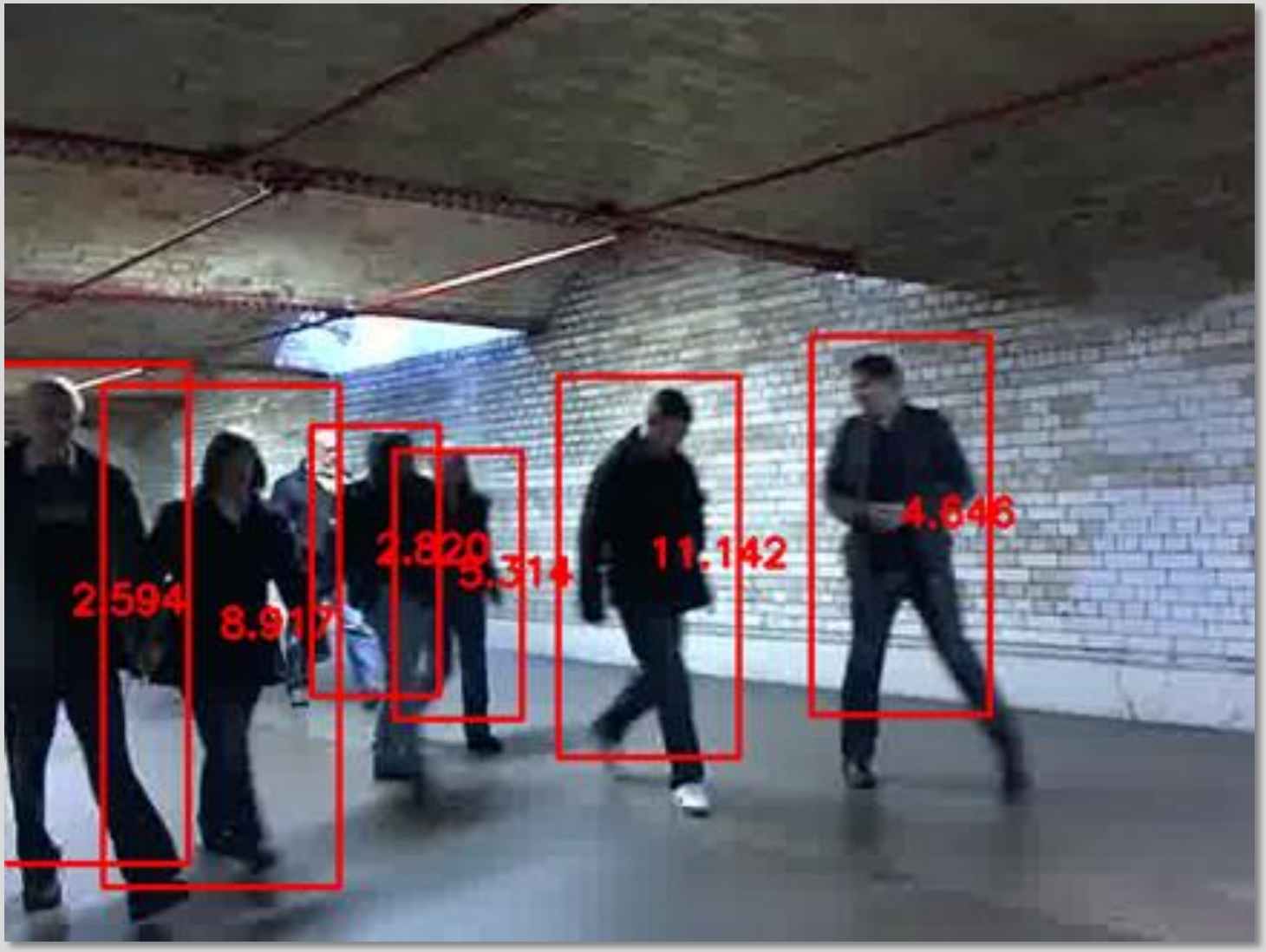
Extremely fast classification by tree structure algorithms

Imperial College London

Object Detection

Imperial College London

Object Detection



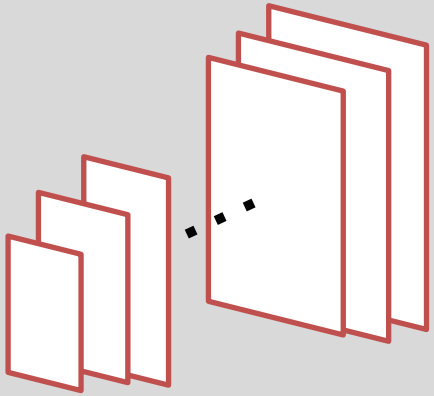
Imperial College London

Object Detection

Number of windows

# of pixels

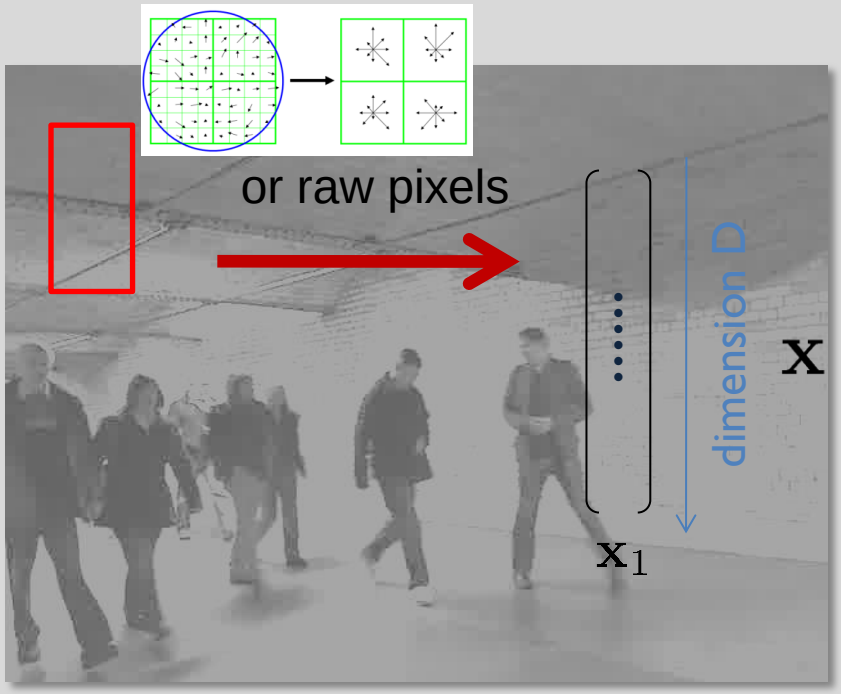
X



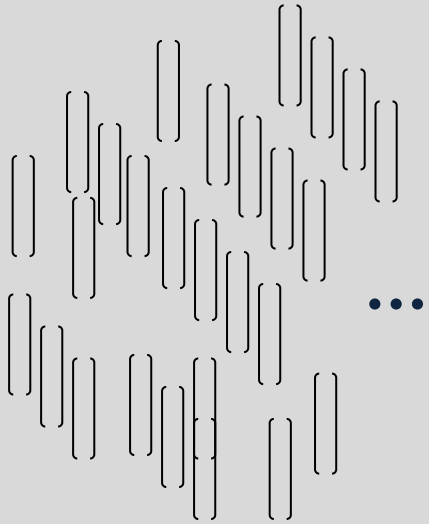
# of scales

→ **Number of Windows:** 747,666

Time per window



$\mathbf{x} \in \mathbb{R}^D$

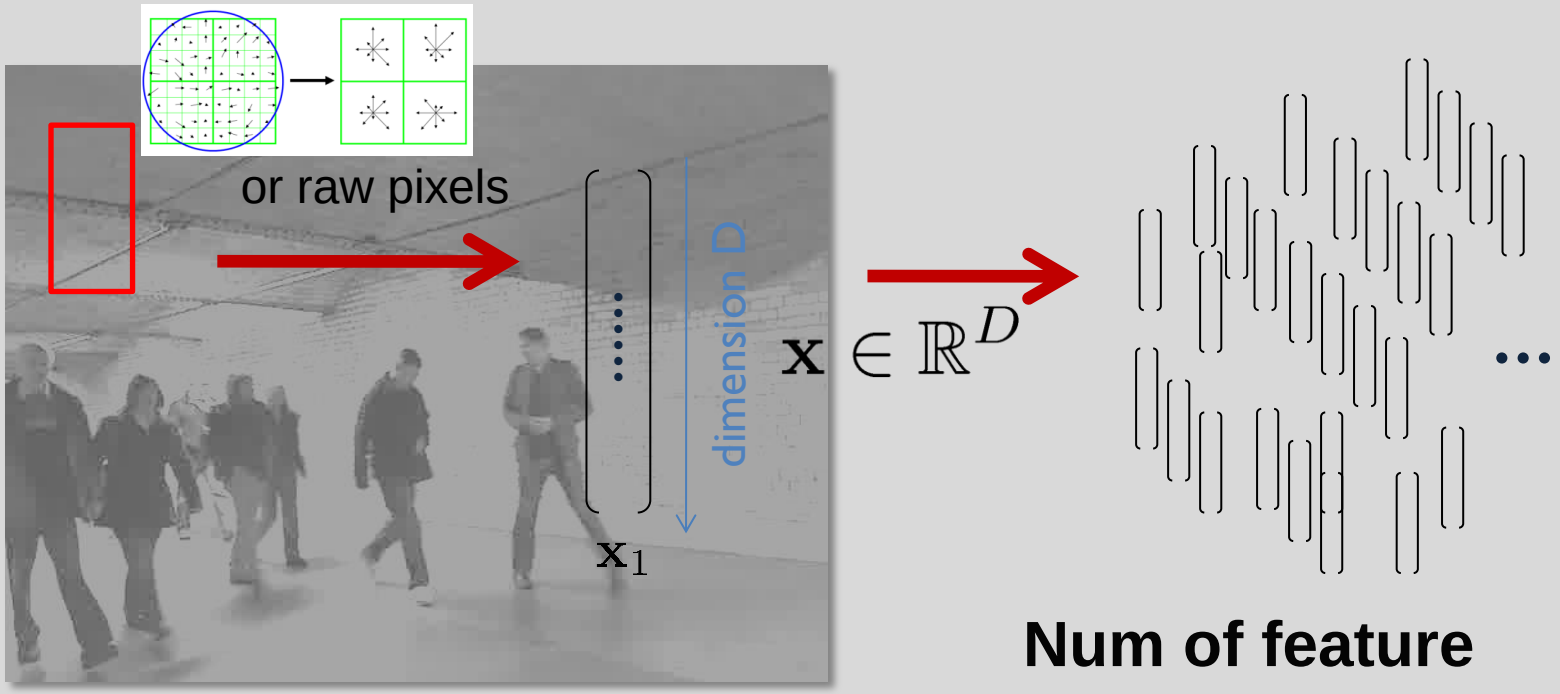


Num of feature vectors: 747,666

$$f : x \longrightarrow t$$

Classification: $t \in \{1, 2, ..., n\}$

Time per window



Num of feature vectors: 747,666

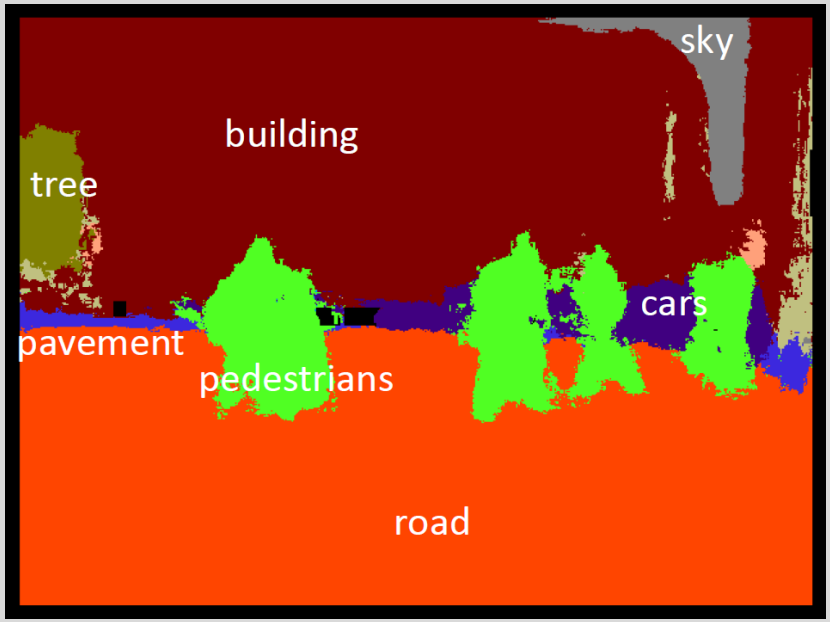
In order to finish the task in say 1 sec

→ **Time per window (or vector):**
0.00000134 sec

Semantic Segmentation

- Requiring pixel-wise ***multi-class*** classification

Input



Output

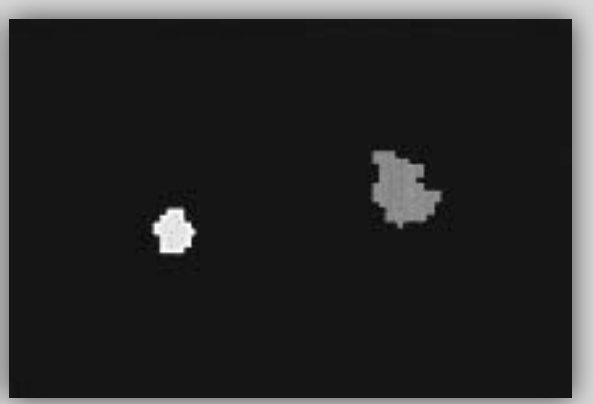
Imperial College London

Boosting as a Tree-structured Classifier

More traditionally… Narrow down the search space

- Integrating Visual Cues [Darrell et al IJCV 00]





  - Face pattern detection output (left).
  - Connected components recovered from **stereo** range data (mid).
  - Flesh hue regions from **skin hue** classification (right).

Since about 2001…

Boosting Simple Features [Viola &Jones 01]

- Adaboost classification

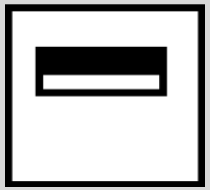
Strong classifier

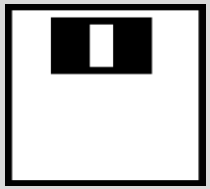
Weak classifier

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

- Weak classifiers: Haar-basis like functions (45,396 in total)

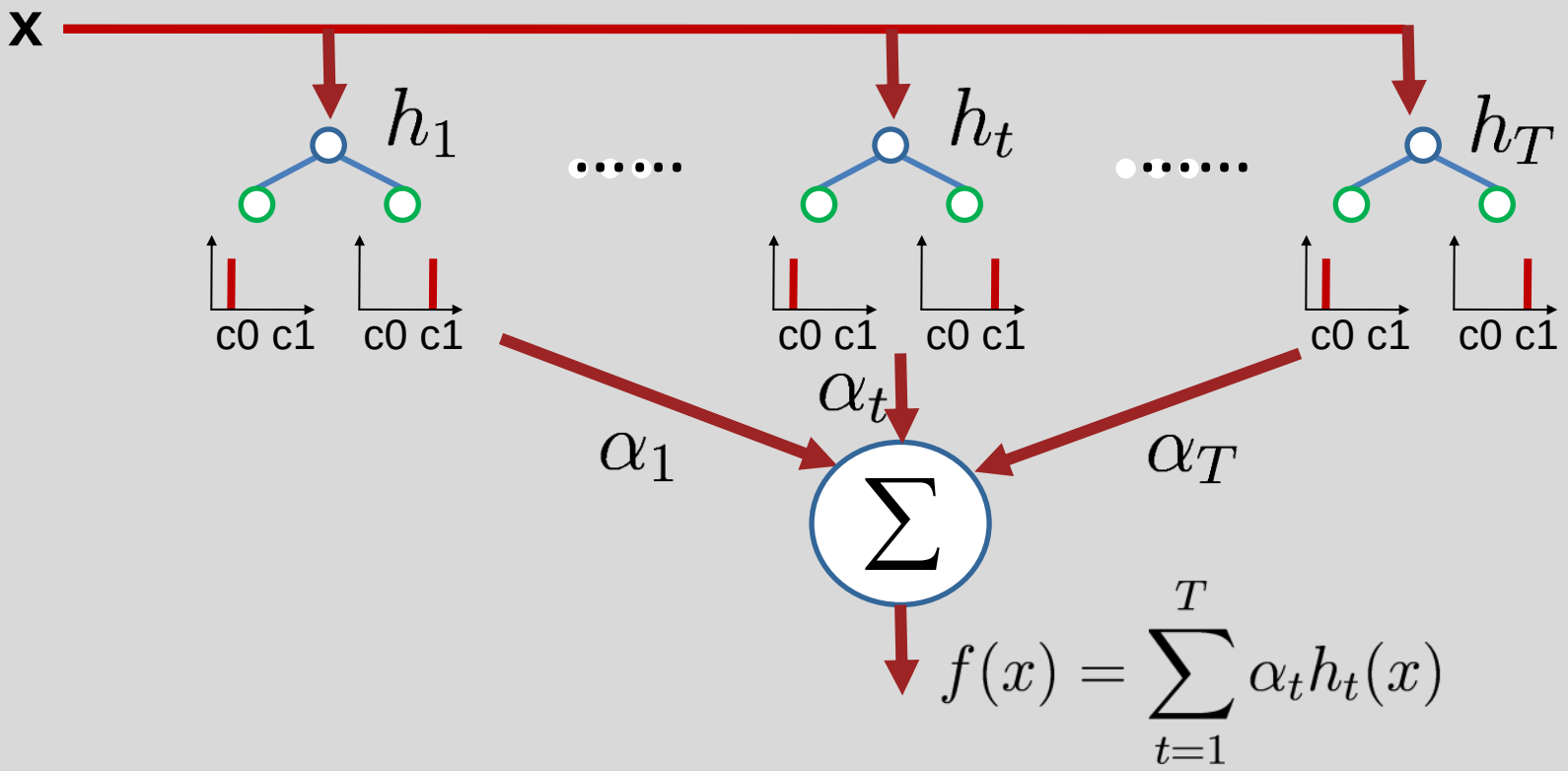
$$h_t(x) = \begin{bmatrix} +1 & if\ f_t(x) \geq \theta \\ -1 & \text{otherwise} \end{bmatrix}$$





Boosting (very shallow network)

- The strong classifier H as boosted decision stumps has a flat structure



  - Cf. Decision "ferns" has been shown to outperform "trees" [Zisserman et al, 07] [Fua et al, 07]

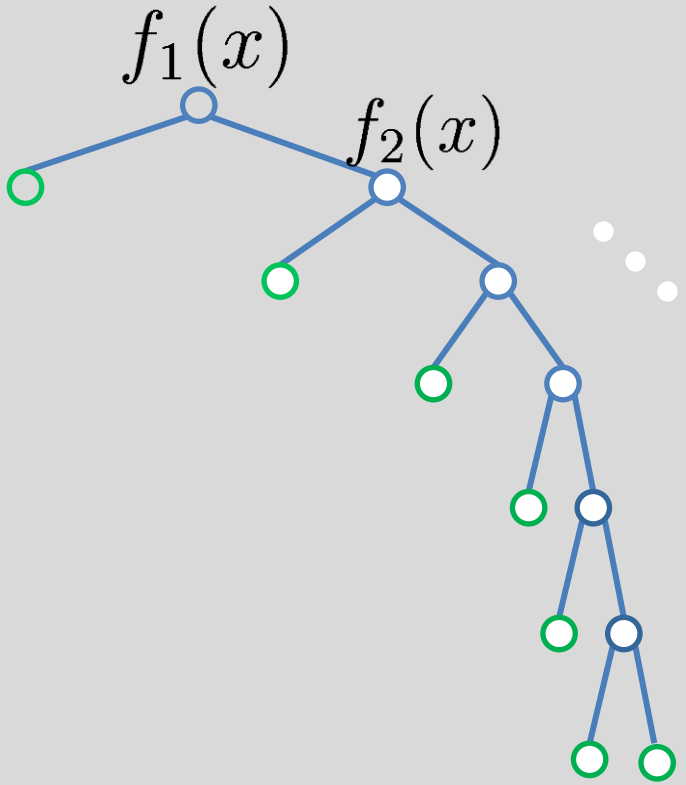
Boosting -continued

- Good generalisation by a flat structure
- Fast evalution
- Sequential optimisation

Boosting Cascade [viola & Jones 04], Boosting chain [Xiao et al]

- Very unbalanced tree
- Speeds up for unbalanced binary problems
- Hard to design

A strong boosting classifier

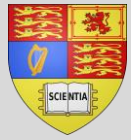


Imperial College London

Introduction to Boosting

Classifiers

- AdaBoost (Adaptive Boosting)



Boosting

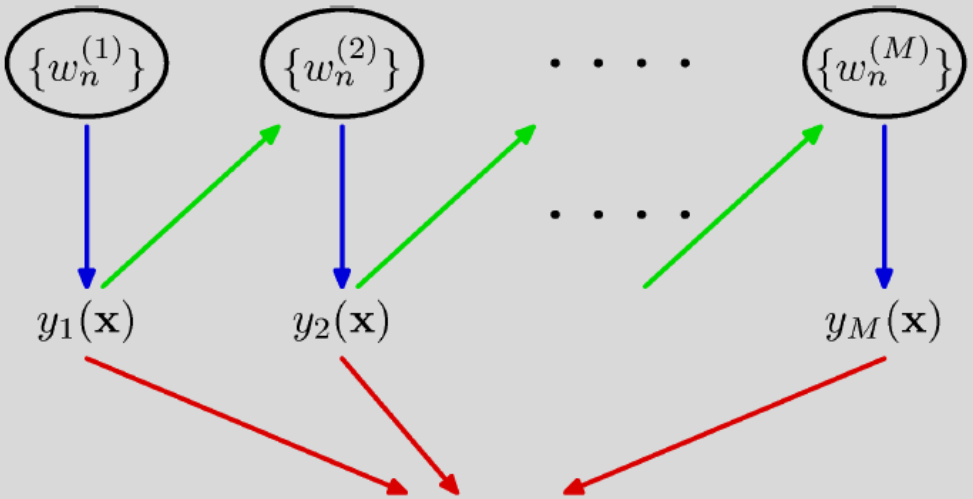
- Boosting gives good results even if the base classifiers have a performance slightly better than *random* guessing.

- Hence, the base classifiers are called *weakclassifiers* or *weaklearners*.

Boosting

For a two (binary)-class classification problem, we train with

training data $x_{1,\ldots,} x_N$
target variables $t_{1,\ldots,} t_N$, where $t_N \in \{-1,1\}$,
data weight $w_{1,\ldots,} w_N$
weak (base) classifier candidates $y(\mathbf{x}) \in \{-1, 1\}$.



$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

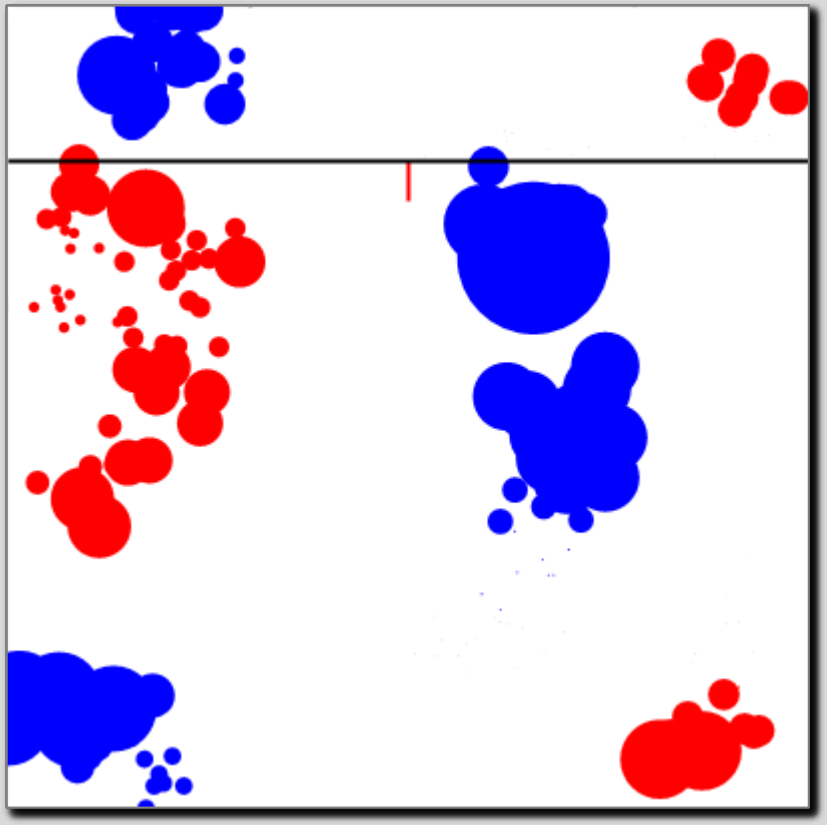
Boosting does

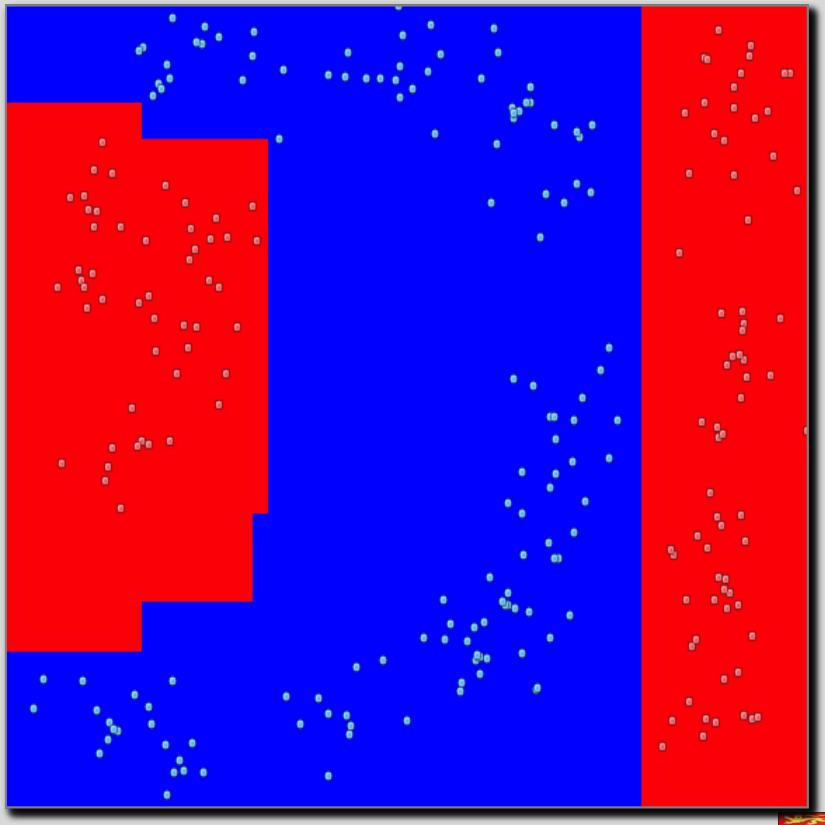
Iteratively,

1) reweighting training samples,

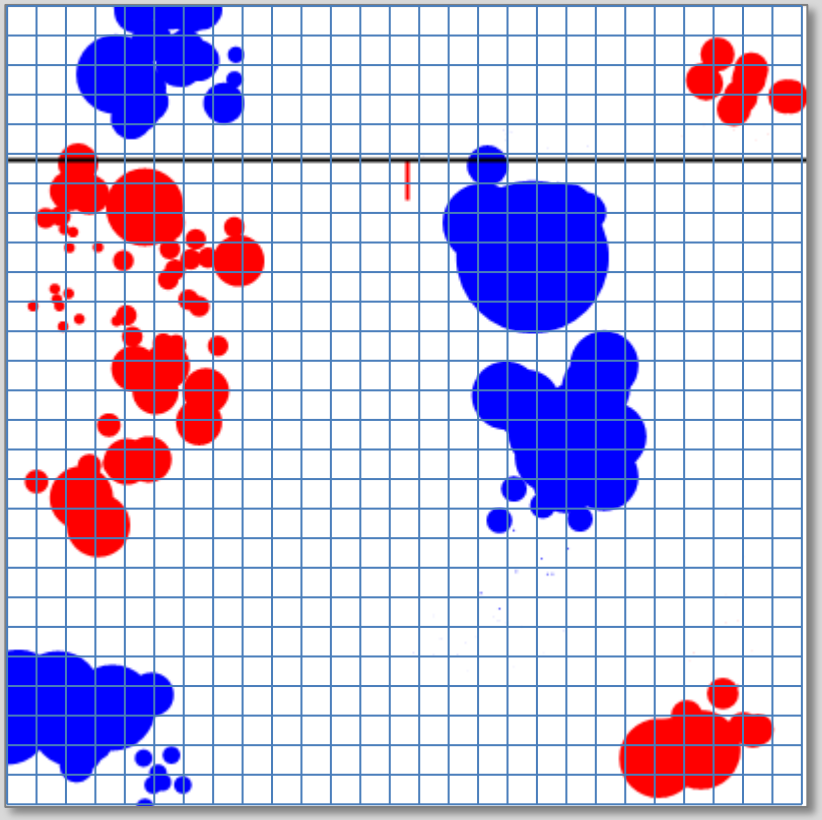
- by assigning higher weights to previously misclassified samples,

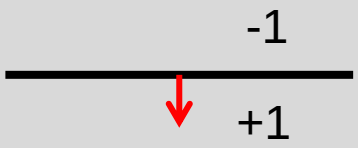
2) finding the best weakclassifier for the weighted samples.





In the previous example, the weaklearner was defined by a horizontal or vertical line, and its direction.





AdaBoost (adaptive boosting)

1. Initialise the data weights $\{w_n\}$ by $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$.

2. For $m = 1, \ldots, M$: the number of weak classifiers to choose

(a) Learn a classifier $y_m(x)$ that minimises the weighted error, among all weak classifier candidates

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n) \qquad \text{Eqn 1}$$

where I is the impulse function.

(b) Evaluate

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \qquad \text{Eqn 2}$$

and set

$$\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\} \qquad \text{Eqn 3}$$

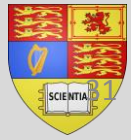
(c) Update the data weights

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}) \neq t_n)\} \qquad \text{Eqn 4}$$

3. Make predictions using the final model by

$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right) \qquad \text{Eqn 5}$$

Boosting as an optimisation framework



Minimising Exponential Error

- AdaBoost is the sequential minimisation of the exponential error function

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x}_n)\}$$

where $t_n \in \{-1, 1\}$ and $f_m(\mathbf{x})$ is a classifier as a linear combination of base classifiers $y_l(\mathbf{x})$

$$f_m(\mathbf{x}) = \frac{1}{2} \sum_{l=1}^{m} \alpha_l y_l(\mathbf{x})$$

- We minimise E with respect to the weight α_l and the parameters of the base classifiers $y_l(\mathbf{x})$.

- ***Sequential Minimisation:*** suppose that the base classifiers $y_1(\mathbf{x})$,. …, $y_{m-1}(\mathbf{x})$ and their coefficients $\alpha_1, \ldots, \alpha_{m-1}$ are fixed, and we minimise only w.r.t. α_m and $y_m(\mathbf{x})$.

- The error function is rewritten by

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x}_n)\}$$

$$= \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

$$= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

where $w_n^{(m)} = \exp\{-t_n f_{m-1}(\mathbf{x}_n)\}$ are constants.

- Denote the set of data points correctly classified by $y_m(\mathbf{x}_n)$ *by* T_m, and those misclassified M_m , then

$$E = e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{M}_m} w_n^{(m)}$$

$$= \left(e^{\alpha_m/2} - e^{-\alpha_m/2}\right) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

- When we minimise w.r.t. $y_m(\mathbf{x}_n)$, the second term is constant and minimising E is equivalent to

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n) \qquad \text{Eqn 1}$$

- By setting the derivative w.r.t. α_m to 0, we obtain $\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$ Eqn 3

where $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$. Eqn 2

- From $E = \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$

$$\rightarrow \quad w_n^{(m+1)} = w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

As $t_n y_m(\mathbf{x}_n) = 1 - 2I(y_m(\mathbf{x}) \neq t_n)$,

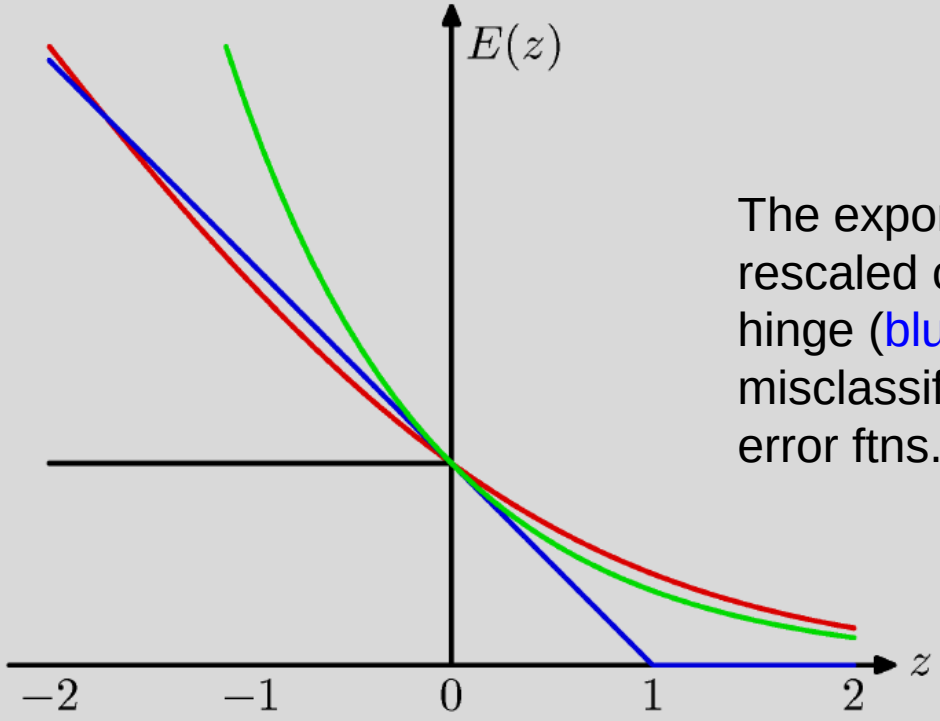
$$w_n^{(m+1)} = w_n^{(m)} \exp(-\alpha_m/2) \exp\{\alpha_m I(y_m(\mathbf{x}) \neq t_n)\}$$

The term $exp(-\alpha_m/2)$ is independent of n, thus we obtain

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}) \neq t_n)\}$$ Eqn 4

Exponential Error Function

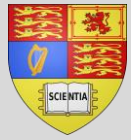
- *Pros:* it leads to simple derivations of Adaboost algorithms.
- *Cons:* it penalises large negative values. It is prone to outliers.



The exponential (green) rescaled cross-entropy (red) hinge (blue), and misclassification (black) error ftns.

Imperial College
London

Robust real-time object detector
Viola and Jones, CVPR 01



Imperial College London

Boosting Simple Features

[Viola and Jones CVPR 01]

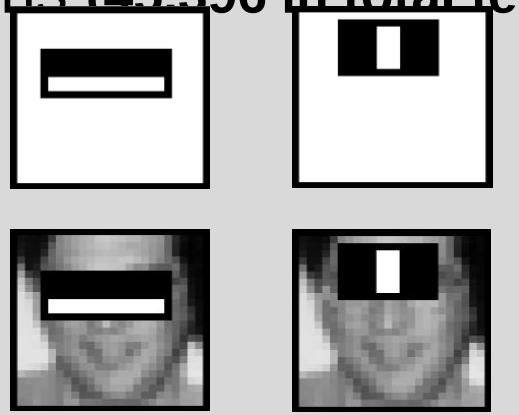
- Adaboost classification

Strong classifier **Weak classifier**

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

- Weak classifiers: Haar-basis like functions **(45,396 in total feature pool)**

$$h_t(x) = \begin{bmatrix} +1 & if\ f_t(x) \geq \theta \\ -1 & \text{otherwise} \end{bmatrix}$$

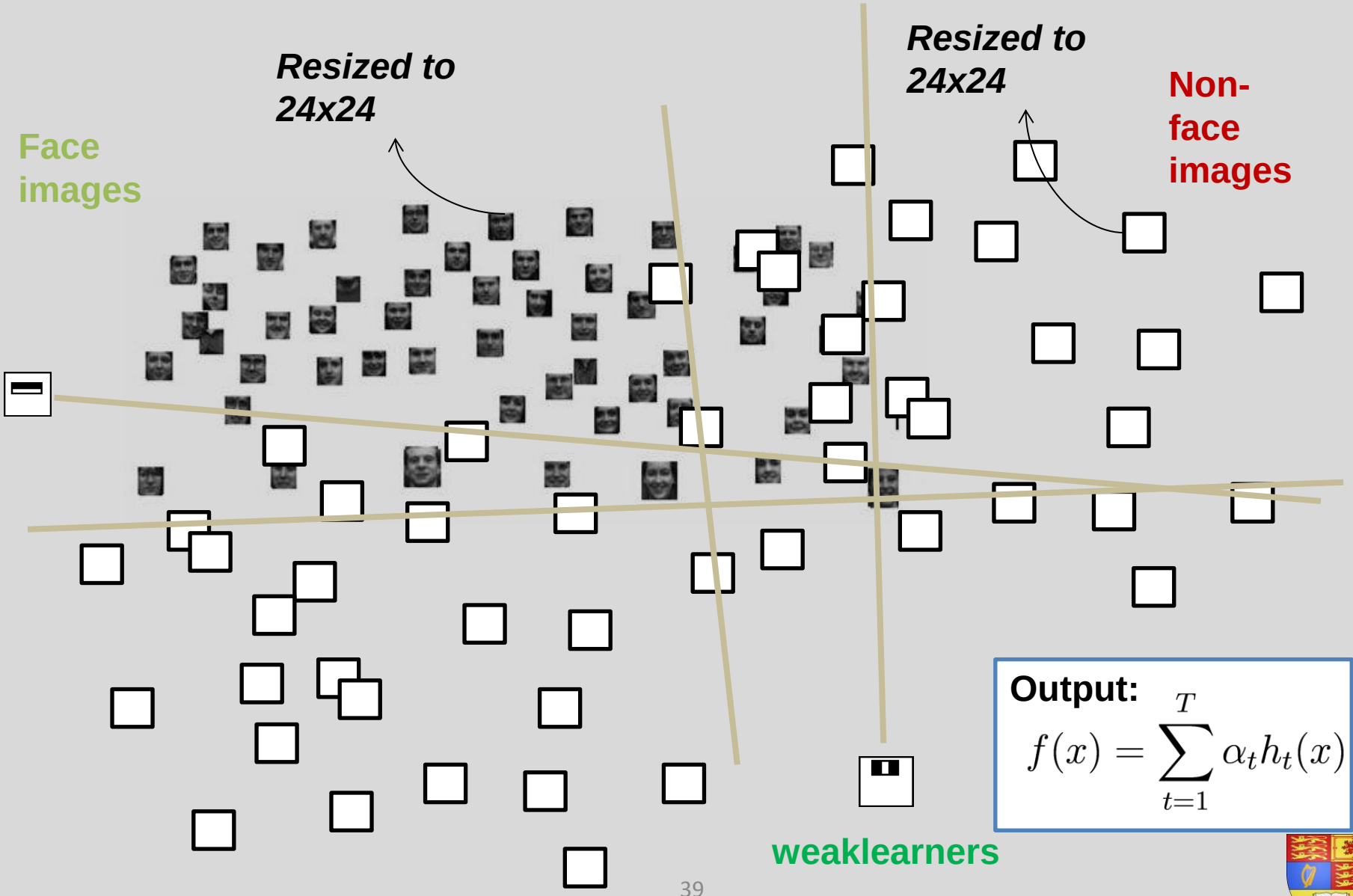


.....

24 pixels

24 pixels

Learning (concept illustration)



Output:

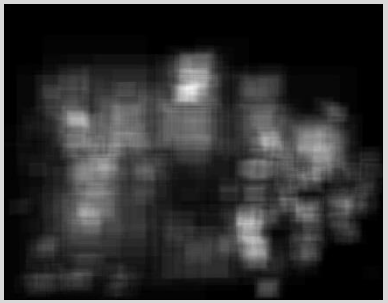
$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Evaluation (testing)

The learnt boosting classifier i.e. $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ is applied to every scan-window.

The **response map** is obtained, then non-local maxima suppression is performed.

Non-local maxima suppression



Receiver Operating Characteristic (ROC)

- Boosting classifier score (prior to the binary classification) is compared with a threshold.

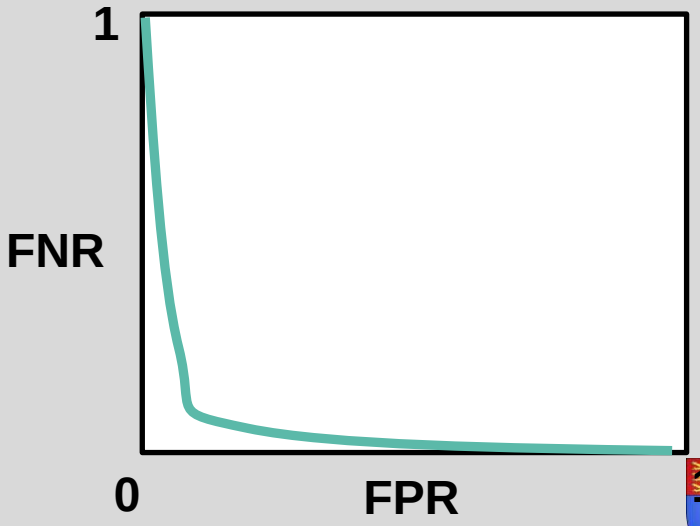
$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

> Threshold → Class 1 (face)

< Threshold → Class -1 (no face)

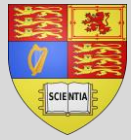
- The ROC curve is drawn by the false negative rate against the false positive rate at various threshold values:
  - False positive rate (FPR) = FP/N
  - False negative rate (FNR) = FN/P

where P positive instances,
N negative instances,
FP false positive cases, and
FN false negative cases.



Imperial College London

How to accelerate the boosting training and evaluation



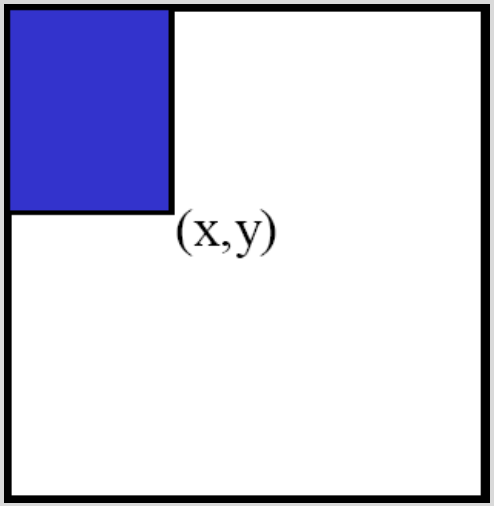
Integral Image

– A value at (x,y) is the sum of the pixel values above and to the left of (x,y).

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y'),$$

$$s(x,y) = s(x,y-1) + i(x,y)$$

$$ii(x,y) = ii(x-1,y) + s(x,y)$$

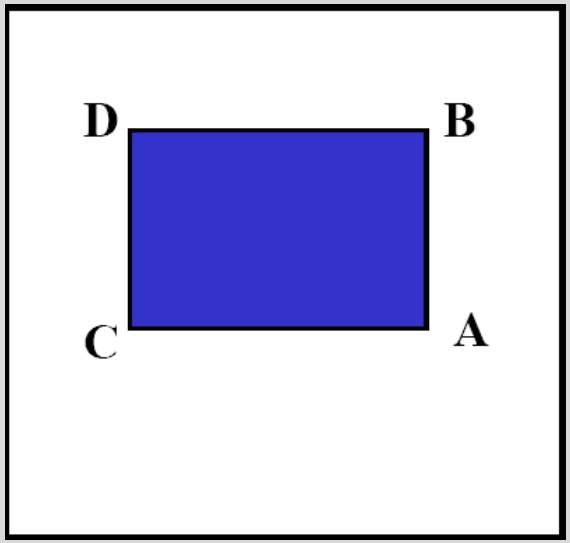


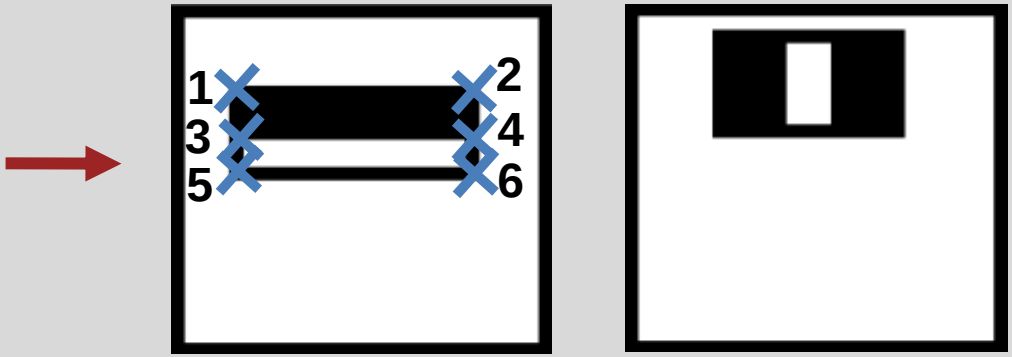
(where $s(x,y)$ is the cumulative row sum, $s(x,-1) = 0$, and $ii(-1,y) = 0$)

– The integral image can be computed in one pass over the original image.

Boosting Simple Features
[Viola and Jones CVPR 01]

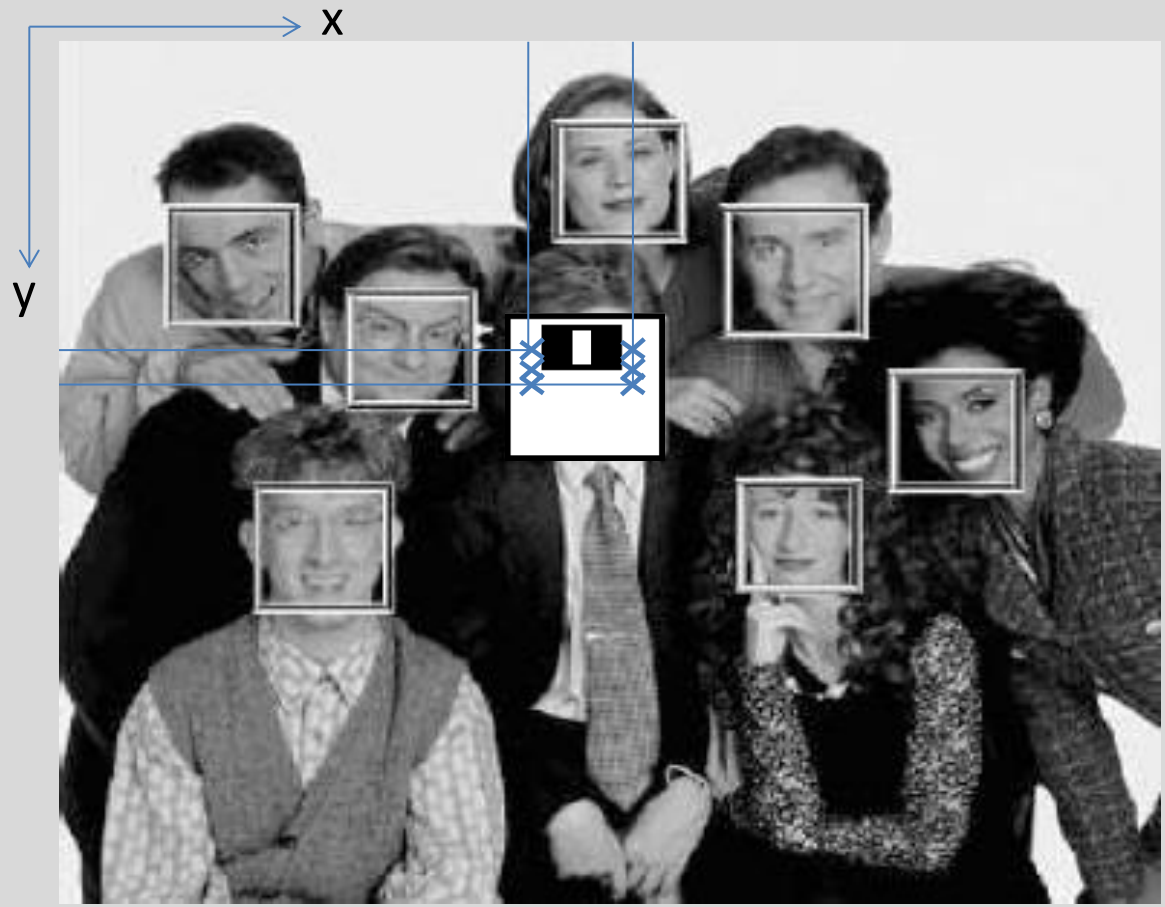
- Integral image
  - The sum of original image values within the rectangle can be computed: Sum = A-B-C+D
  - This provides the fast evaluation of Haar-basis like features





(6-4-5+3)-(4-2-3+1)

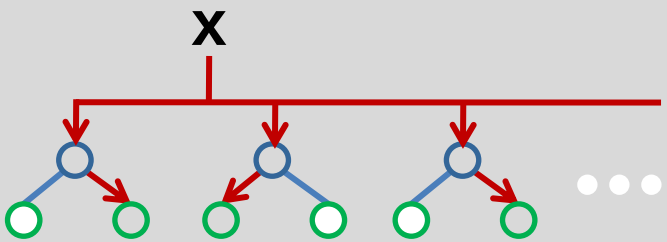
Evaluation (testing)



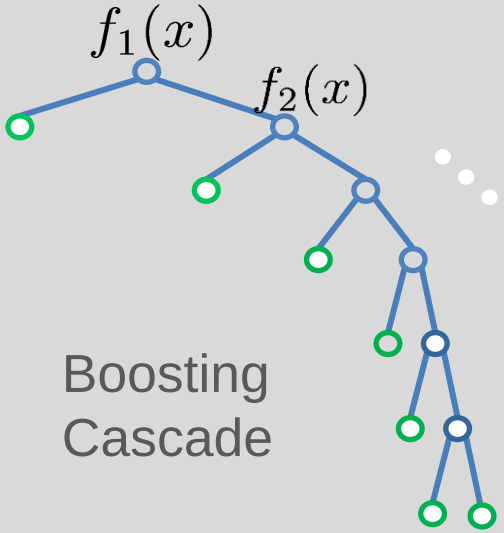
ii(x,y)

Imperial College
London

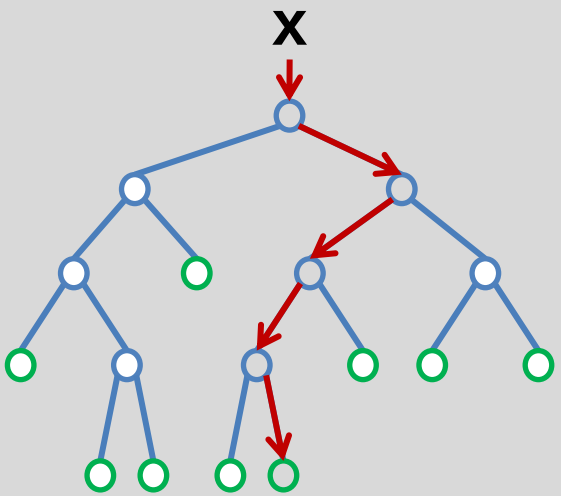
Further speeding up



Boosting



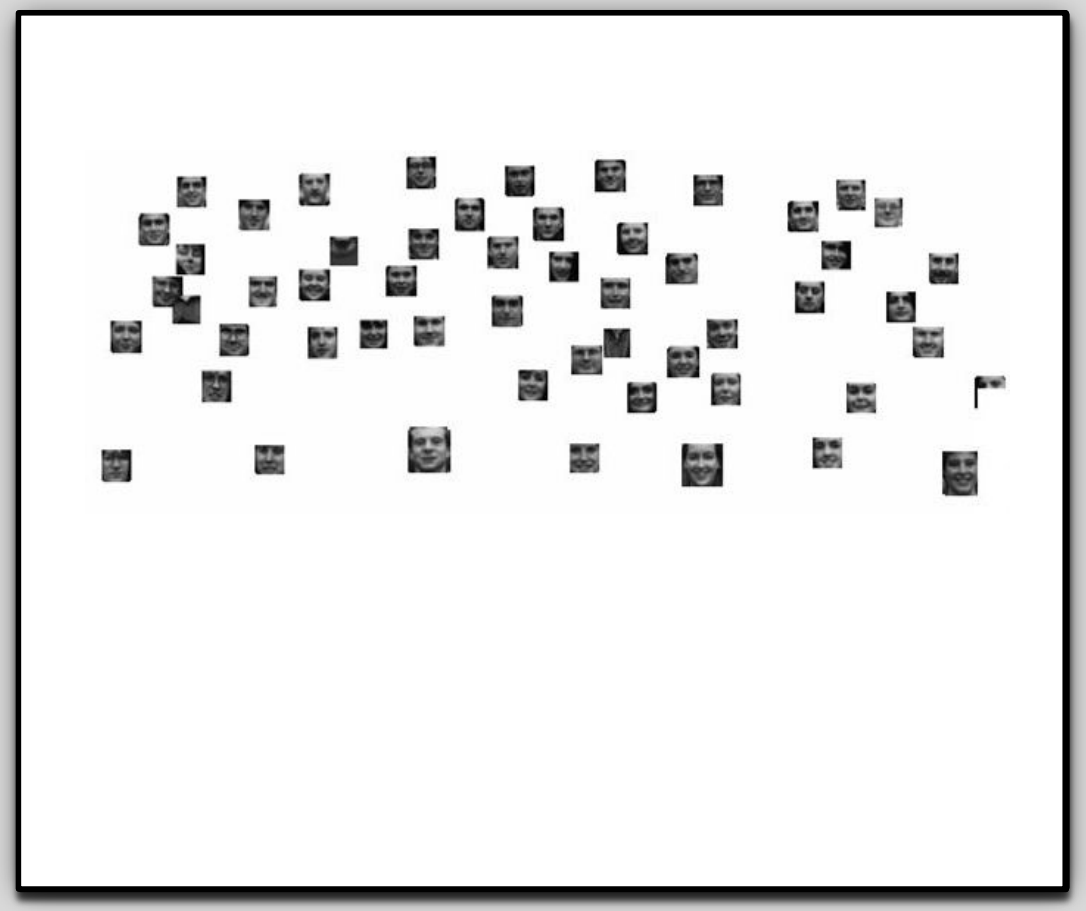
Boosting
Cascade

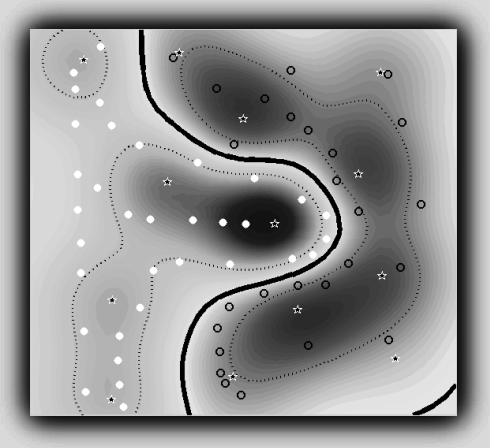


Tree structure Boosting

Object Detection by Boosting Cascade

It speeds up object detection by coarse-to-fine search.





Romdhani et al. ICCV01

A cascade of classifiers

- The detection system requires good detection rate and extremely low false positive rates.
- False positive rate and detection rate are

$$F = \prod_{i=1}^{K} f_i, \qquad D = \prod_{i=1}^{K} d_i,$$

f_i is the false positive rate of i-th classifier on the examples that get through to it.

- The expected number of features evaluated is

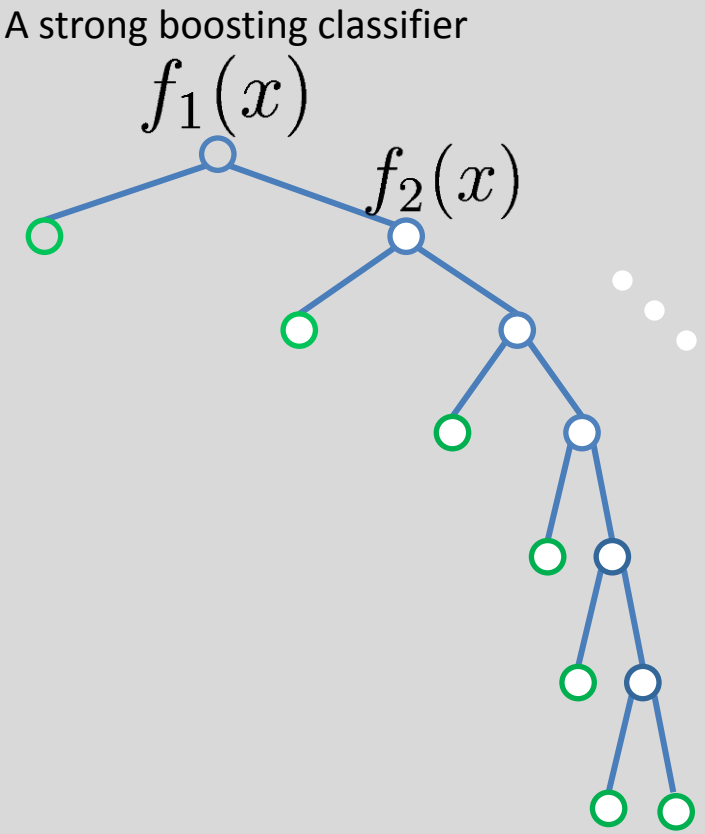
$$N = n_0 + \sum_{i=1}^{K} \left(n_i \prod_{j<i} p_j \right)$$

p_j is the proportion of windows input to i-th classifier.

Boosting Cascade

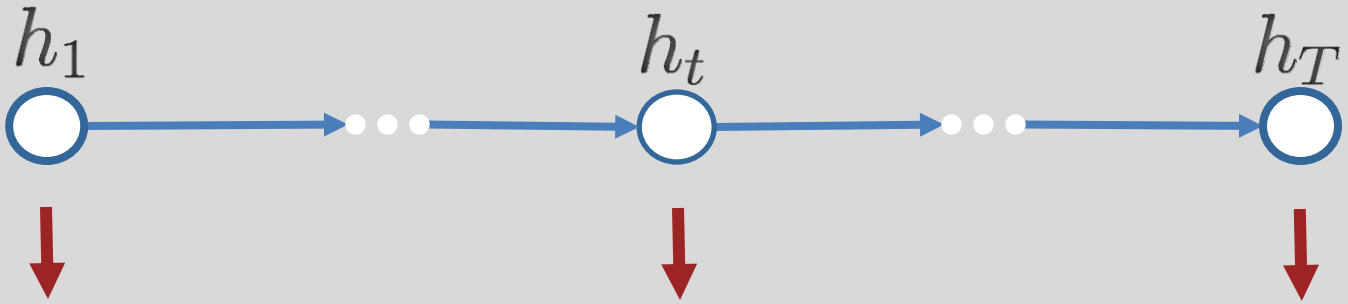
Boosting Cascade [viola & Jones 04], Boosting chain [Xiao et al]

- Very unbalanced tree
- Speeds up for unbalanced binary problems
- Hard to design



A sequential structure of varying length

- FloatBoost [Li et al PAMI04]
  - A backtrack mechanism deleting non-effective weak-learners at each round.
- WaldBoost [Sochman and Matas CVPR05]
  - Sequential probability ratio test for the shortest set of weak-learners for the given error rate.

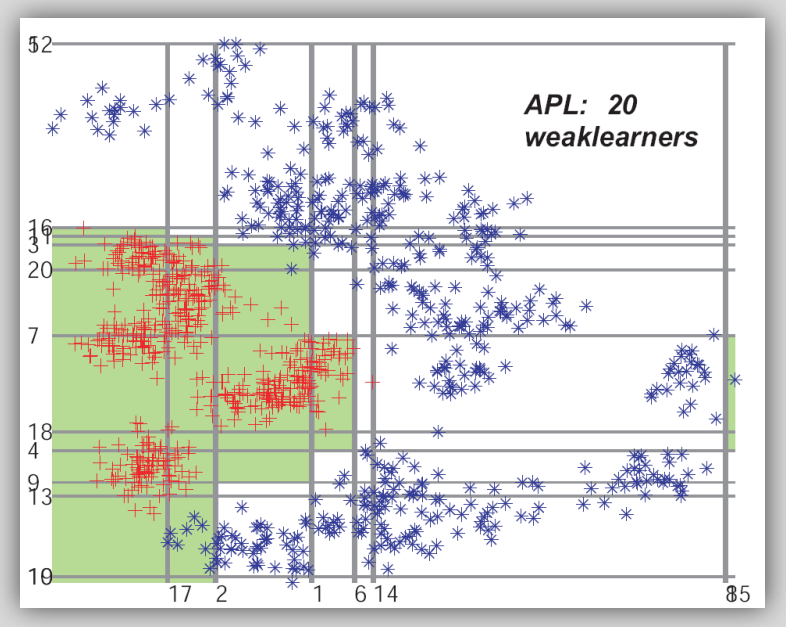


$$H_t(x) = \sum_{i=1}^{t} \alpha_i h_i(x) \geq \theta_B^{(t)}$$ Classify x as + and exit

$$H_t(x) \leq \theta_A^{(t)}$$ Classify x as - and exit

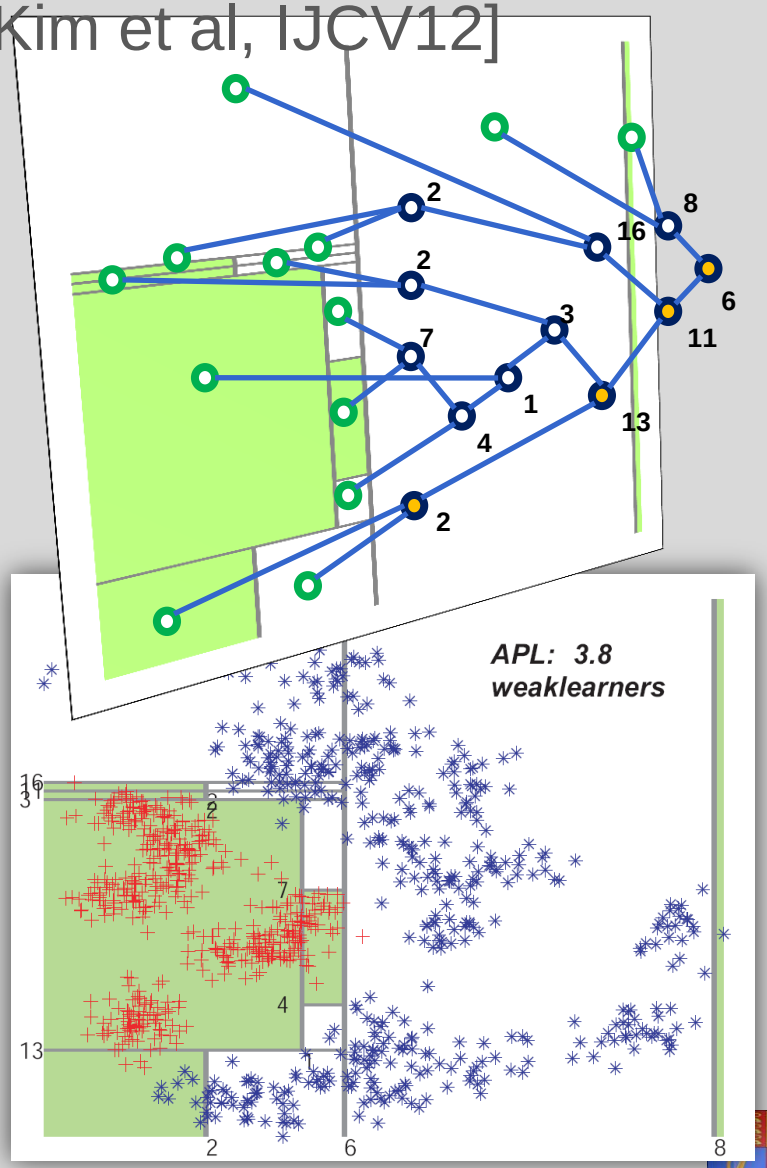
Converting a boosting classifier to a decision tree [Kim et al, IJCV12]

- Many short paths for speeding up
- Preserving (smooth) decision regions for good generalisation



Boosting

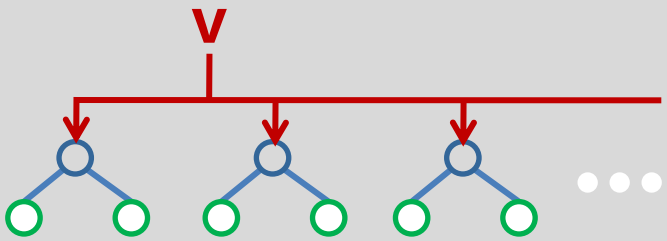
5 times speed up

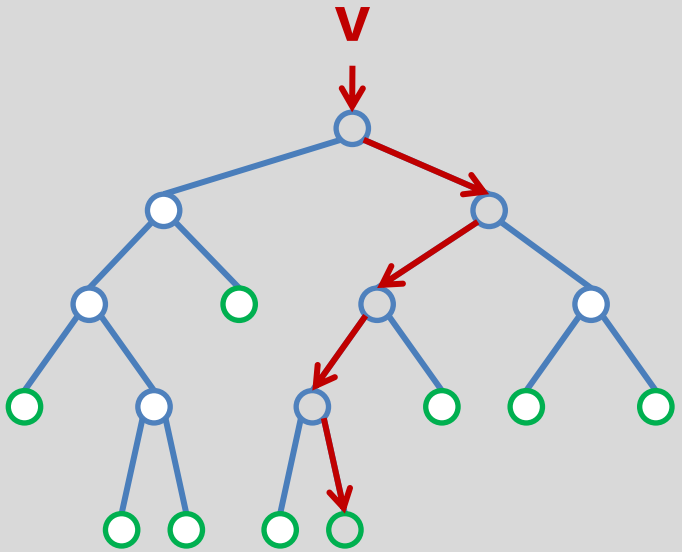


Super tree

Making a shallow network deep

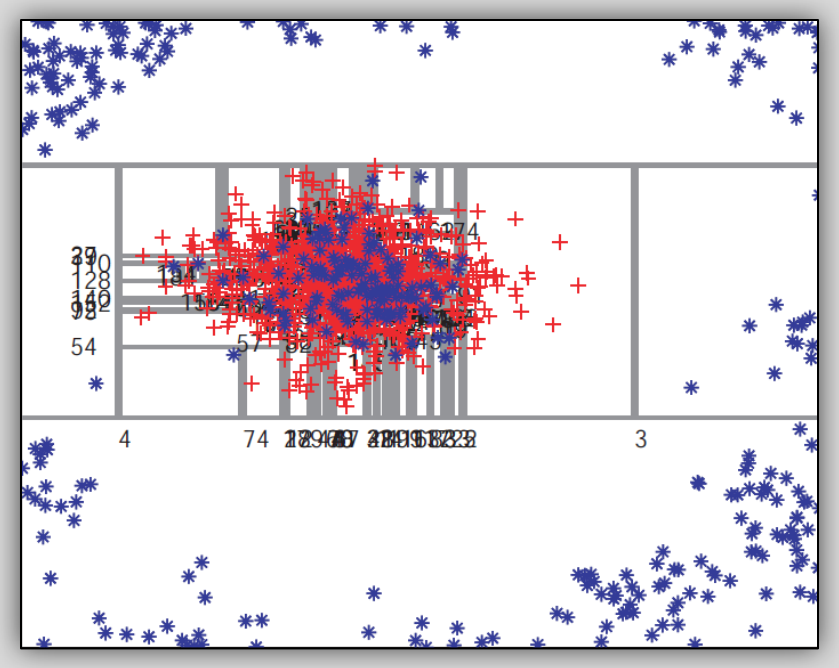
- Super tree [Kim, Budvytis, Cipolla, IJCV 2011]



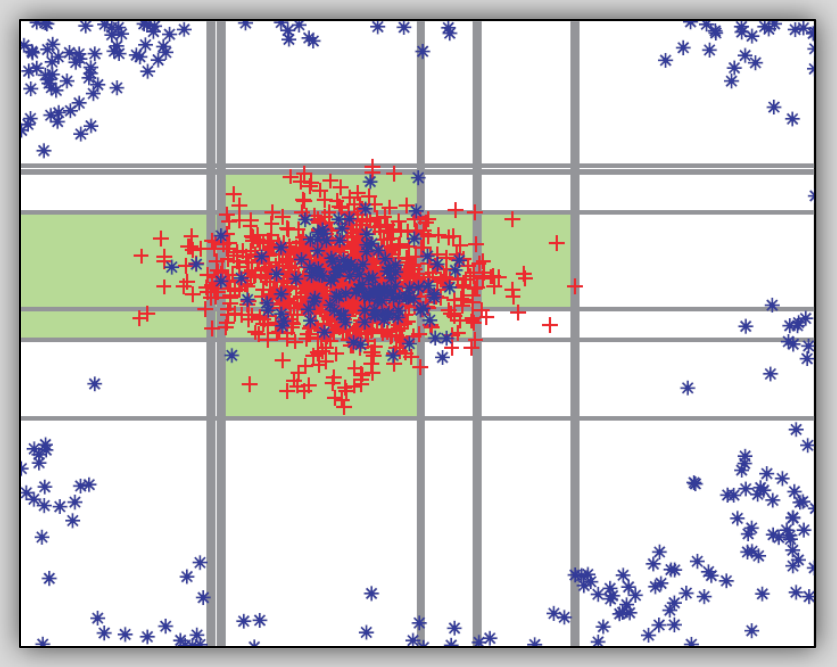


Converting a boosting classifier to a decision tree

- Many short paths for speeding up
- Preserving (smooth) decision regions for good generalisation



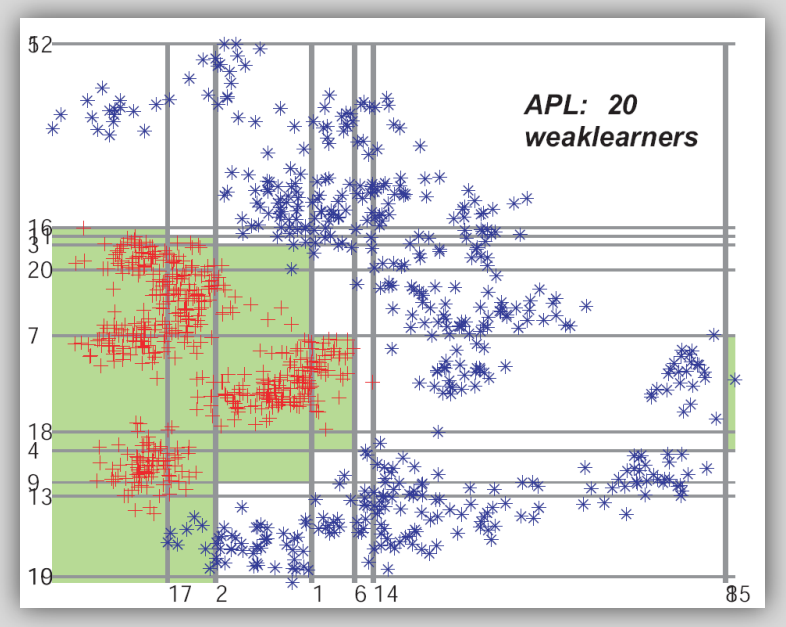
Decision tree



Boosting

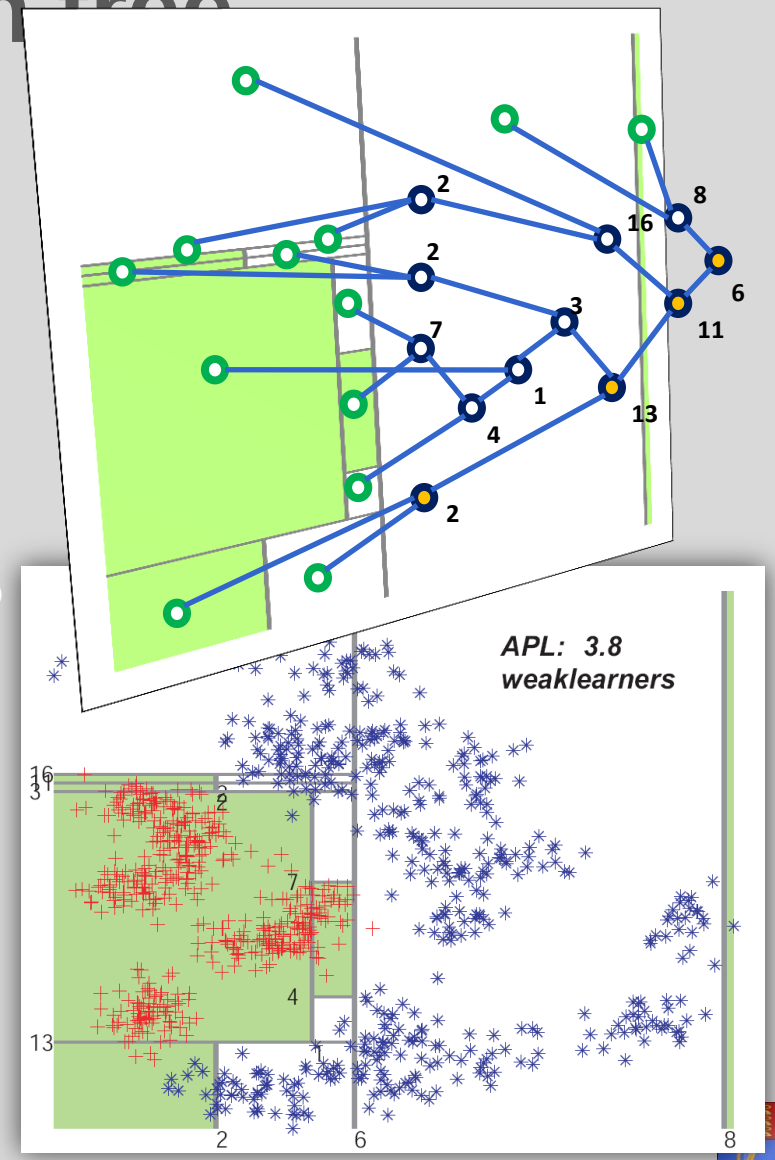
Converting a boosting classifier to a decision tree

- Many short paths for speeding up
- Preserving (smooth) decision regions for good generalisation



Boosting

5 times speed up



Super tree


Super tree on
2D toy data
with 3D visualisation

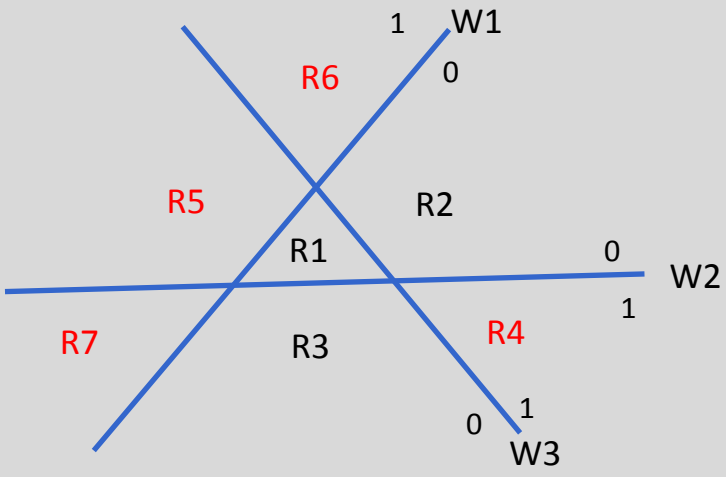
Boolean optimisation formulation

- For a learnt boosting classifier

$$H(x) = \sum_{i=1}^{m} \alpha_i h_i(x)$$

split a data space into 2^m primitive regions by m binary weak-learners.

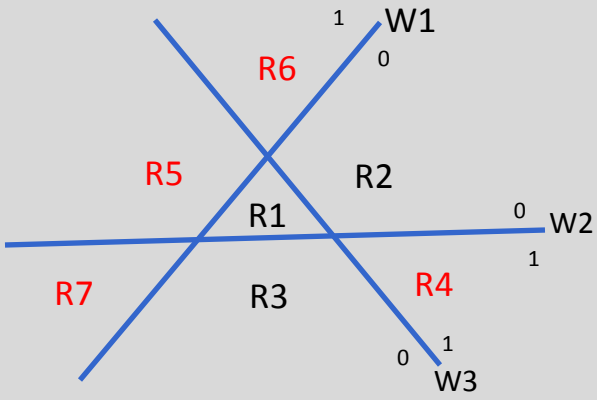
- Code regions R_i i=1,..., 2^m by boolean expressions.



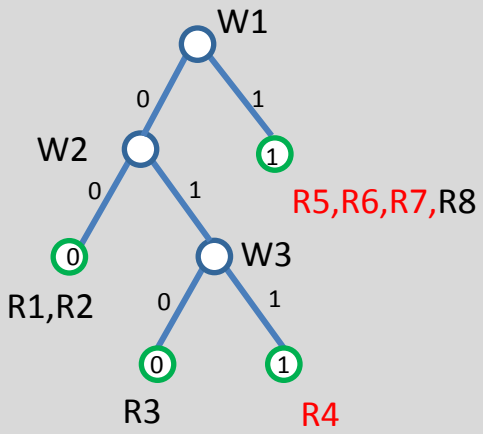
	W1	W2	W3	C
R1	0	0	0	0
R2	0	0	1	0
R3	0	1	0	0
R4	0	1	1	1
R5	1	0	0	1
R6	1	0	1	1
R7	1	1	0	1
R8	1	1	1	x

Boolean expression minimization

- Optimally joining the regions of the same class label or *don't care* label.
- A short tree built from the minimised boolean expression.



	W1	W2	W3	C
R1	0	0	0	0
R2	0	0	1	0
R3	0	1	0	0
R4	0	1	1	1
R5	1	0	0	1
R6	1	0	1	1
R7	1	1	0	1
R8	1	1	1	x



$\overline{W}1W2W3 \vee W1\overline{W}2\overline{W}3 \vee W1\overline{W}2W3 \vee W1W2\overline{W}3 \rightarrow W1 \vee \overline{W}1W2W3$

Boolean optimisation formulation

- Optimally short tree defined with average expected path length of data points

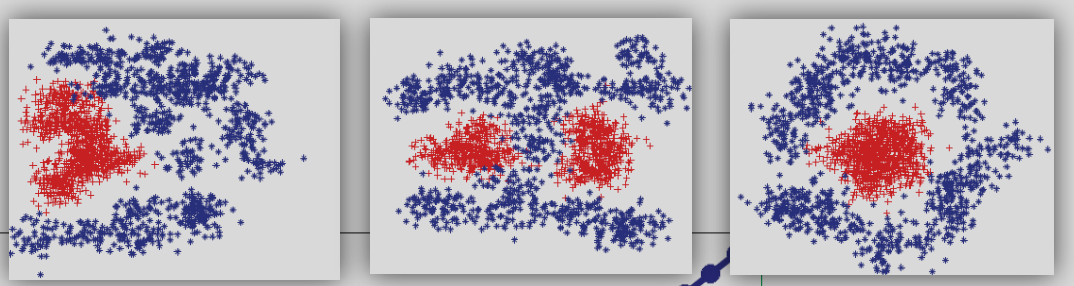
$$T^* = \arg\min_T \sum_i E(l_T(R_i))p(R_i)$$

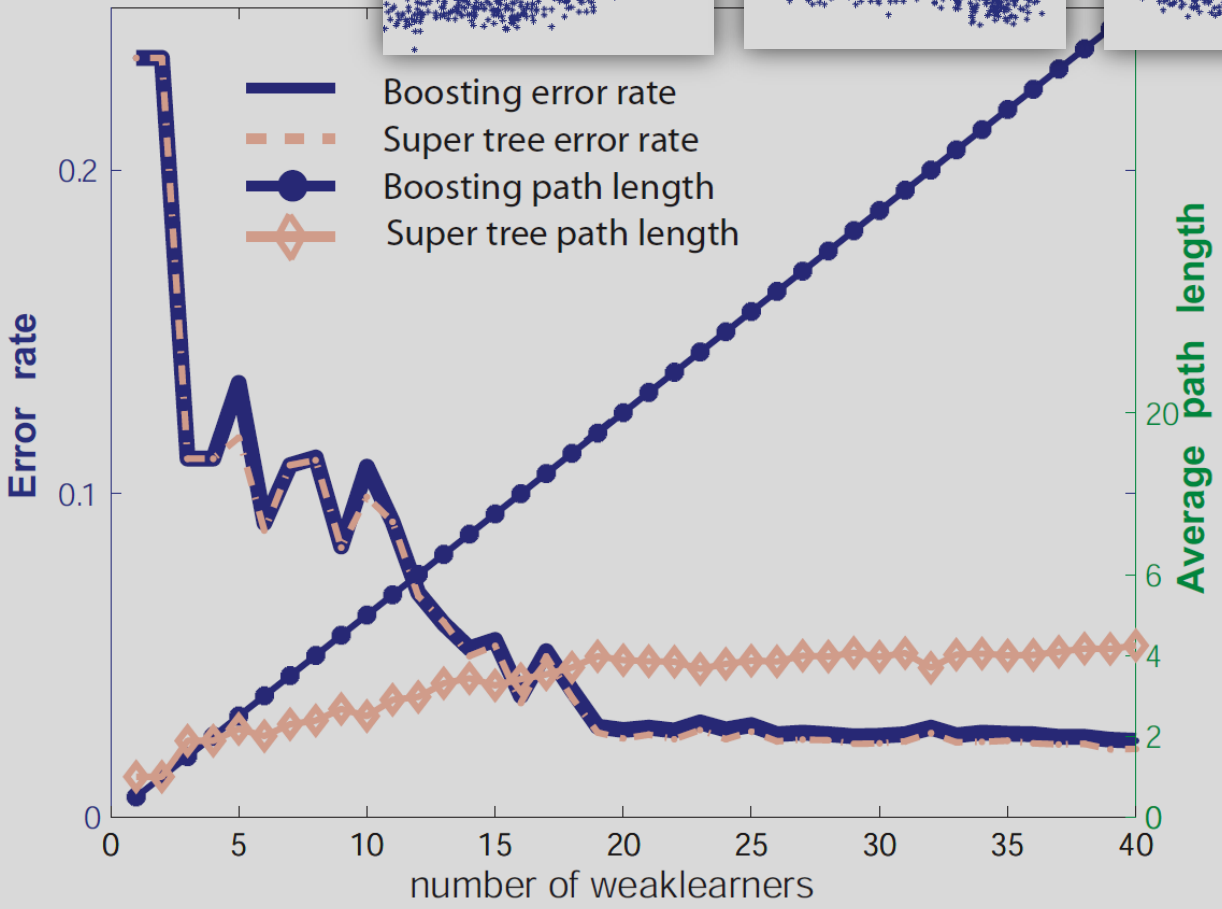
where $p(R_i)=M_i/M$ and the tree must duplicate the Boosting decision regions.

- Solutions
  - Boolean expression minimization
  - Growing a tree from the decision regions
  - Extended region coding

Synthetic data exp1

Examples generated from GMMs





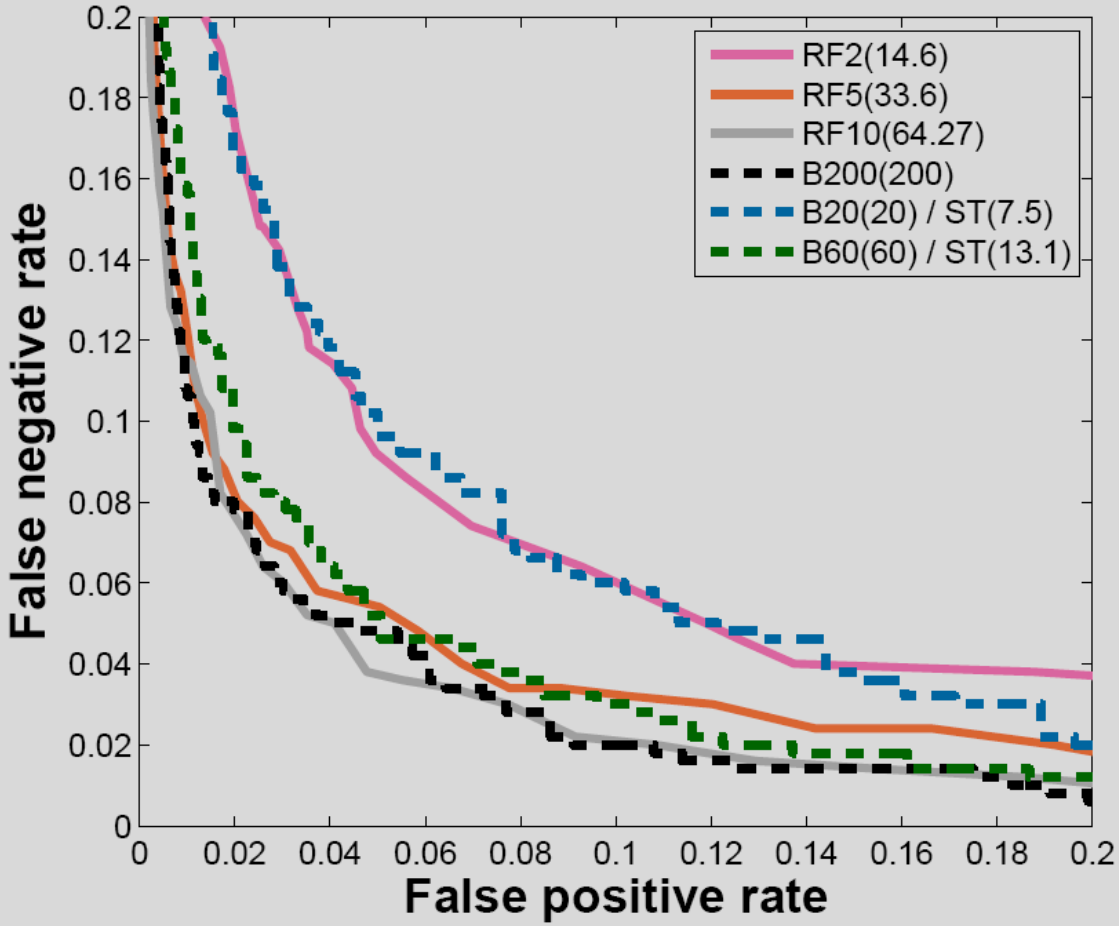
Face detection experiment

Total data points = 57499

	Boosting			Fast exit [Zhou 05]			Super tree		
No. of weak learners	False positive rate	False negativer ate	Average path length	False positive rate	False negativer ate	Average path length	False positiv e rate	False negative rate	Average path length
20	501	120	20	501	120	11.70	476	122	7.51
40	264	126	40	264	126	23.26	258	128	15.17
60	222	143	60	222	143	37.24	246	126	13.11
100	148	146	100	148	146	69.28	145	152	15.1
200	120	143	200	120	143	146.19	128	146	15.8

- The proposed solution is about 3 to 5 times faster than boosting and 1.5 to 2.8 times faster than [Zhou 05], at the similar accuracy.

ST vs Random forest

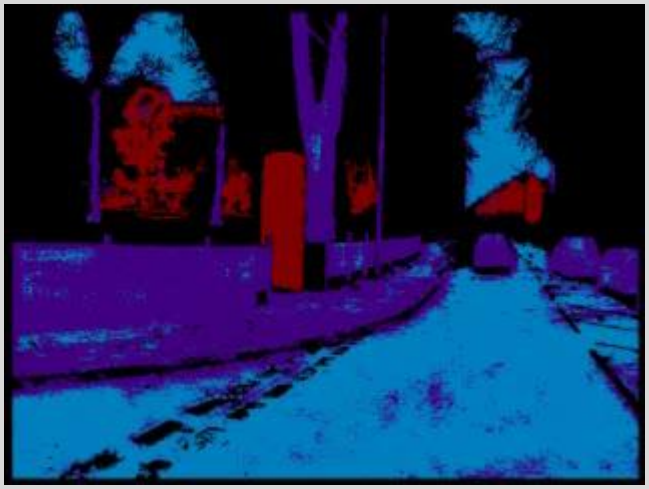


- ST is about 2 to 3 times faster than RF at similar accuracy

Imperial College London

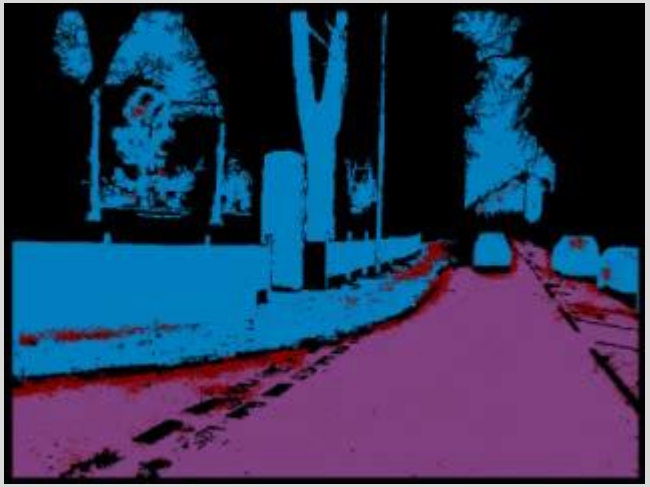
Segmentation results

Building Non-building Error



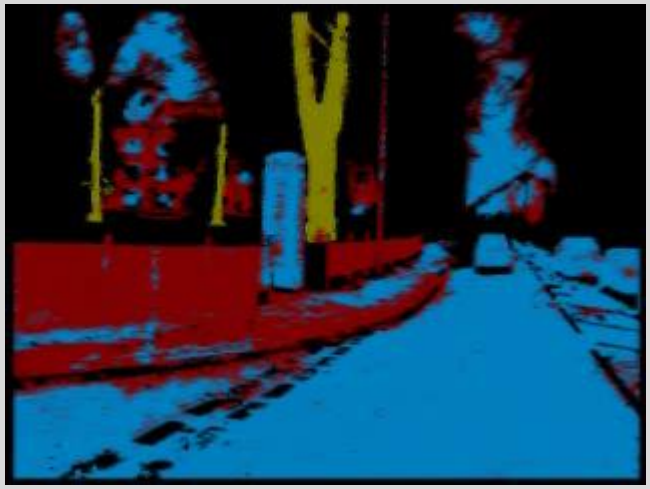
global : 74.50%
average: 79.89%

Road Non-road Error



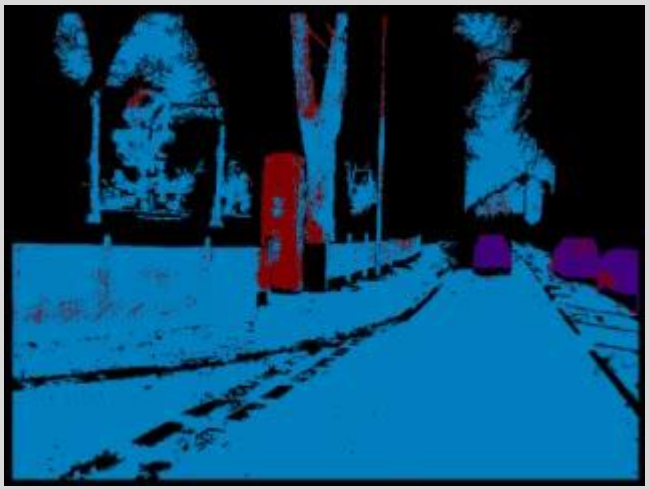
global: 88.33%,
average: 87.26%

Tree Non-tree Error



global: 77.46%,
average: 80.45%

Car Non-car Error



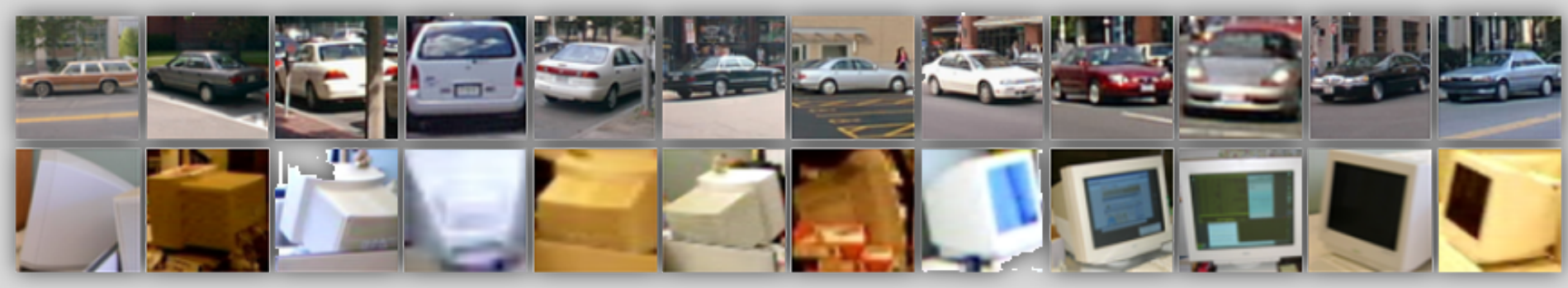
global: 85.55 %,
average: 85.24 %

Imperial College
London

Multi-Class Boosting

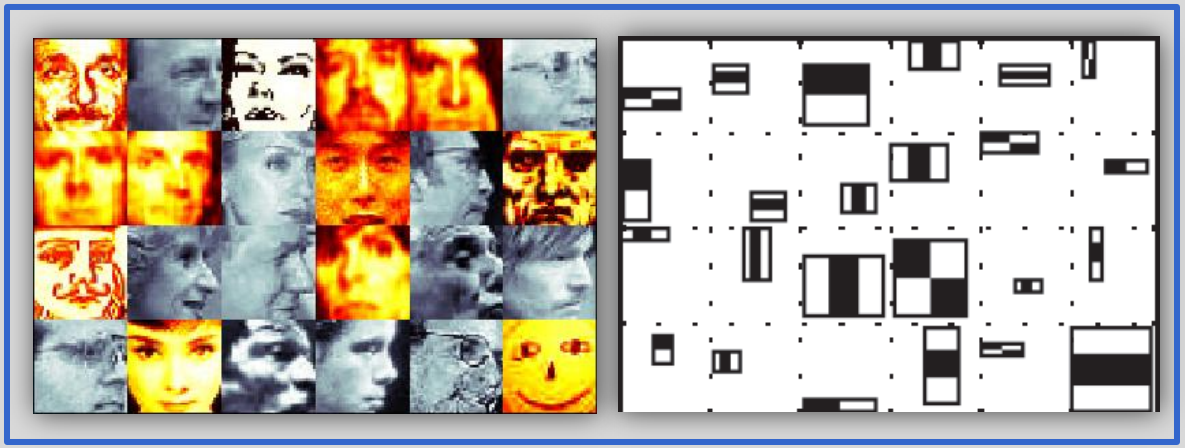
Multi-view and multi-category object detection

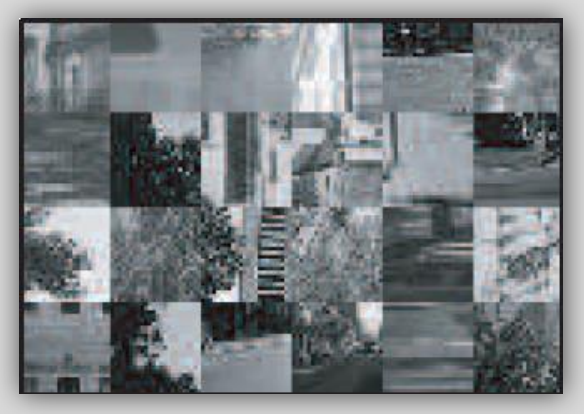
- Images exhibit multi-modality.
- A single boosting classifier is not sufficient.
- Manual labeling sub-categories.



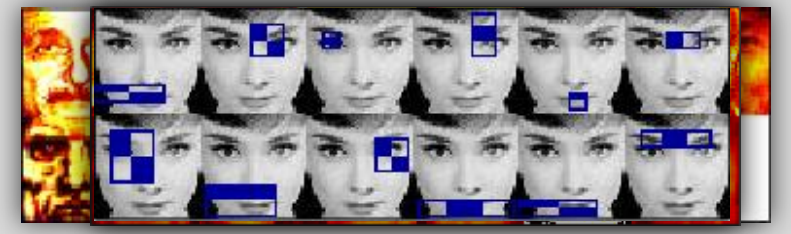
Images from Torralba et al 07

Problem

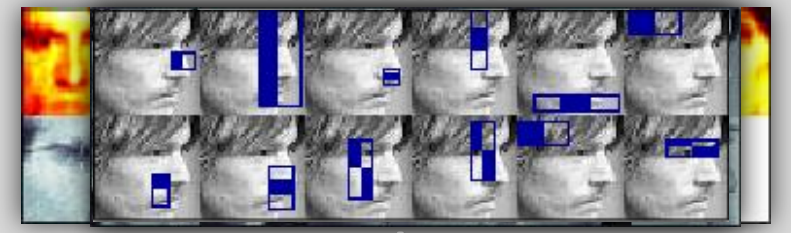




K-means
MCBoost
clustering



Face cluster 1



Face cluster 2

MCBoost: Multiple Classifier Boosting [Kim et al NIPS08]

- Objective ftn: $J = \log \prod_i P(x_i)^{y_i}(1 - P(x_i))^{(1-y_i)}, \ y_i \in \{0, 1\}$

- The joint prob. as a Noisy-OR:

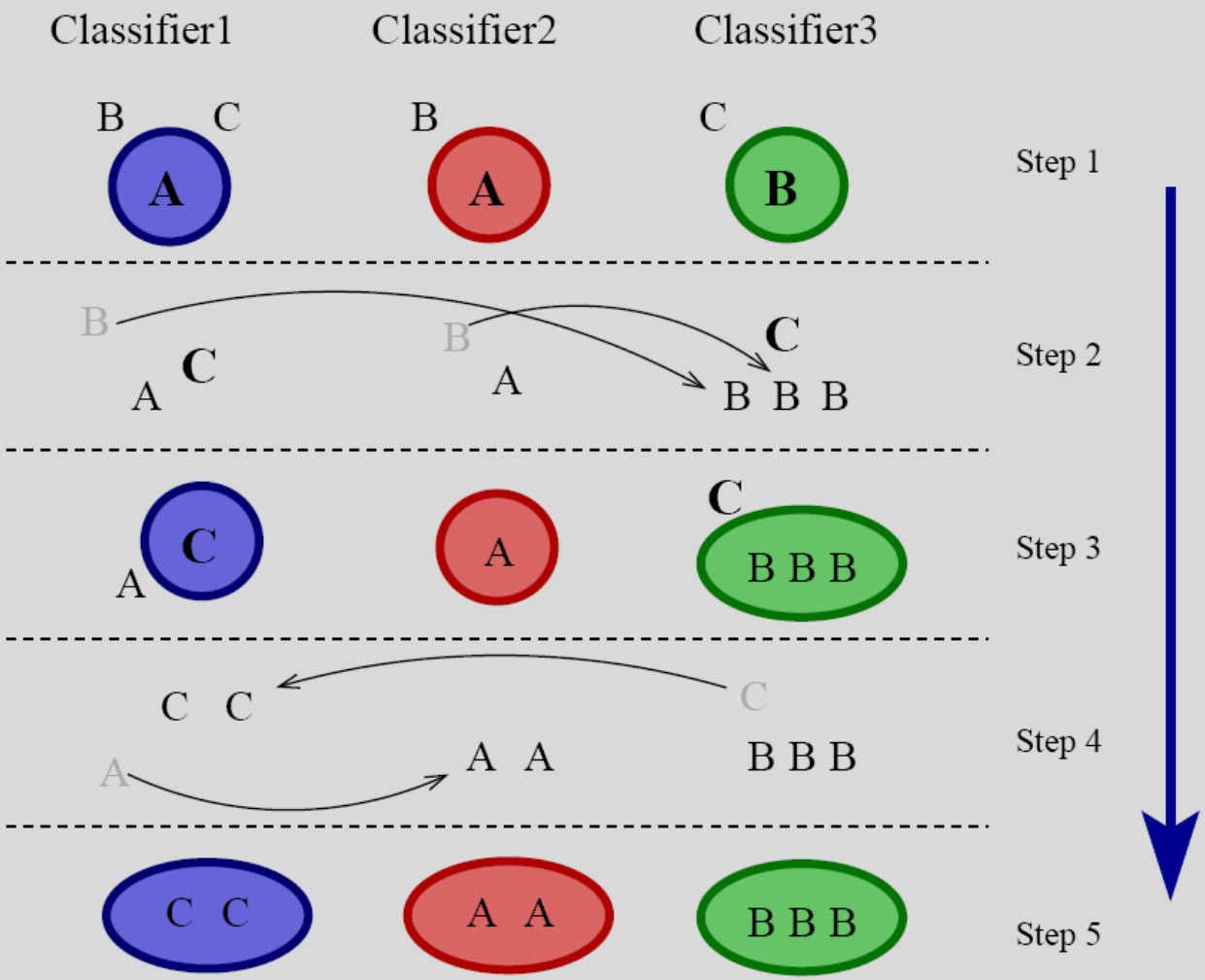
$$P(x_i) = 1 - \prod_k (1 - P_k(x_i)), \ k = 1, ..., K$$

where $P_k(x_i) = 1/(1 + \exp(-H_k(x_i))), \ H_k(x_i) = \sum_t \alpha_{kt} h_{kt}(x_i)$

- By AnyBoost Framework [Mason et al. 00]
  - For t=1 to T
    - For k=1 to K
      - Update sample weights

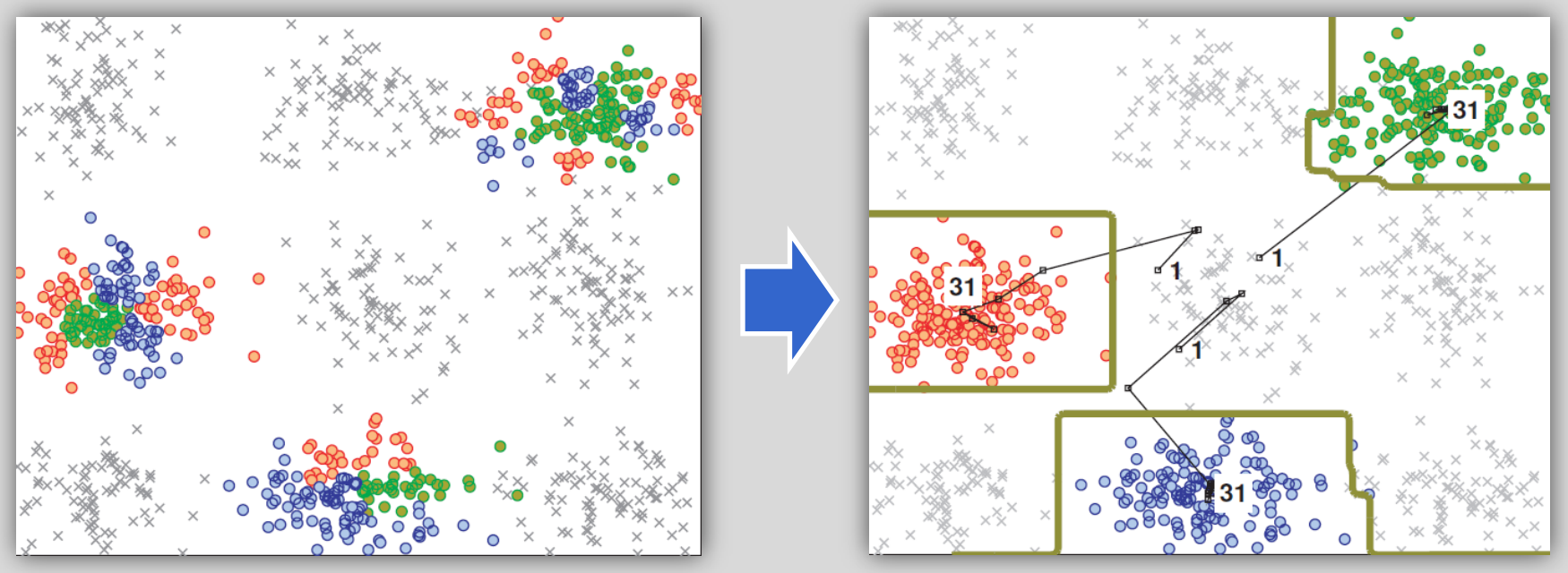
$$w_{ki} = \frac{\partial J}{\partial H_k(x_i)} = \frac{y_i - P(x_i)}{P(x_i)} \cdot P_k(x_i)$$

State diagram

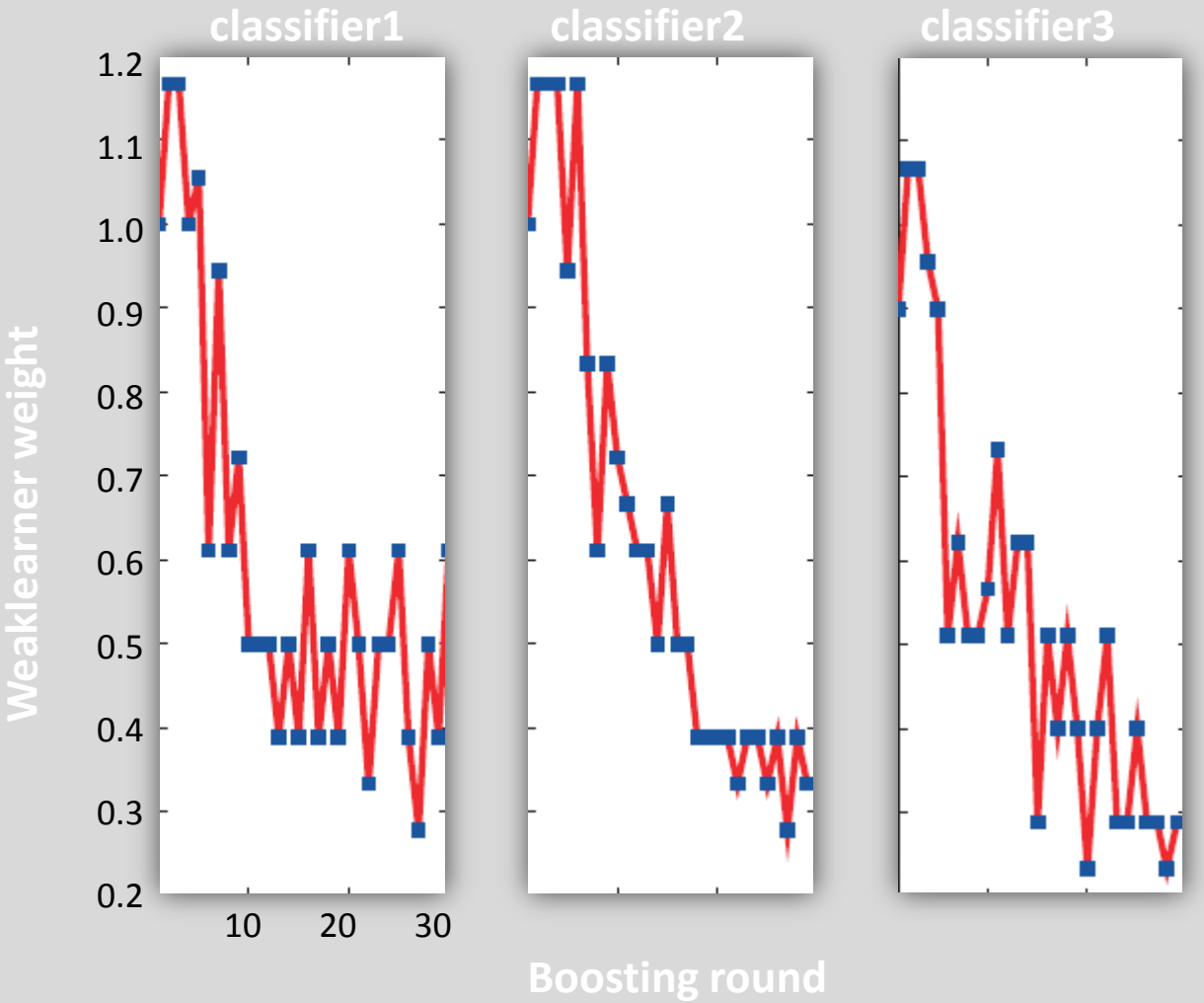


Toy XOR classification problem

- Discriminative clustering positive samples



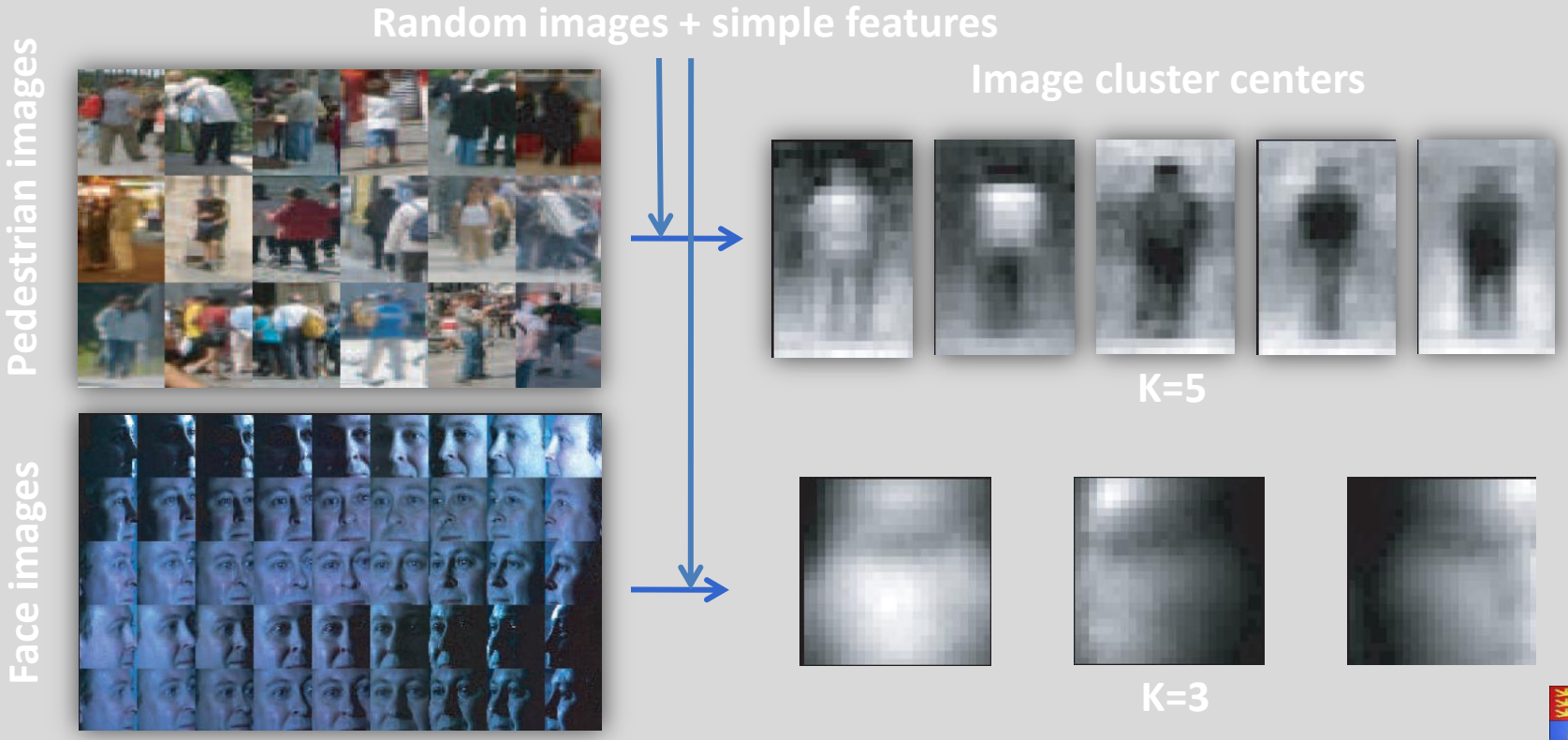
Toy XOR classification problem



- Matlab demo

Experiments

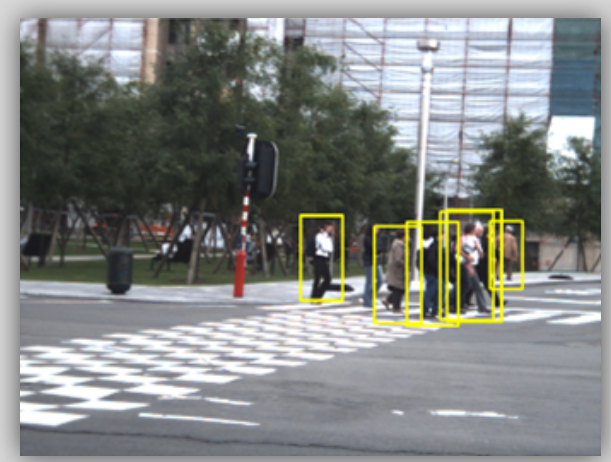
- INRIA pedestrian data set containing 1207 pedestrian images and 11466 random images.
- PIE face data set involving 1800 face images and 14616 random images.
- A total number of 21780 simple rectangle features.

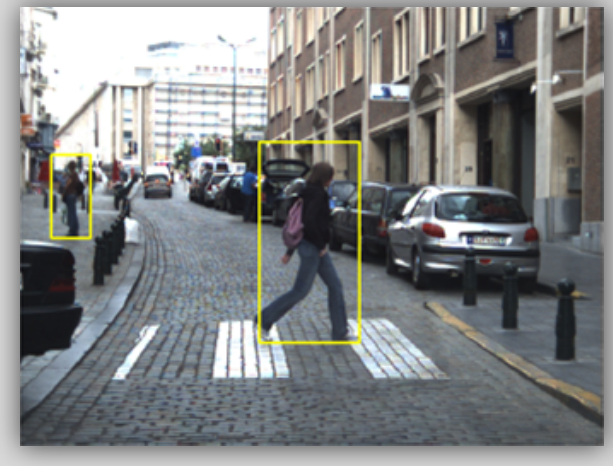


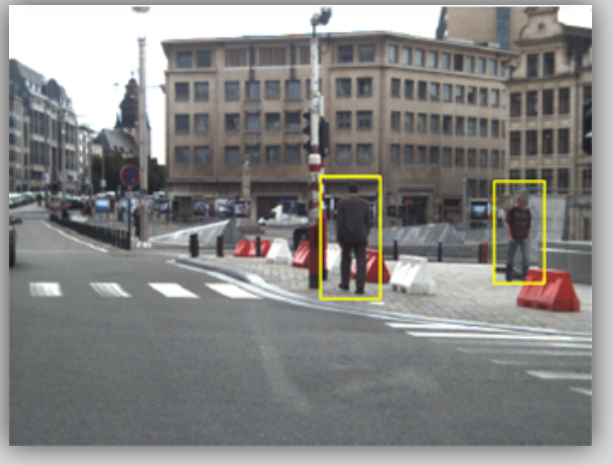
Pedestrian detection by MCBoost

[Wojek, Walk, Schiele et al CVPR09]

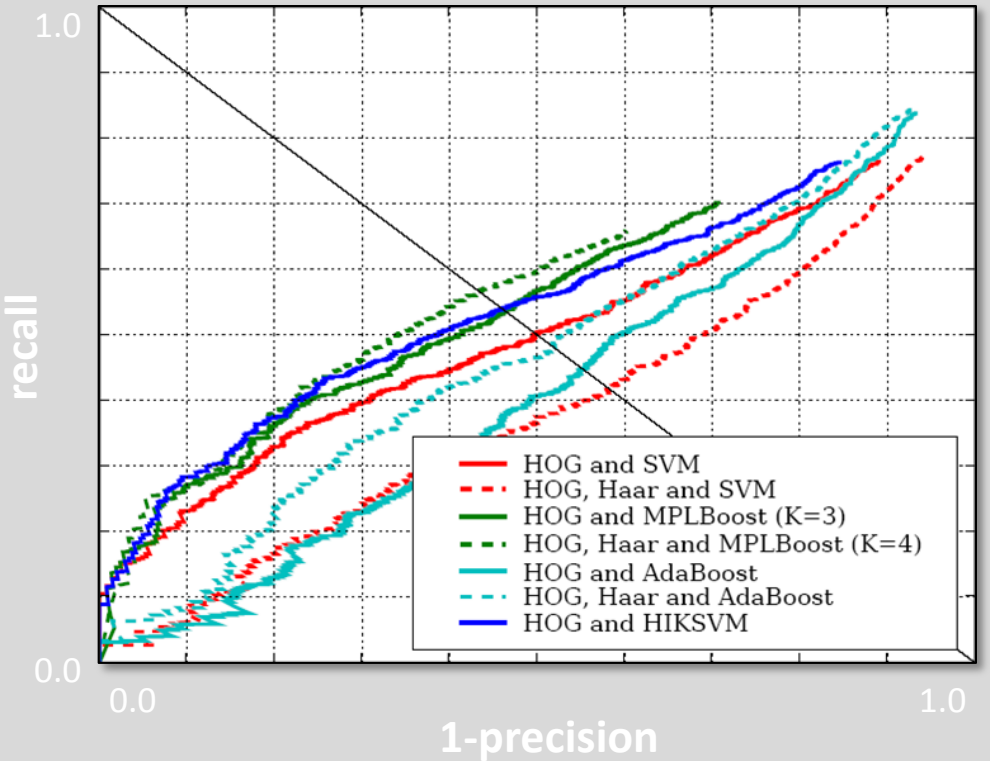
- TUD Brussels onboard dataset

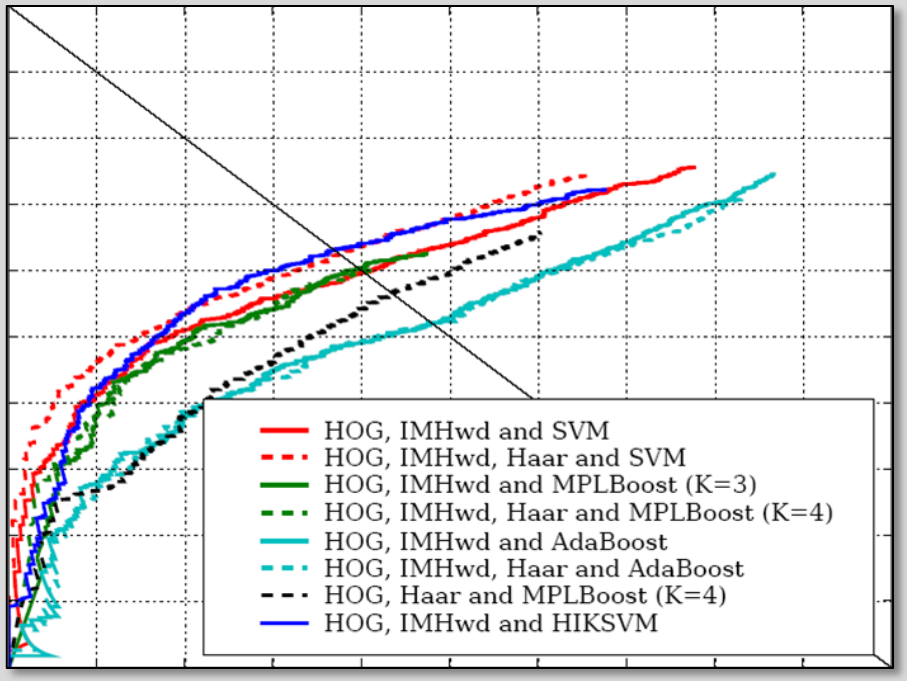






Pedestrian detection by MCBoost [Wojek, Walk, Schiele et al CVPR09]

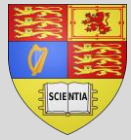




Imperial College London

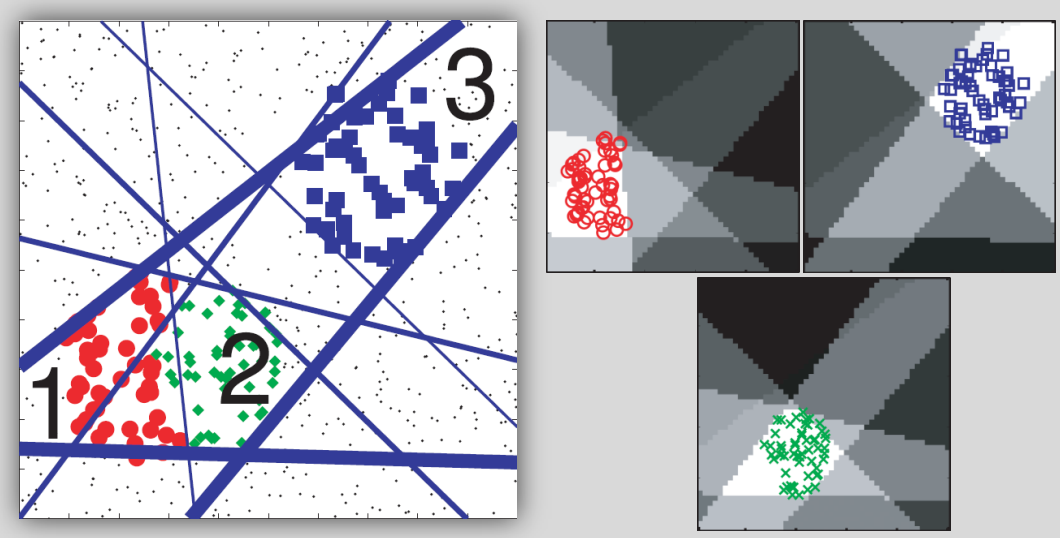
Multi-class Boosting + Speeding up

Torralba et al PAMI 07
Wu et al ICCV07
Huang et al ICCV05
Tu ICCV05
Grossmann 04

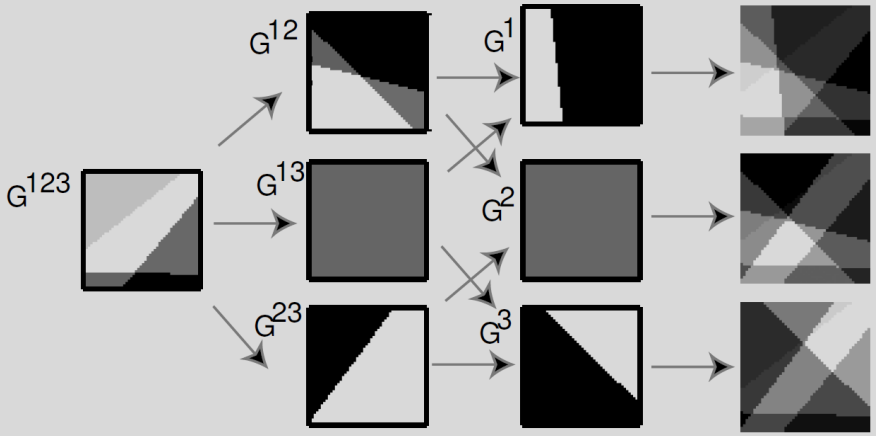


Multiclass object detection [Torralba et al PAMI 07]

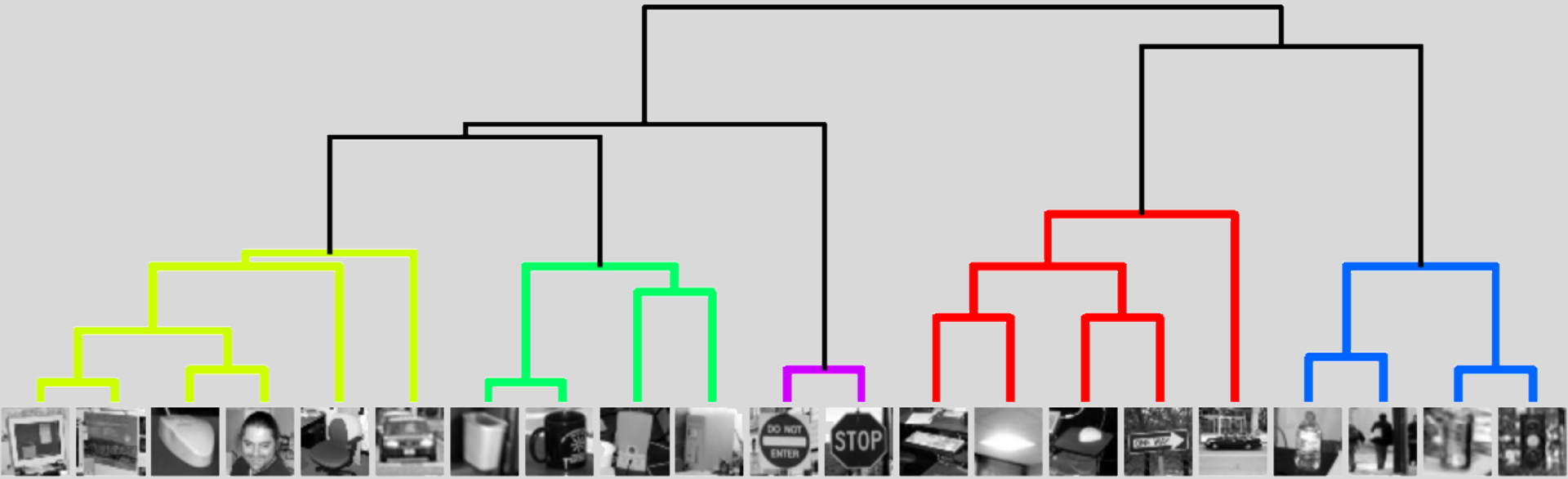
- Learning multiple boosting classifiers by sharing features

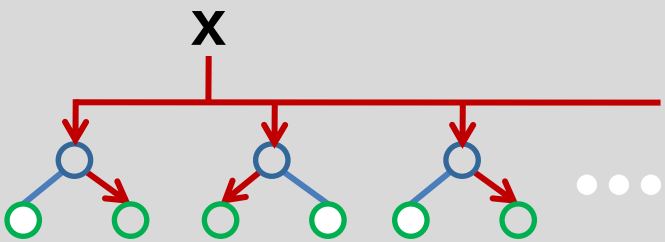


- Tree structure speeds up the classification

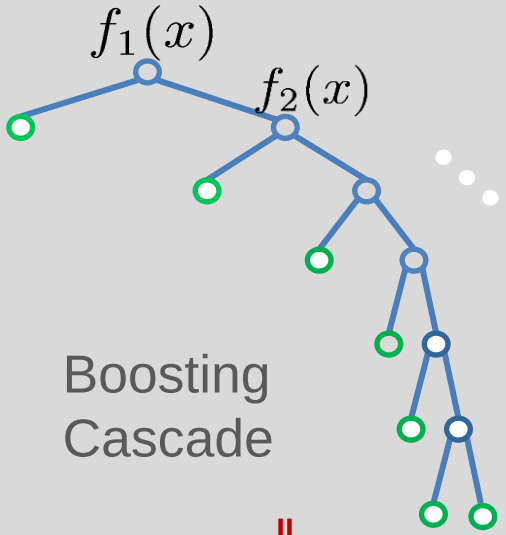


Multiclass object detection [Torralba et al PAMI 07]

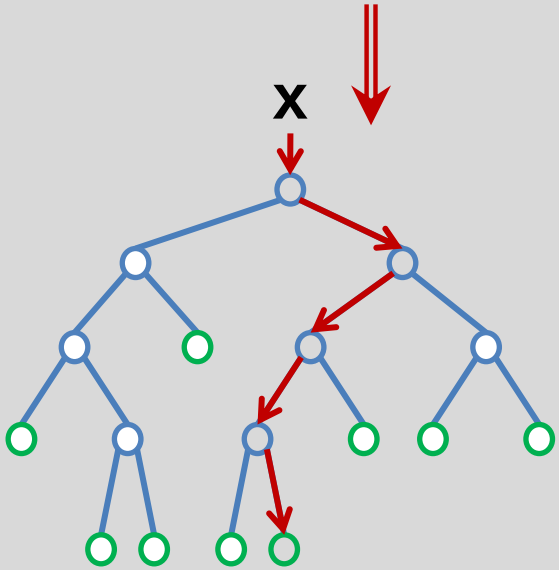




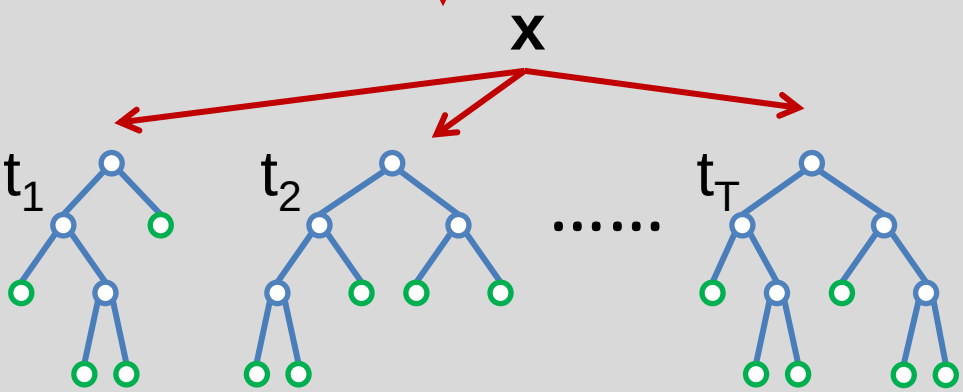
Boosting



Boosting Cascade



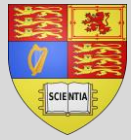
Tree structure Boosting



Randomised Decision Forest

Part II. Randomised Decision Forests

Basics and Motivations

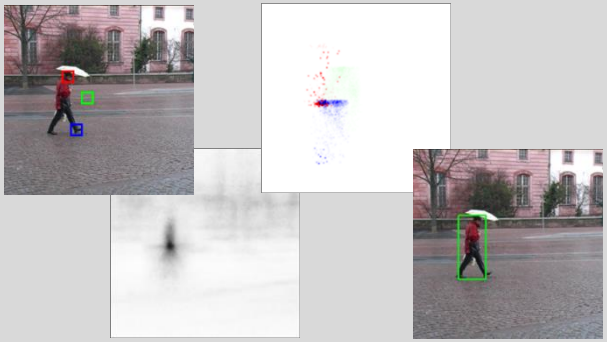


Randomised Forests in the field

KINECT

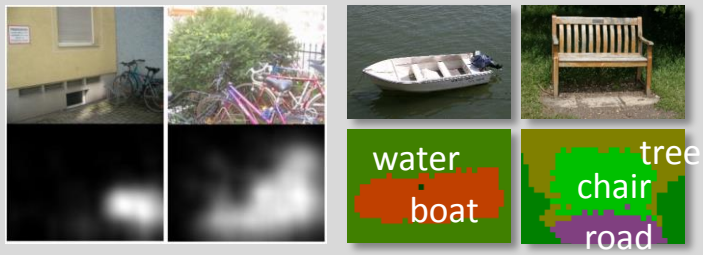
[Shotton *et al.*, 11]

Hough Forest



[Gall *et al.*, 09]

Visual Words



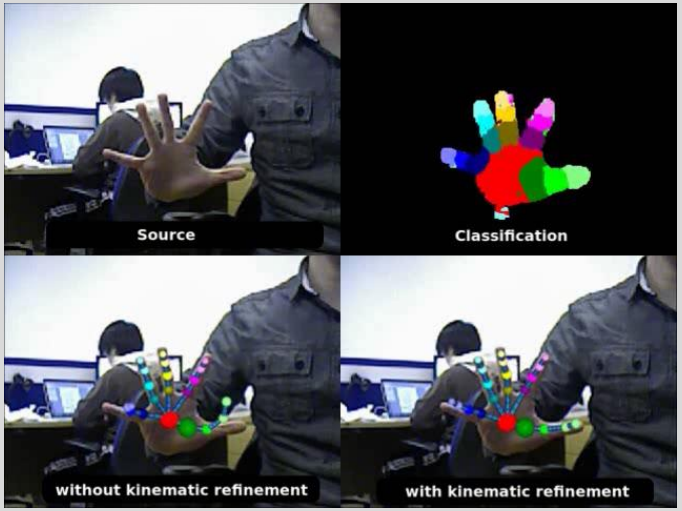
[Moosmann *et al.*, 06]
[Shotton *et al.*, 08]

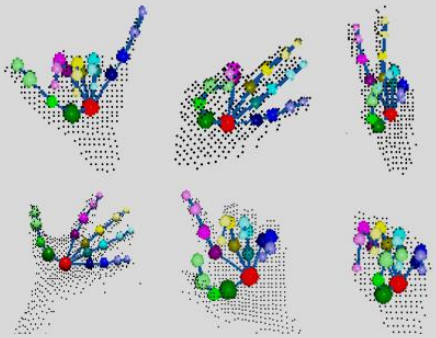
Keypoint Recognition

[Lepetit *et al.*, 06]

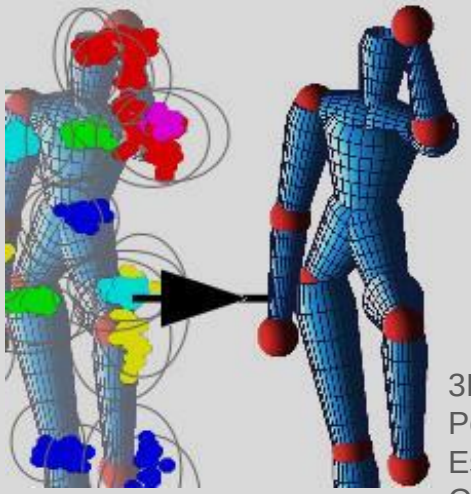
Imperial College London

Randomised Forests @ ICL

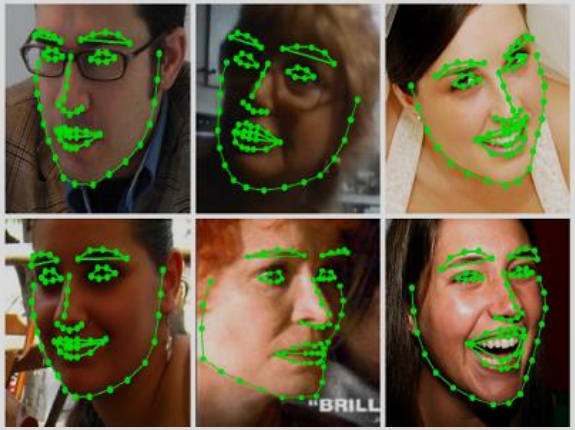




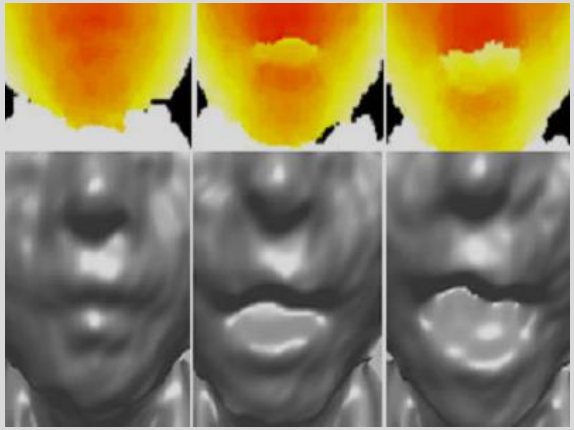
Articulated Hand Pose Estimation
ICCV13 (oral), CVPR14 (oral)



3D Body Pose Estimation
CVPR13



Unified Face Analysis, CVPR14



Lip Reading, ICCV13

http://www.iis.ee.ic.ac.uk/ComputerVision/Publications

Imperial College London

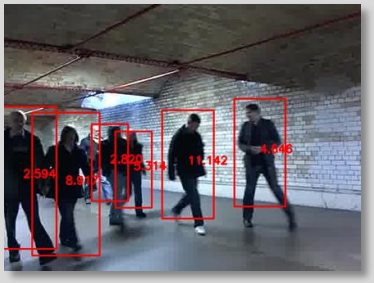
Randomised Forests @ ICL

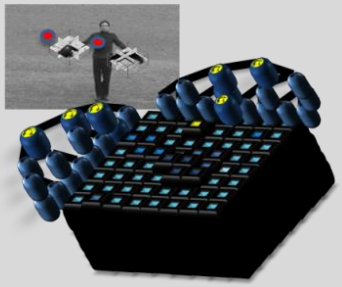
Object Detection in Depth Images, ECCV14

Active RF, ECCV14

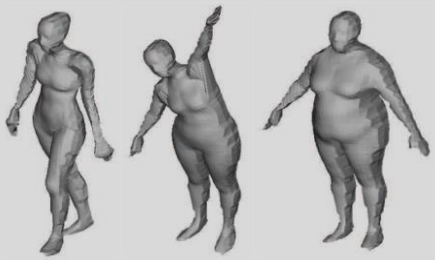
Object Recognition in Videos, CVPRW14

Pedestrian Detection, BMVC12

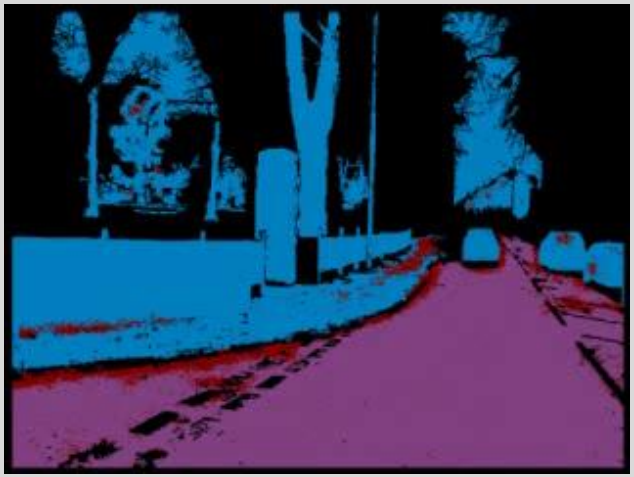




Action Recognition, BMVC10



Shape and Pose Estimation
ECCV 10, ICCV 11

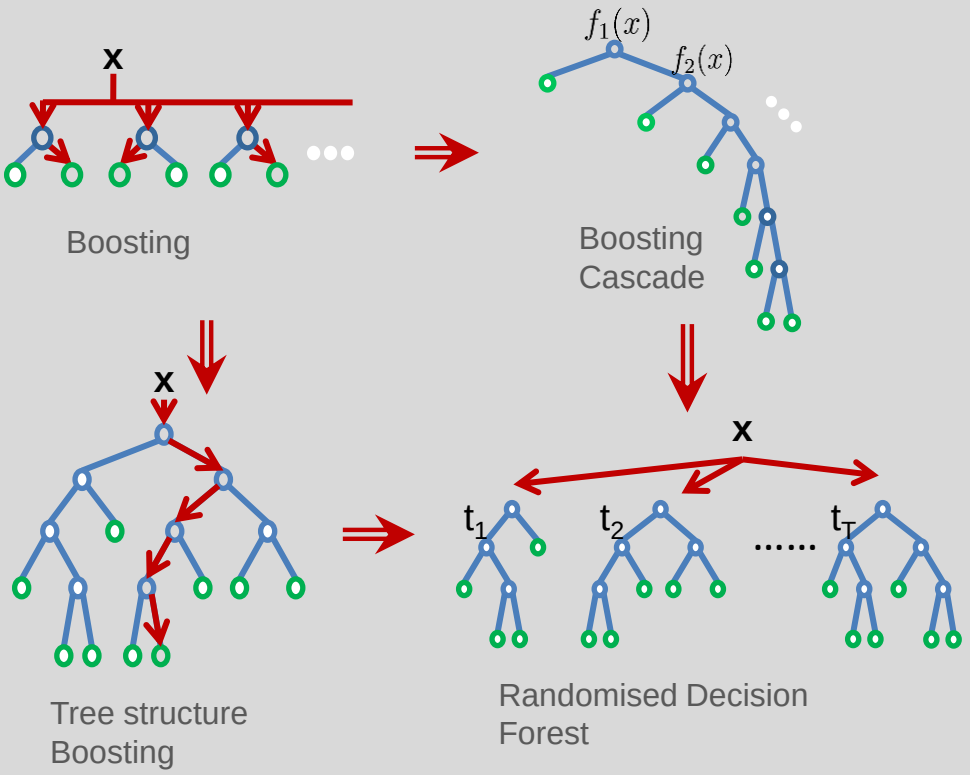


Scene Segmentation, IJCV12

http://www.iis.ee.ic.ac.uk/ComputerVision/Publications

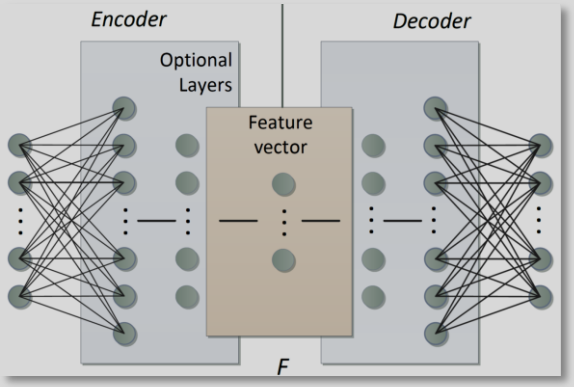
Boosting Classifiers and Decision Forests

- Boosting can be seen in a flat structure.
- Boosting cascade designed to accelerate run-time is a highly imbalanced tree.
- Tree-structured boosting classifiers have been studied to tackle multi-class problems.

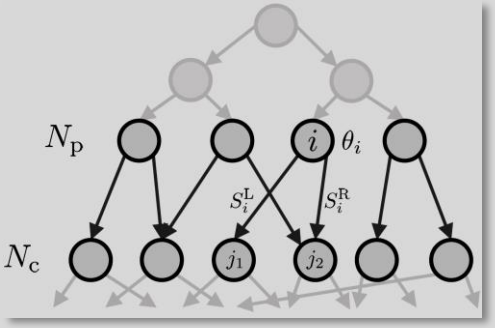


Deep Convolutional Neural Networks and Decision Forests

- Connectivity (memory)
  - CNN is fully connected.
  - Memory used in decision trees grows exponentially with depth.
  - Decision Jungle (ensembles of directed acyclic graphs) allows multiple paths.

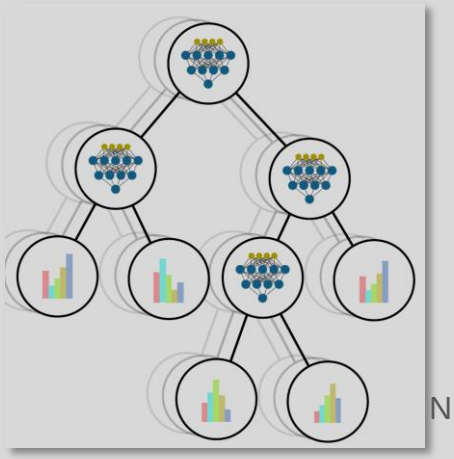


Sparse Autoencoders



Decision Jungle (Shotton et al. NIPS13)

- Data representation
  - DF requires a set of features manually defined.
  - Neural Decision Forests jointly tackles data representation and discriminative learning, using randomized Multi-Layer Perceptrons as split nodes.
  - Hough Networks (cf. Hough Forest) (Riegler et al. BMVC14) jointly perform classification and regression.



NDF (Bulo et al. CVPR14)

Resource

- ICCV09 Tutorial on Boosting and Random Forest (by T-K Kim et al) http://www.iis.ee.ic.ac.uk/icvl/iccv09_tutorial.html
- MVA13 Tutorial on Randomised Forests and Tree-structured Algorithms (by T-K Kim) http://www.iis.ee.ic.ac.uk/icvl/res_interest.htm

- ICCV11 Tutorial on Decision Forests (by Criminisi et al) http://research.microsoft.com/en-us/projects/decisionforests/
- ICCV13 Tutorial on Decision Forests and Field (by Nowozin et al) http://research.microsoft.com/en-us/um/cambridge/projects/iccv2013tutorial/

- 2001, Breiman L, Random Forests. Machine Learning, 45 (1), pp 5-32.

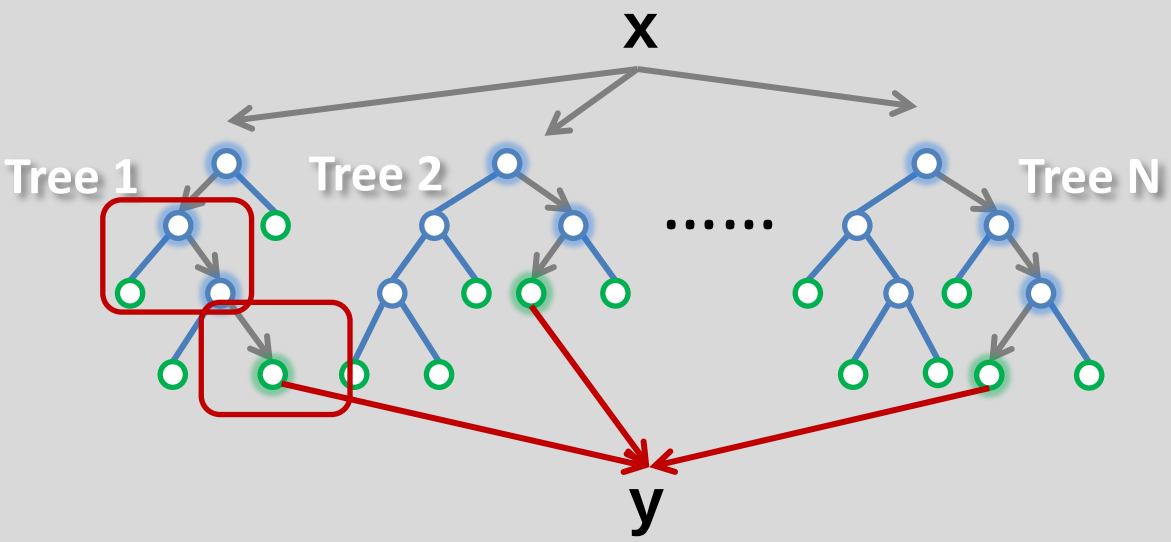
Resource

- For Random Forest Matlab codes
  - (i) P. Dollar's toolbox http://vision.ucsd.edu/~pdollar/toolbox/doc/
  - (ii) http://code.google.com/p/randomforest-matlab/
  - (iii) http://www.mathworks.co.uk/matlabcentral/fileexchange/31036-random-forest
  - (iv) Karpathy's toolbox https://github.com/karpathy/Random-Forest-Matlab.

- For open-source visualisation tool for machine learning algorithms, including RF
  - MLDEMOS http://mldemos.epfl.ch/

Randomised Decision Forests (Basics)

: ensemble of bagged decision tree learners with randomized feature selection

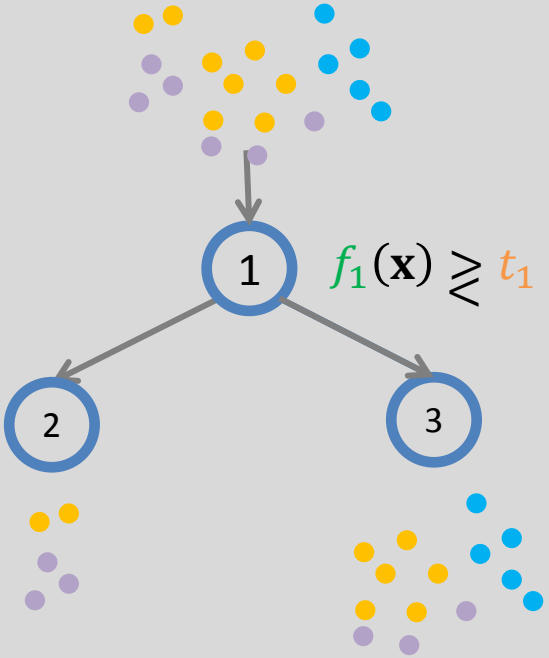


Learning Binary Decision Tree

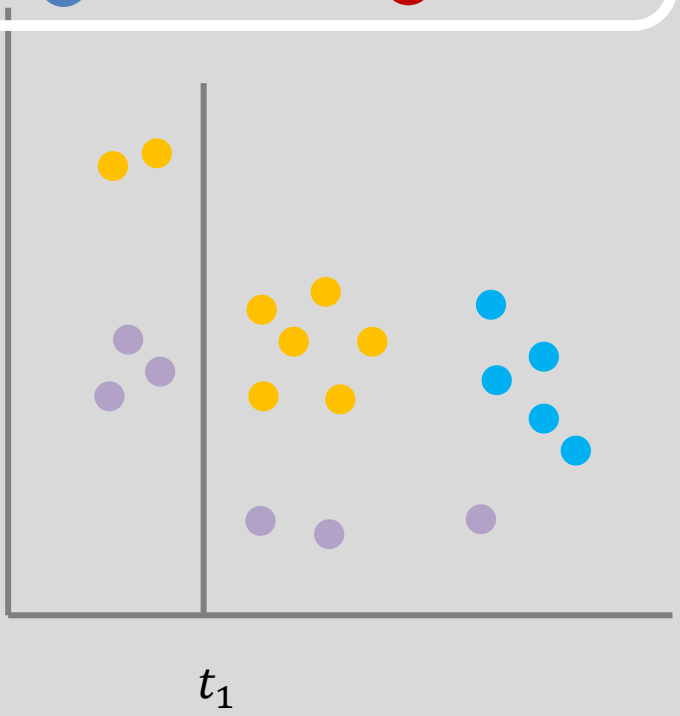
- recursive partitioning

- Given data pairs of $\mathbf{x}_1, \ldots, \mathbf{x}_N \quad l_1, \ldots, l_N$,

tree grows by successively partitioning the input data or space.



Node n Split function f_n Threshold t_n
Class prob dist P_n
Split nodes Leaf nodes

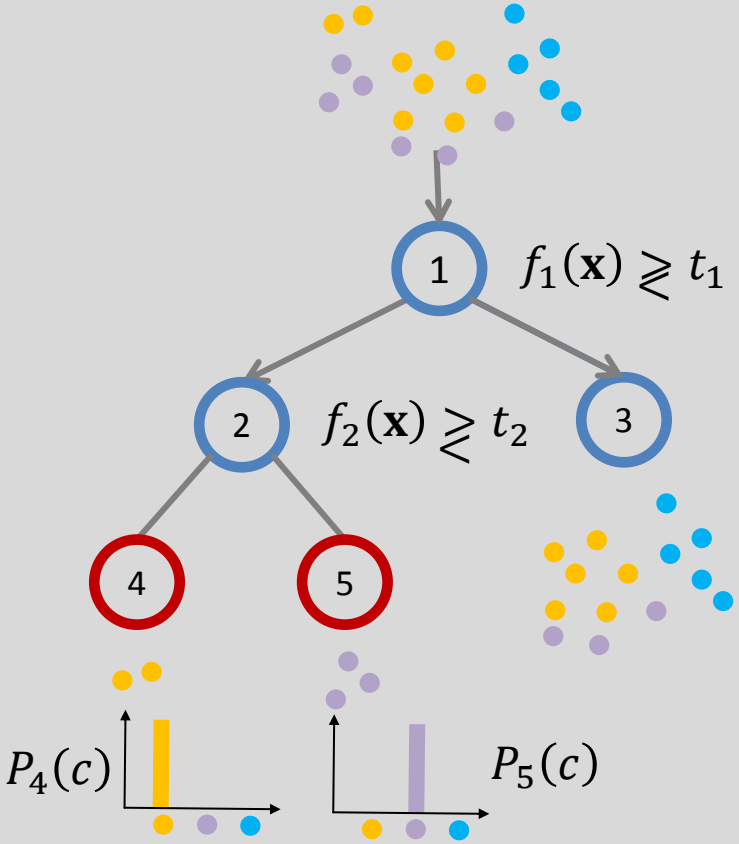


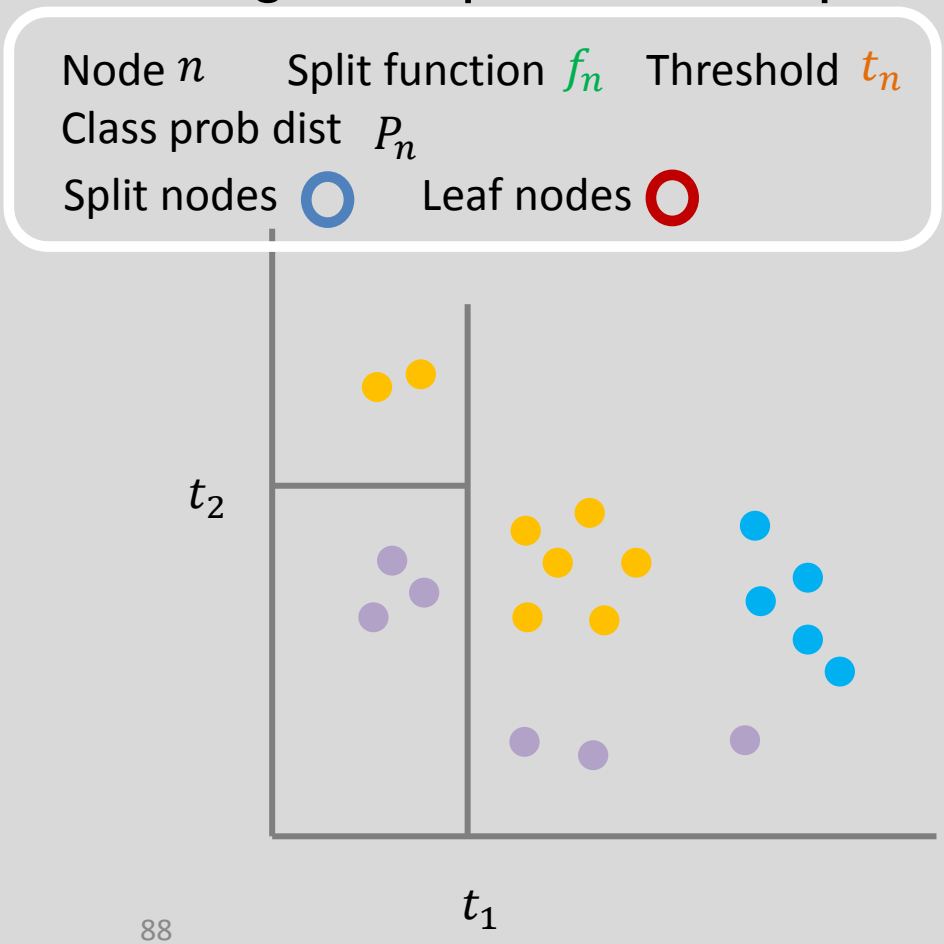
Learning Binary Decision Tree

- recursive partitioning

- Given data pairs of $\mathbf{x}_1, \ldots, \mathbf{x}_N \quad l_1, \ldots, l_N$,

tree grows by successively partitioning the input data or space.



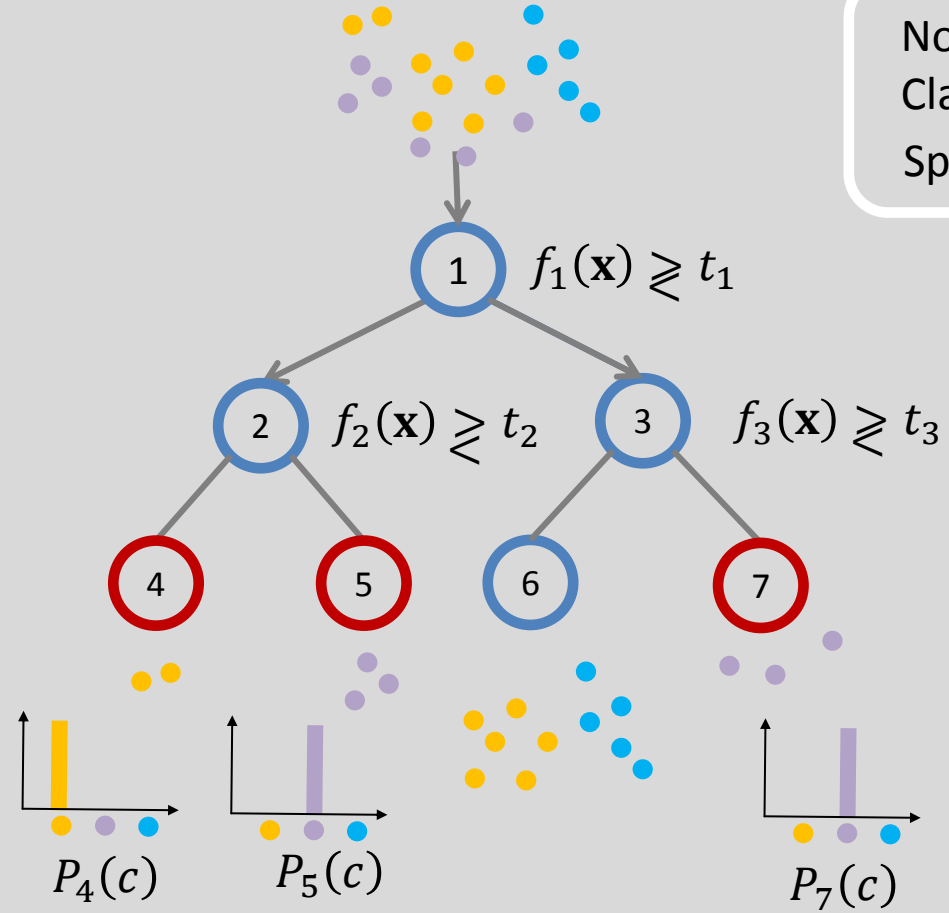


Learning Binary Decision Tree

- recursive partitioning

- Given data pairs of $\mathbf{x}_1, \ldots, \mathbf{x}_N \quad l_1, \ldots, l_N$,

tree grows by successively partitioning the input data or space.

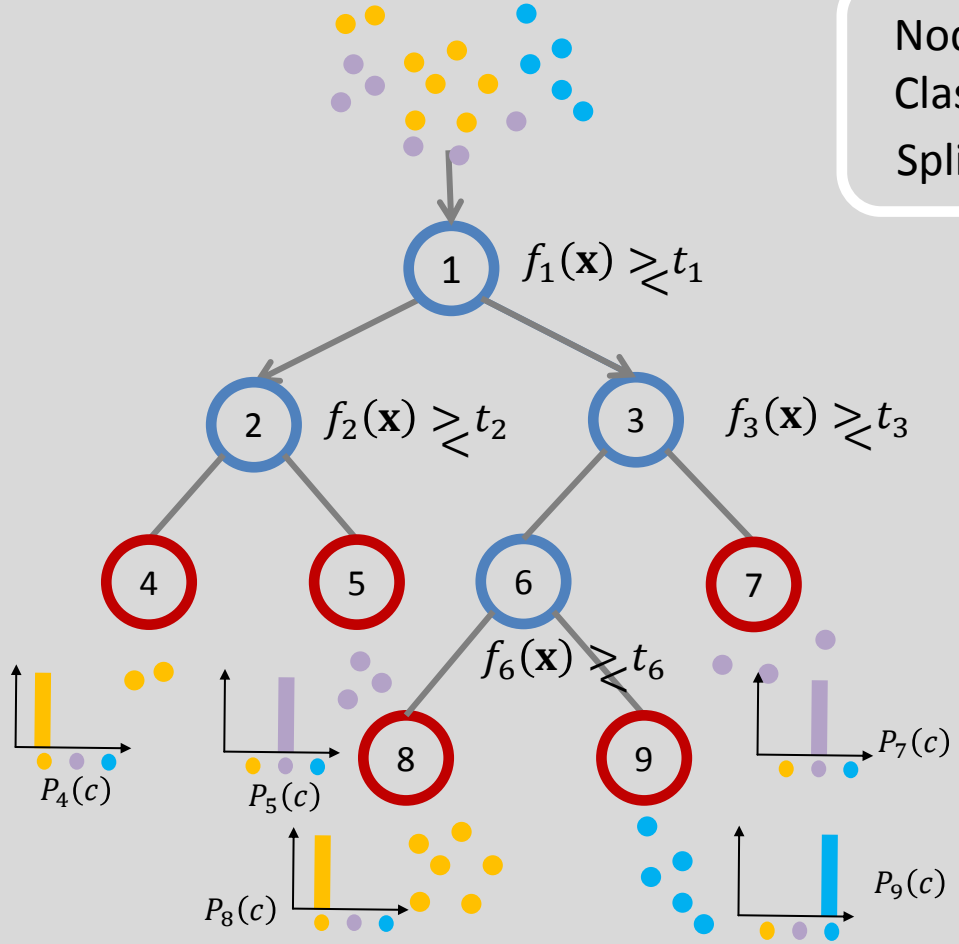


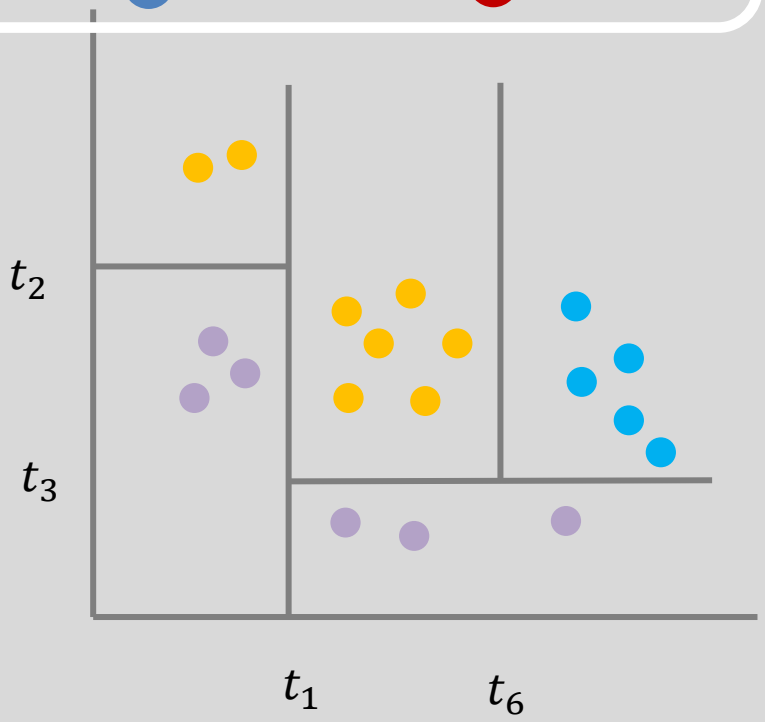
Learning Binary Decision Tree

- recursive partitioning

- Given data pairs of $\mathbf{x}_1, \ldots, \mathbf{x}_N \quad l_1, \ldots, l_N$,

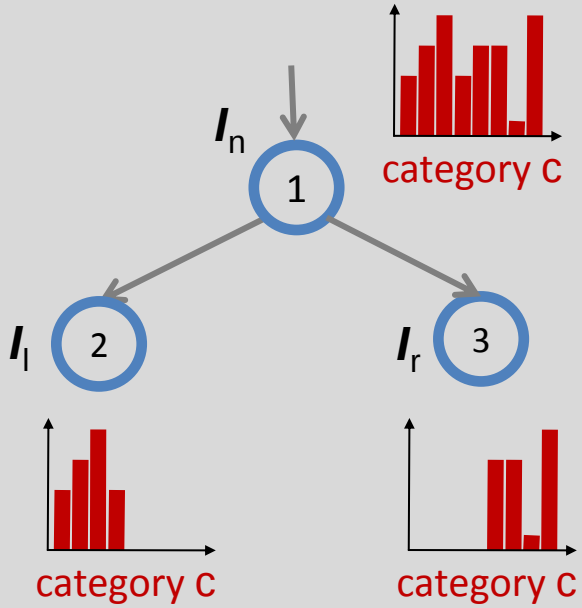
tree grows by successively partitioning the input data or space.





Recursive Node Splitting

- Split training data I_n that reach node n.
  - Left split $I_l = \{\mathbf{x} \in I_n \mid f_n(\mathbf{x}) < t_n\}$
  - Right split $I_r = I_n \backslash I_l$



- Try all pairs of (f_n,t_n) and choose the best (f_n,t_n) that maximise the information gain

$$\Delta H = -\frac{|I_l|}{|I_n|} H\left(P_{I_l}(c)\right) - \frac{|I_r|}{|I_n|} H\left(P_{I_r}(c)\right)$$

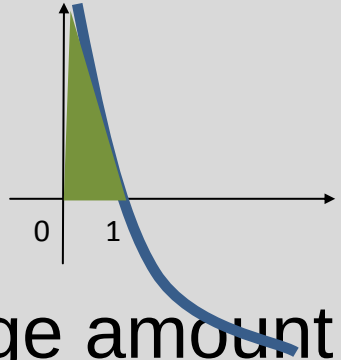
Entropy

- If x and y are unrelated, we define

$$h(x,y) = h(x) + h(y)$$

- As $P(x,y) = P(x)P(y)$, we define

$$h(x) = -logP(x)$$

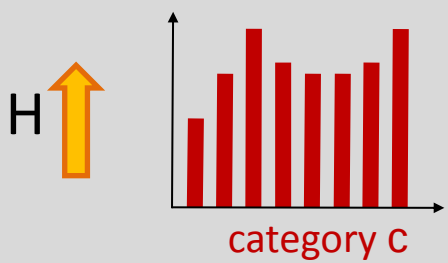


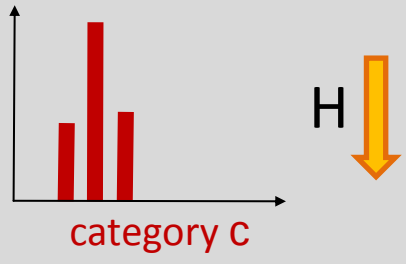
- Average amount of info. is

$$\mathrm{H}(x) = \sum_x P(x)h(x)$$

$$= -\sum_x P(x)\log P(x)$$

- H: entropy (average amount of info.) is large when the distribution is more uniform.



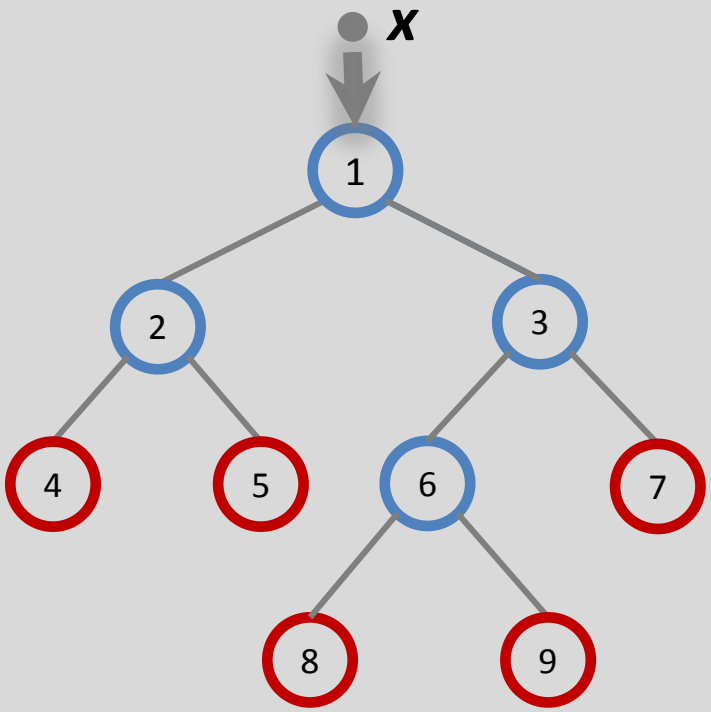


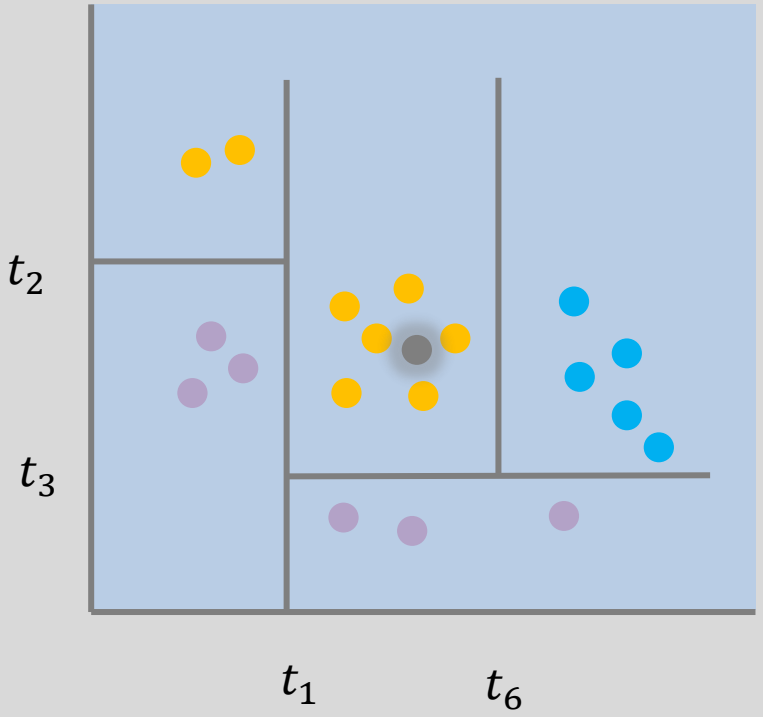
Stopping Criteria

- The tree growth stops when it meets a certain stopping criterion e.g.
  - Minimum number of data points,
  - Maximum tree depth, or
  - When the information gain is smaller than a predefined value.

Fast Evaluation of Decision Tree

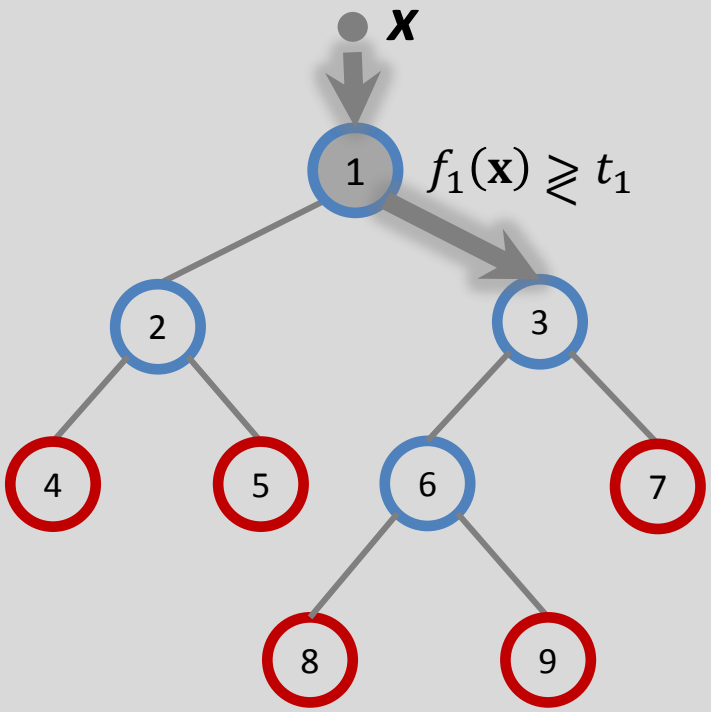
- A test input ***x*** is given by a sequential decision-making on the traversal of a binary tree.

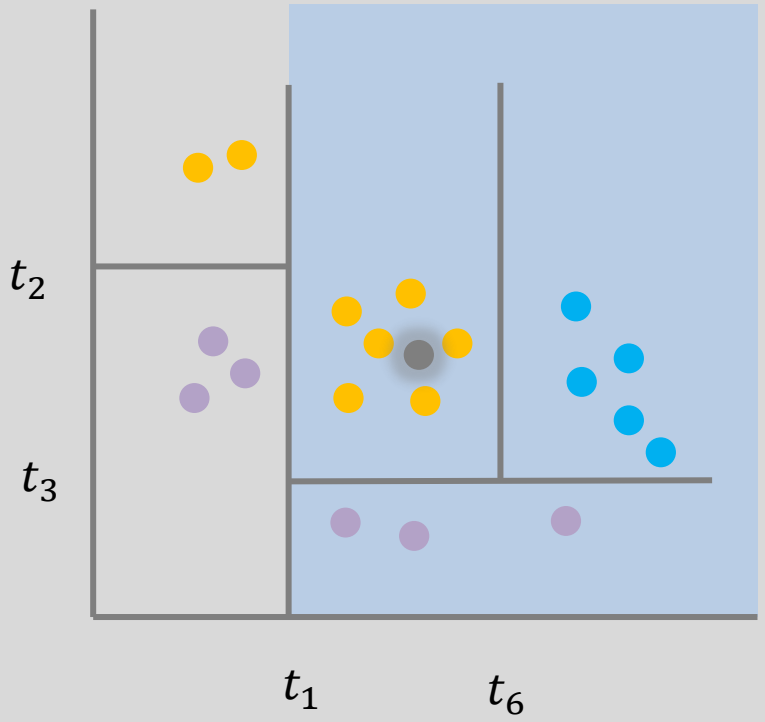




Fast Evaluation of Decision Tree

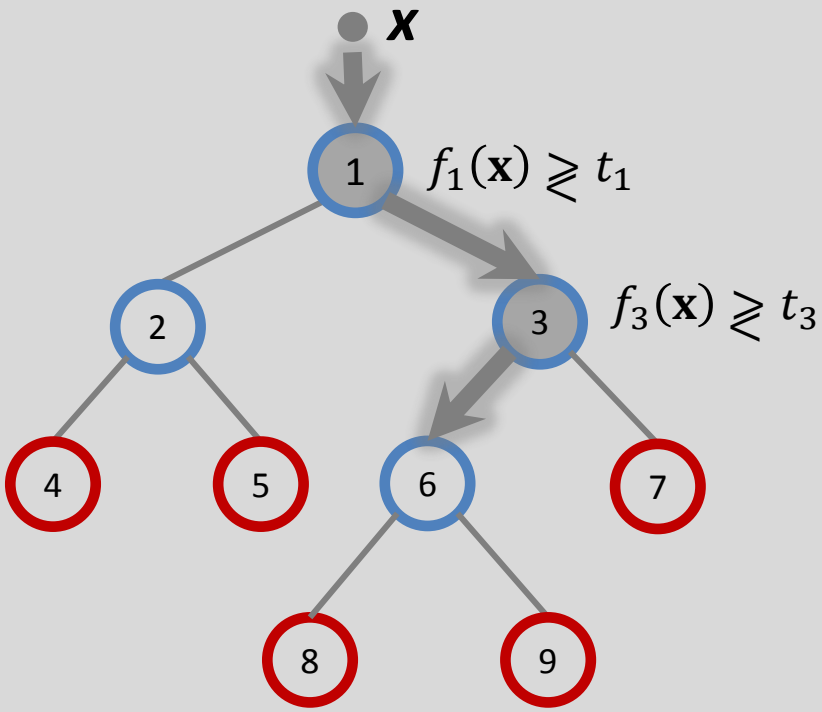
- A test input ***x*** is given by a sequential decision-making on the traversal of a binary tree.

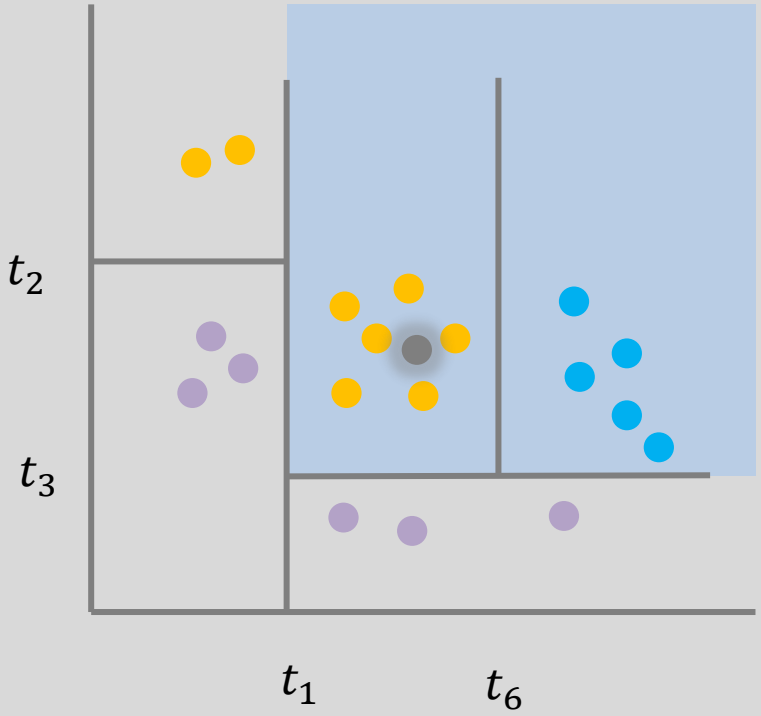




Fast Evaluation of Decision Tree

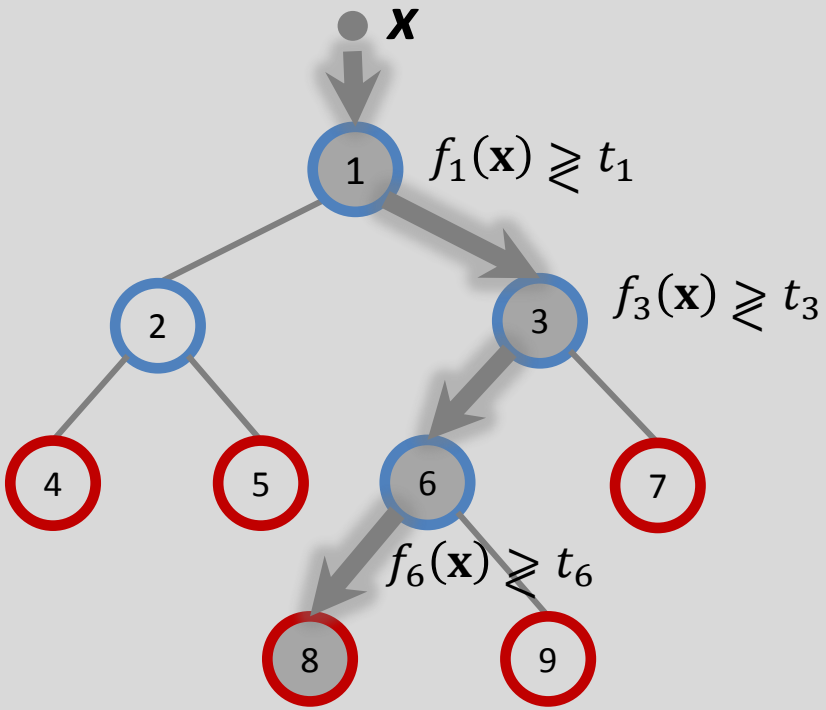
- A test input ***x*** is given by a sequential decision-making on the traversal of a binary tree.

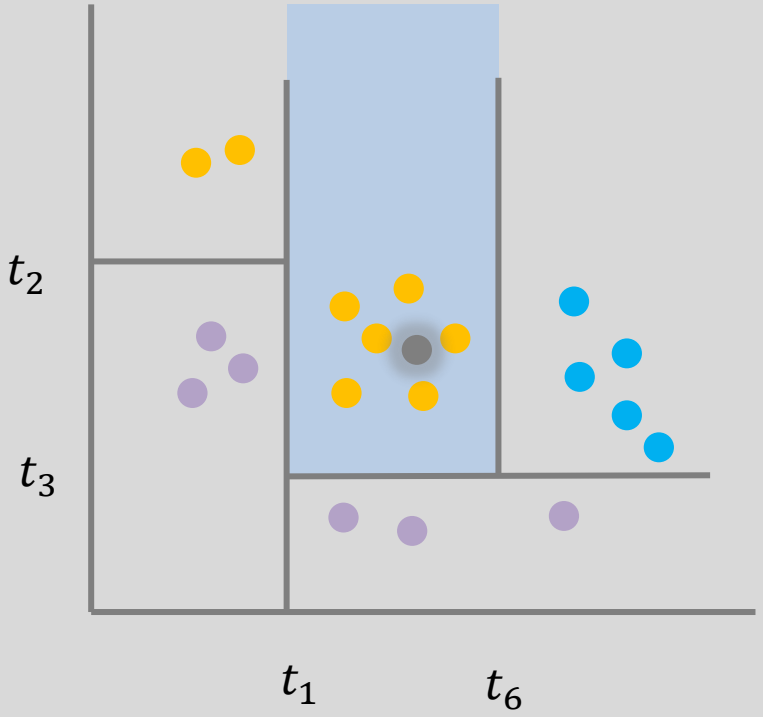




Fast Evaluation of Decision Tree

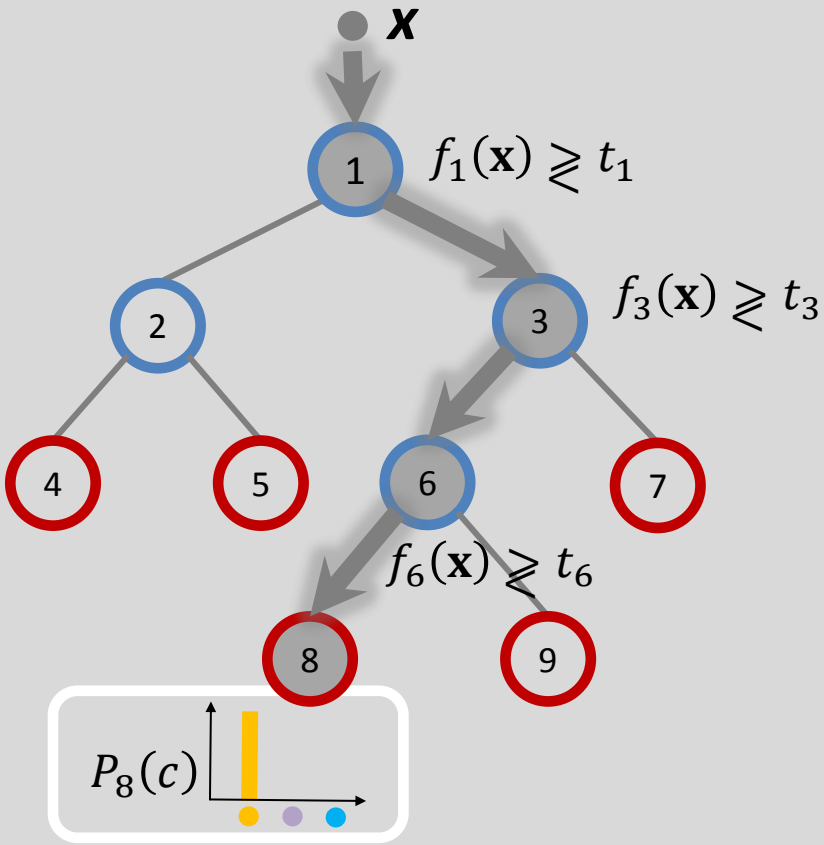
- A test input ***x*** is given by a sequential decision-making on the traversal of a binary tree.

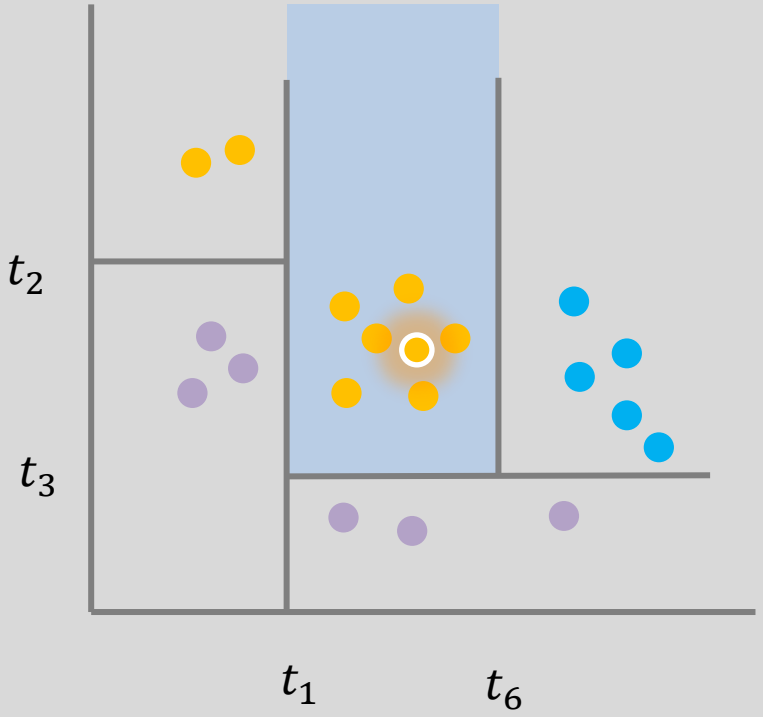




Fast Evaluation of Decision Tree

- A test input ***x*** is given by a sequential decision-making on the traversal of a binary tree.





Summary on Decision Tree

Pros

- It provides very fast evaluation.
- It is a highly scalable algorithm for training.

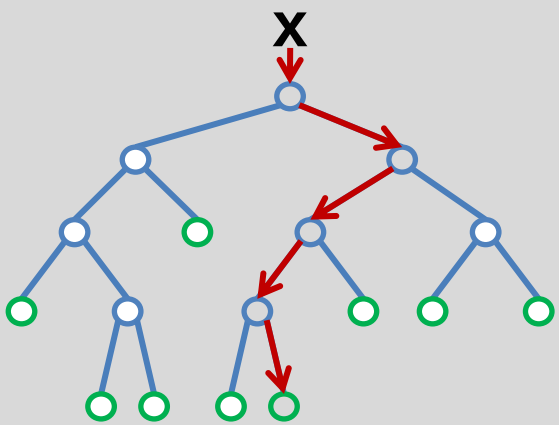
Cons

- It severely overfits to the training data (pruning is hard).

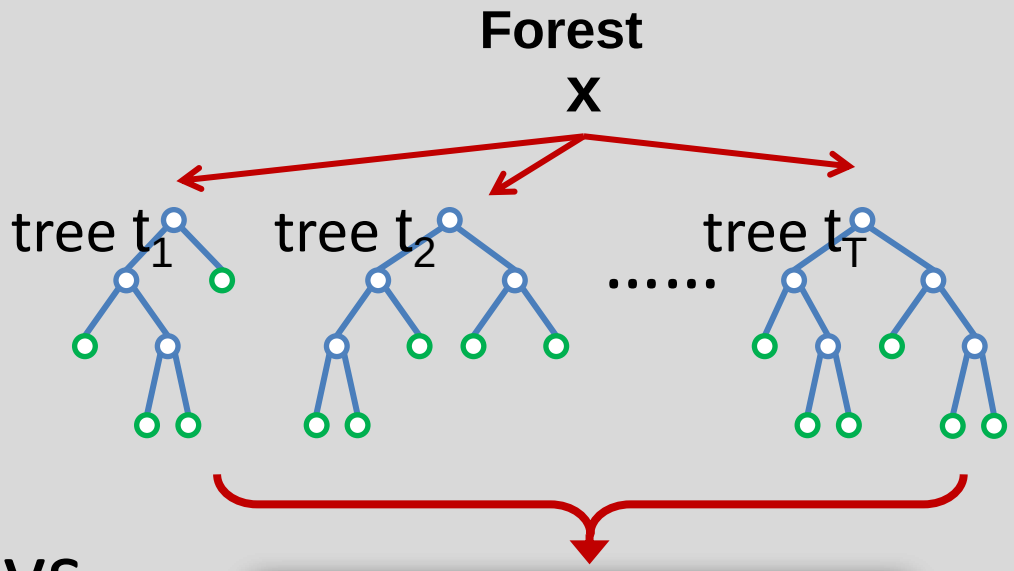
Imperial College London

Overfitting

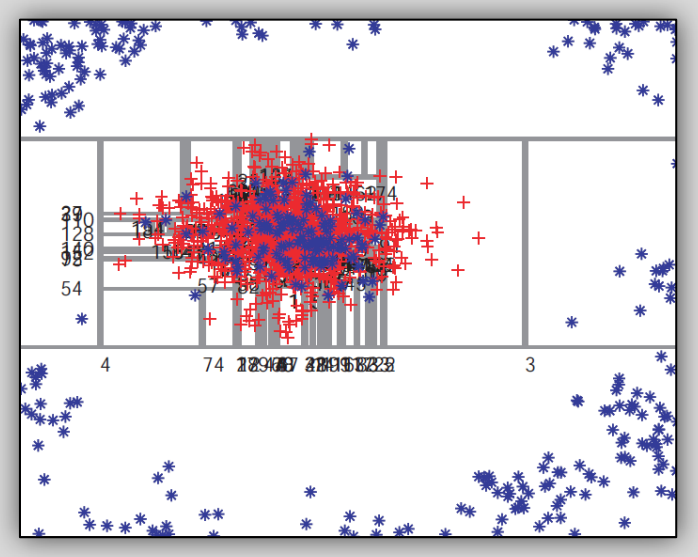
Decision Tree



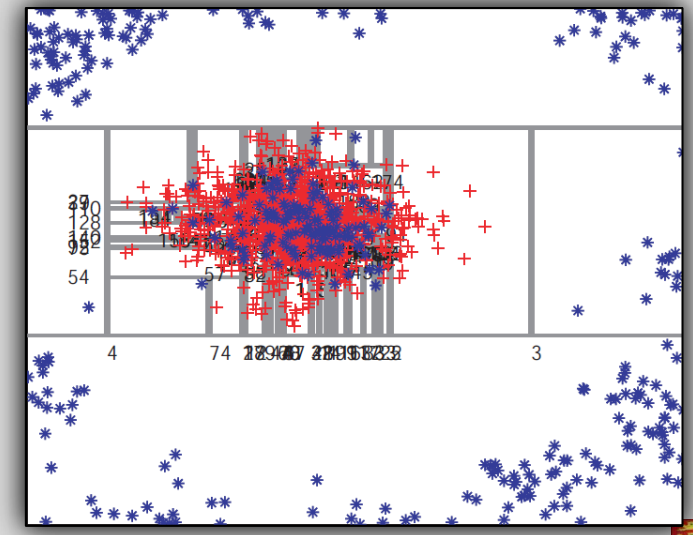
Forest



VS



Overfit

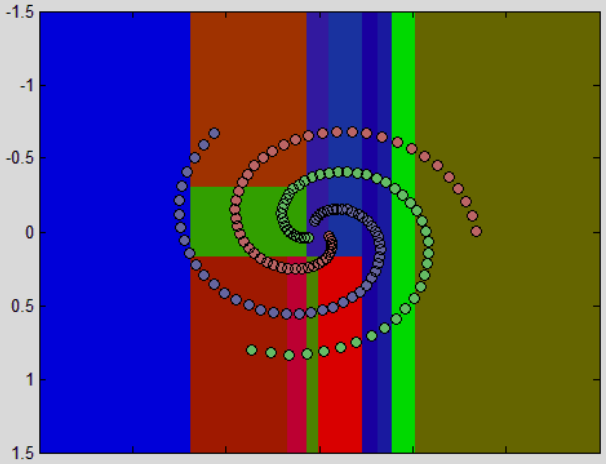


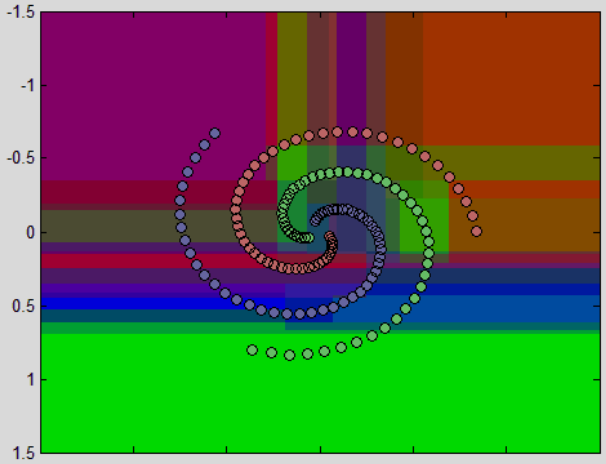
Generalised (smooth)

Smoothing Effect of RF on Toy Data

- For the spiral data, we fix the max tree depth as 5, the number of split trials as 3, then change the number of trees N in RF.

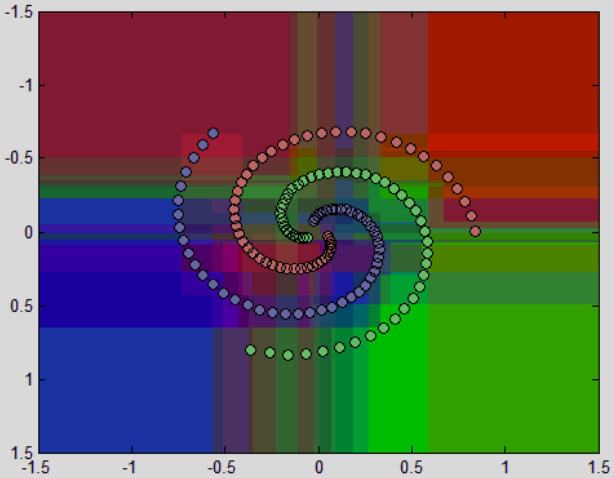
N=1

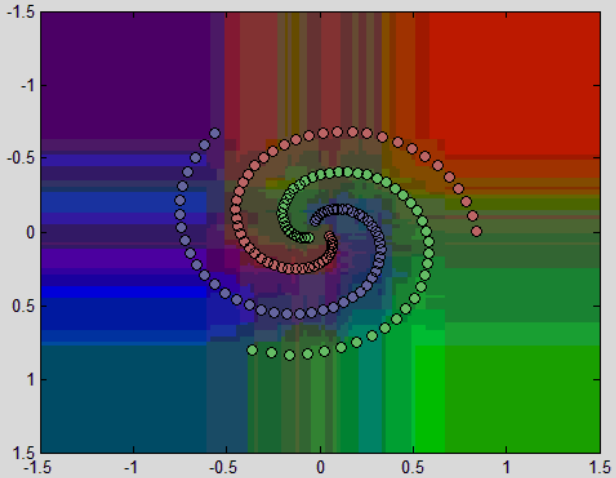




N=3

N=5

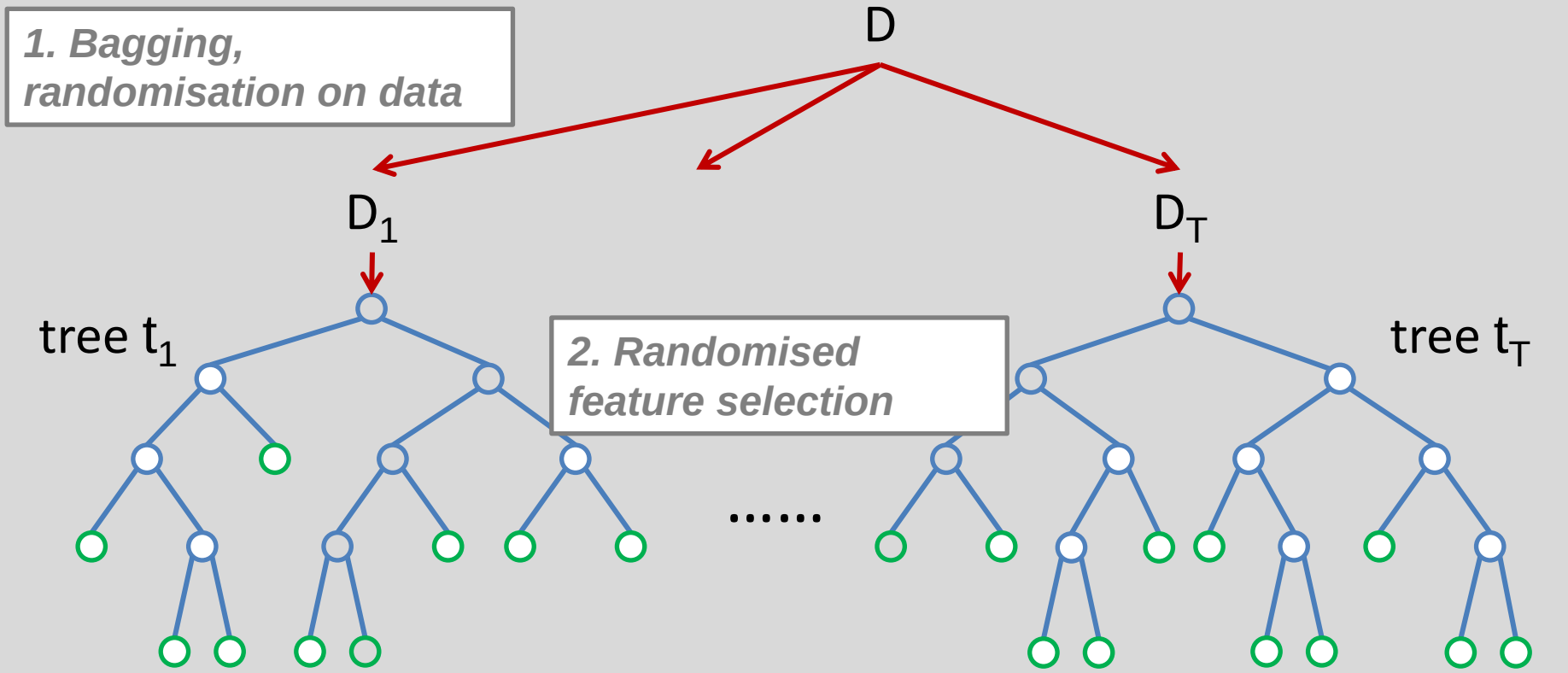




N=10

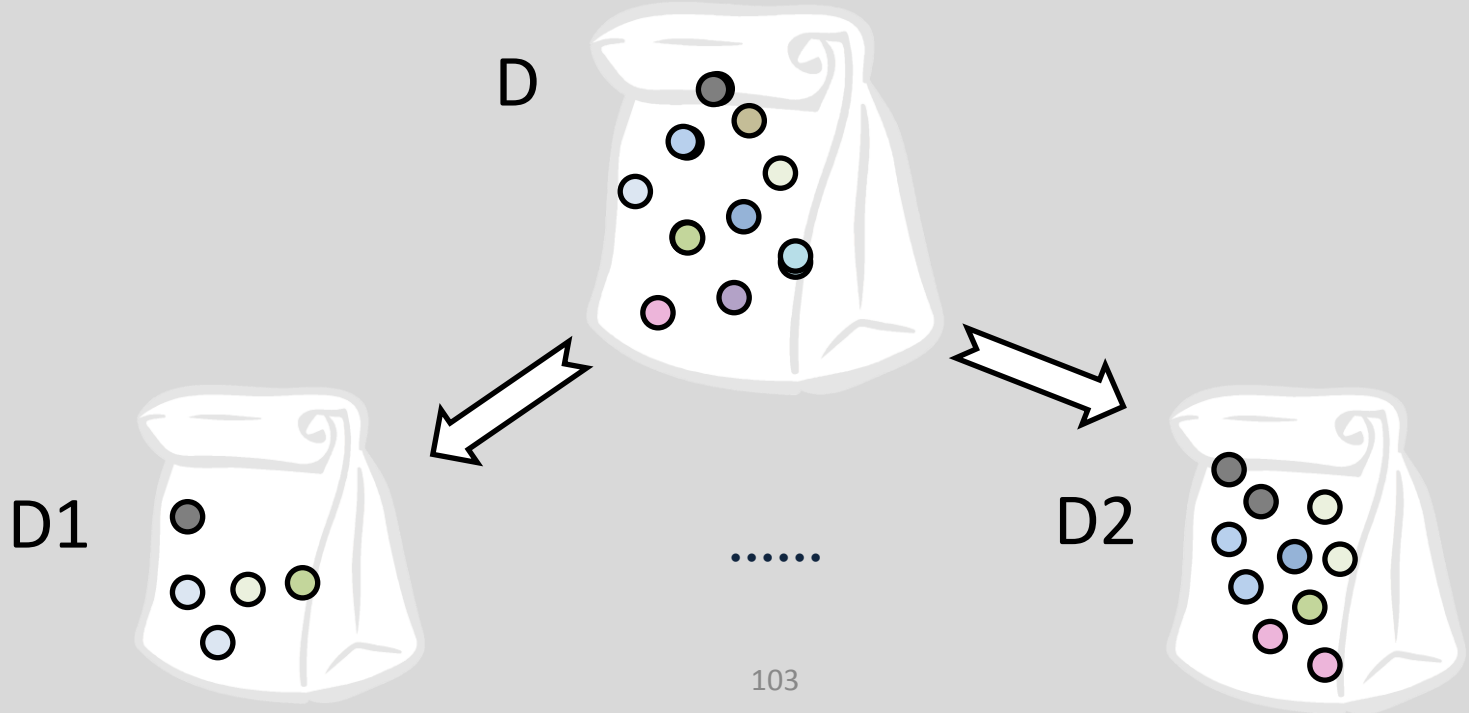
Learning Random Forests

- Forest is an ensemble of bagged decision trees with randomized feature selection.



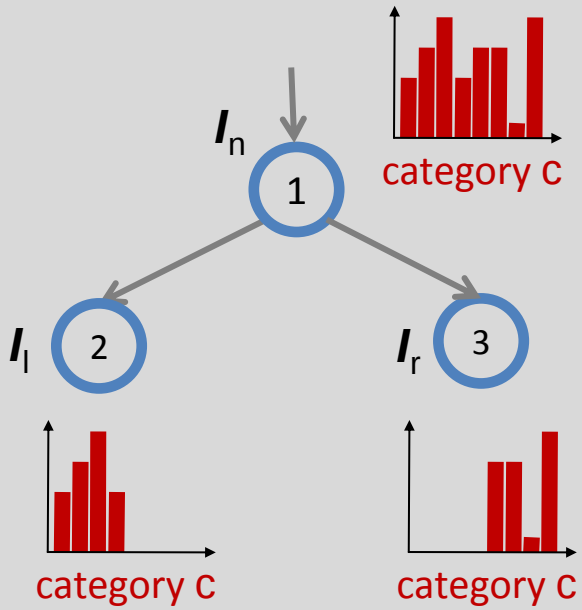
Bagging (Bootstrap AGGregatING)

- Given a data set D of size n, it generates T data subsets D_i .
- Each subset has n_i=n, by sampling data from D uniformly and with replacement.
- Some data are repeated in D_i . If n_i=n and n is large, D_i is likely to have 63.2% of unique data.



Randomised Feature Selection

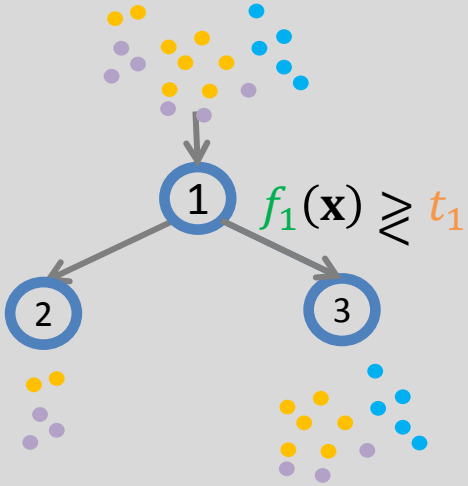
- Split training data I_n that reach node n.
  - Left split $I_l = \{\mathbf{x} \in I_n \mid f_n(\mathbf{x}) < t_n\}$
  - Right split $I_r = I_n \backslash I_l$
- ***Features f and thresholds t are chosen at random.***
- Choose the best (f_n,t_n) that maximise the information gain



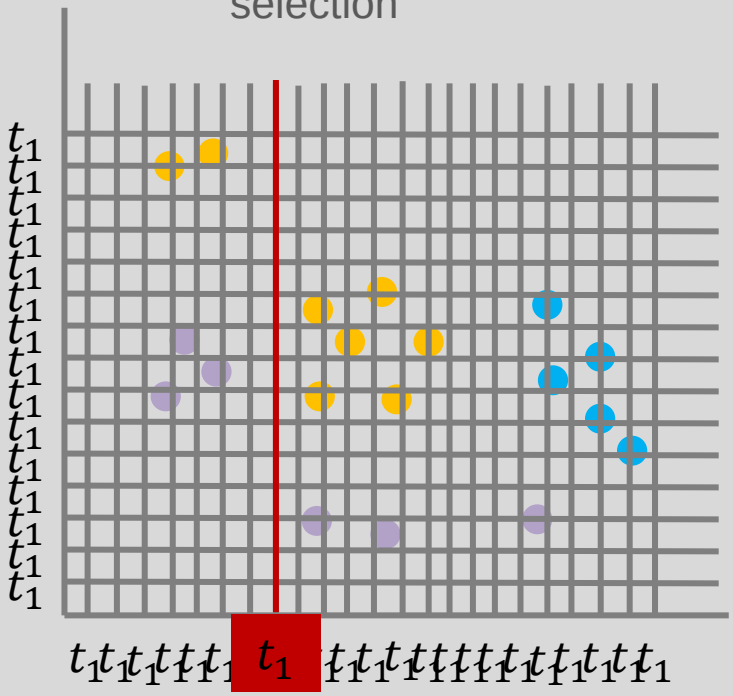
$$\Delta H = -\frac{|I_l|}{|I_n|} H\big(P_{I_l}(c)\big) - \frac{|I_r|}{|I_n|} H\big(P_{I_r}(c)\big)$$

Randomised Feature Selection

Node n (axis aligned) Split function f_n Threshold t_n
Class prob dist P_n
Split nodes Leaf nodes



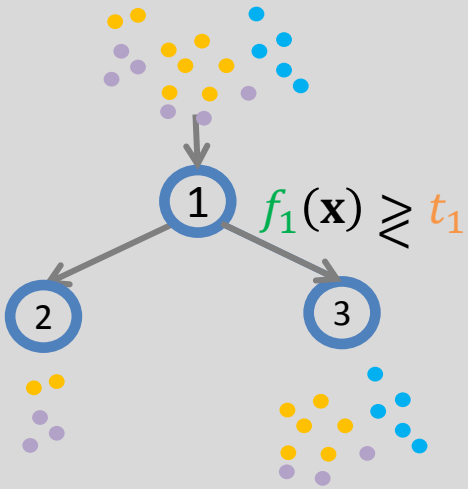
brute-forth feature selection



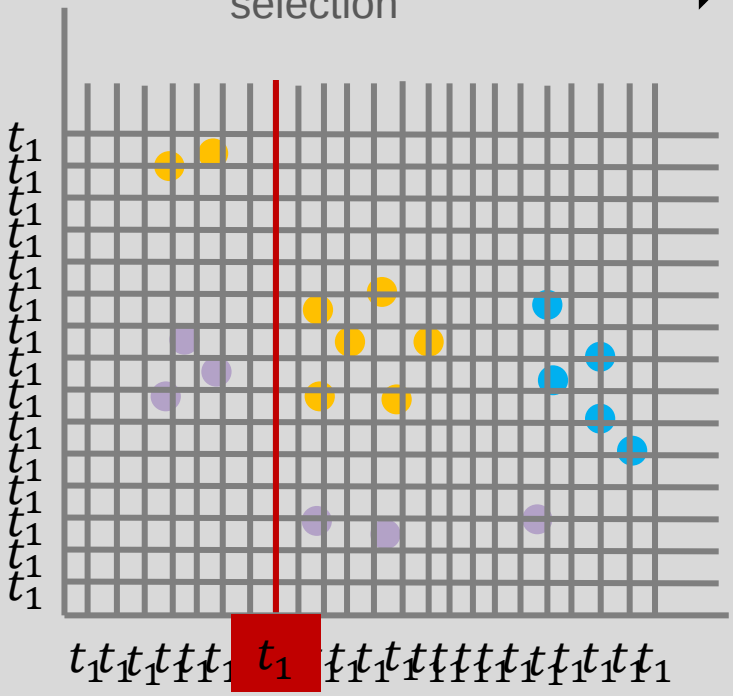
Imperial College London

Randomised Feature Selection

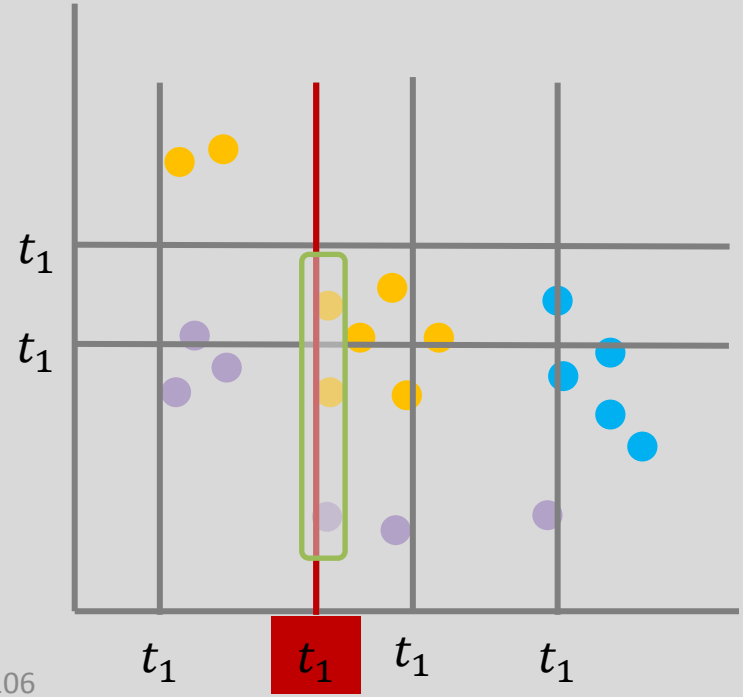
Node n (axis aligned) Split function f_n Threshold t_n
Class prob dist P_n
Split nodes Leaf nodes



brute-forth feature selection



random feature selection



Tree Correlation vs Strength

- Randomisation on data and feature increases diversity among trees.
- For the fixed depth, the randomised feature decreases strength of each tree.

Committee Machine

- Output of each model is $y_m(x) = h(x) + \epsilon_m(x)$

  where $h(x), \epsilon_m(x)$ are the true value and error of each model.

- The average sum-of-squares error is

$$\mathrm{E}[\{y_m(x) - h(x)\}^2] = \mathrm{E}[\epsilon_m(x)^2]$$

- The average error by acting individually is

$$\mathrm{E}_{av} = \frac{1}{M}\sum_{m=1}^{M} \mathrm{E}[\epsilon_m(x)^2]$$

Committee Machine

- The committee machine is

$$y_{com}(x) = \frac{1}{M}\sum_{m=1}^{M} y_m(x)$$

- The expected error of the committee machine is

$$E_{com} = E\left[\left\{\frac{1}{M}\sum_{m=1}^{M} y_m(x) - h(x)\right\}^2\right]$$

$$= E\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(x)\right\}^2\right] = E\left[\frac{1}{M^2}\left(\epsilon_1^2 + \epsilon_1\epsilon_2 + \epsilon_2^2 + \cdots\right)\right]$$

Committee Machine

- If we assume

$$E[\epsilon_m(x)\epsilon_l(x)] = 0, \qquad m \neq l$$

then we obtain

$$E_{com} = \frac{1}{M} E_{av}$$

- In practice, the errors are typically highly correlated, but we can expect that

$$E_{com} \leq E_{av}$$

See more in 2001, Breiman L, Random Forests. Machine Learning, 45 (1), pp 5-32.

Forest Parameters

- ***Number of trees:*** Accuracy grows with more trees and saturates at a certain point. More trees mean more memory and evaluation time.

- ***Depth of trees:*** the maximum tree depth < $\log_2 N$+1

  where *N* is the number of train data points.

- ***Stopping criterion:***

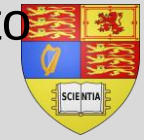
  minimum number of data points in each leaf node

  predefined maximum tree depth

  the information gain < threshold

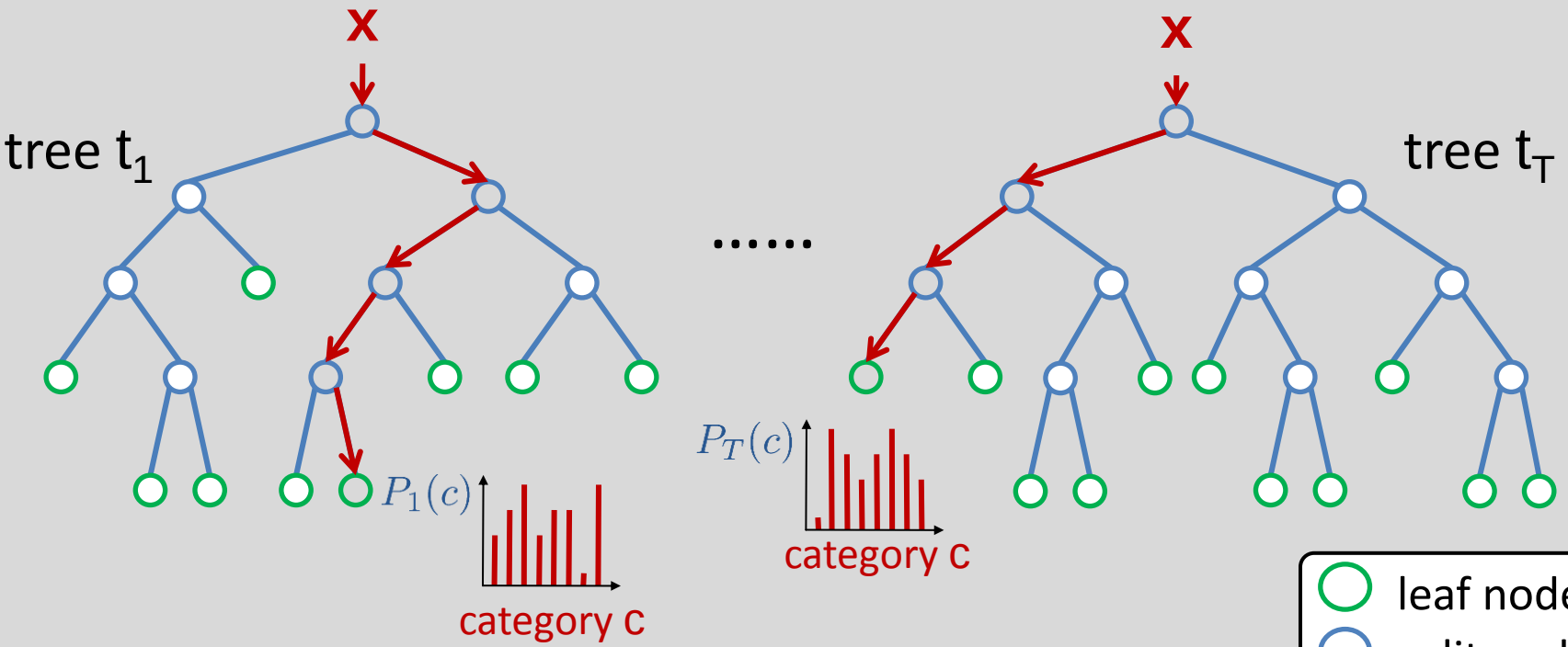
- ***Number of random features:*** not overly sensitive to this parameter. d' << d, where d is the feature dimension.

- ***Split criterion:*** Shannon entropy, Gini index, etc, depending on tasks to solve (**see Applications**).



Forest of Trees : Evaluation

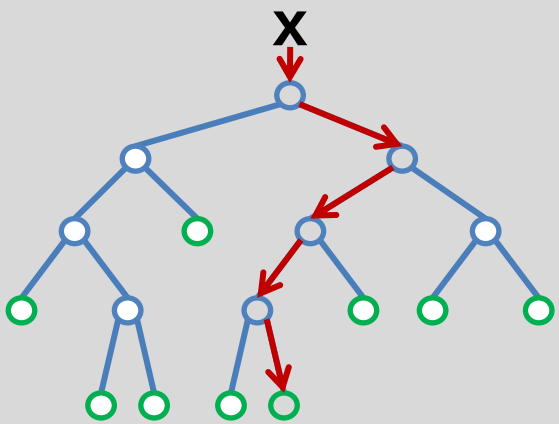
- A data point is passed down all trees, and the leaf nodes that the data point reaches are collected.

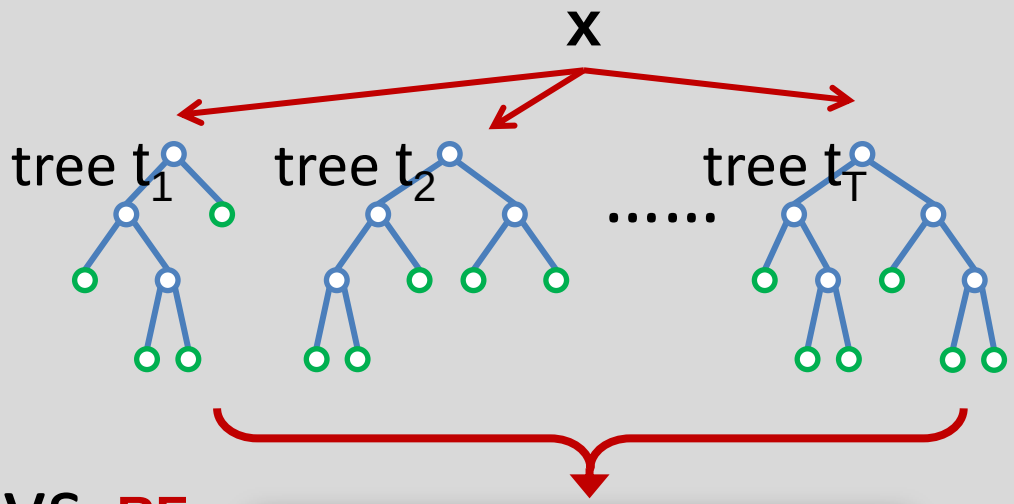


  - Classification is done by

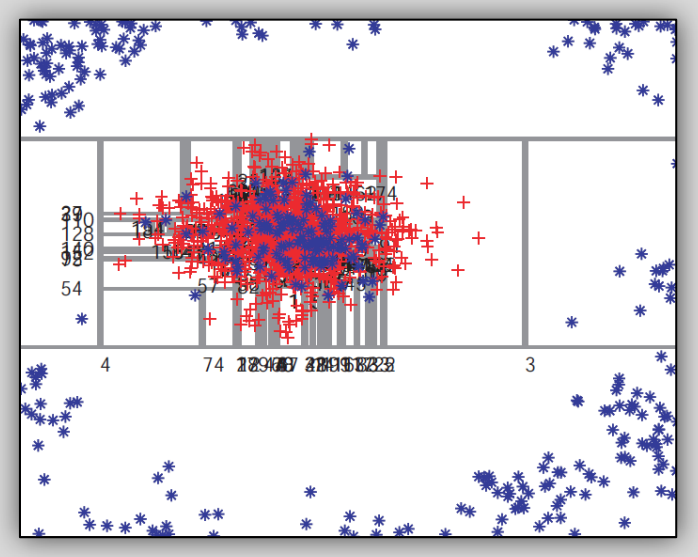
$$P(c|\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{x})$$

Decision Tree vs Random Forest

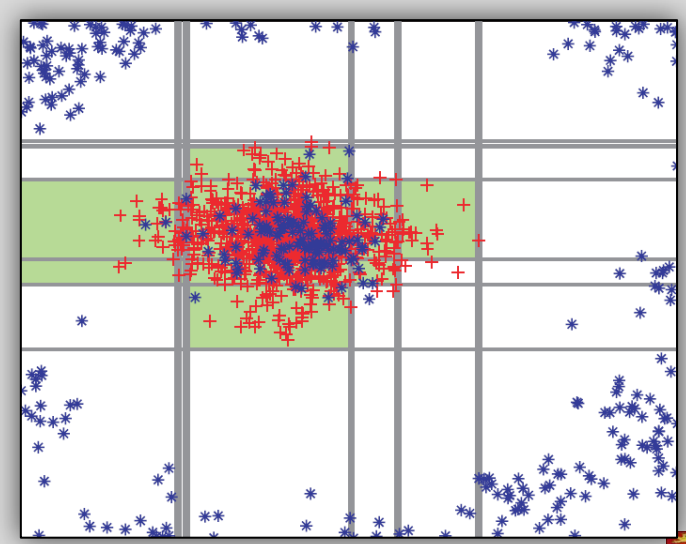




DT VS RF



Overfit

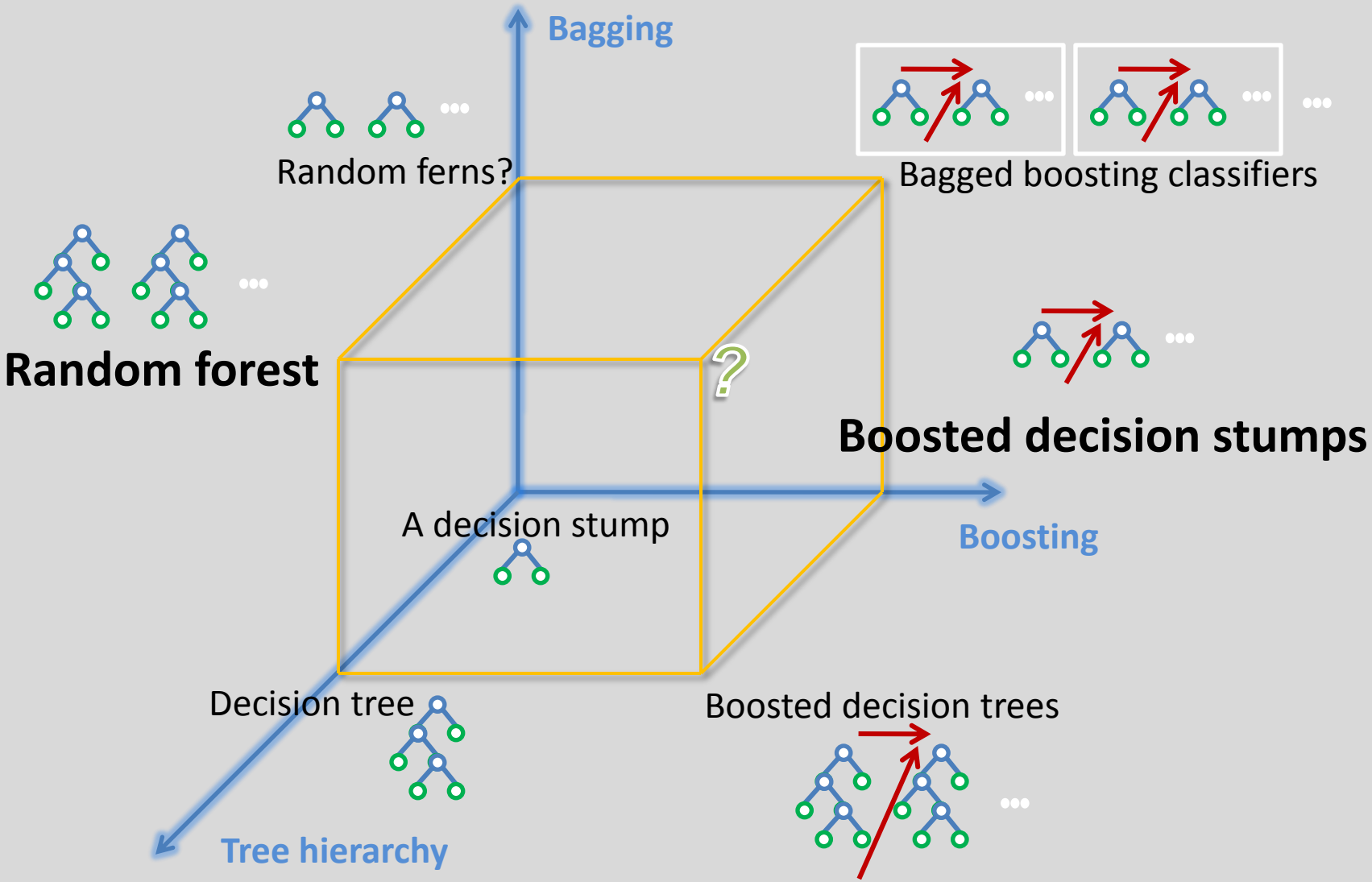


Reasonably Smooth

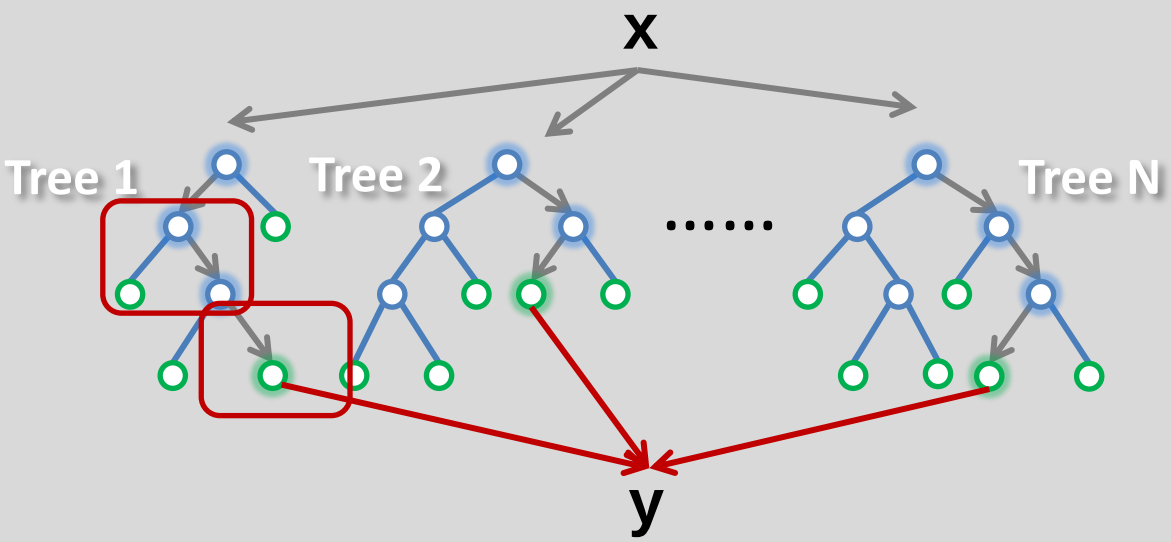
Summary

	Pros	Cons
Random Forest	• Generalization through random samples/features • Extremely fast classification • Highly scalable in training • Inherently multi-classes	• Inconsistency • Difficulty for adaptation
Boosting Decision Stumps	• Generalisation by a flat structure • Fast classification • Optimisation framework	• Slower than RF • Slow training

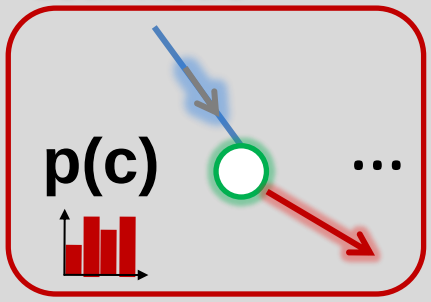
See more in Yin, Crinimisi, CVPR07, and Belle et al, ICPR08


Bagging
Random ferns?
Bagged boosting classifiers
Random forest
?
Boosted decision stumps
A decision stump
Boosting
Decision tree
Boosted decision trees
Tree hierarchy

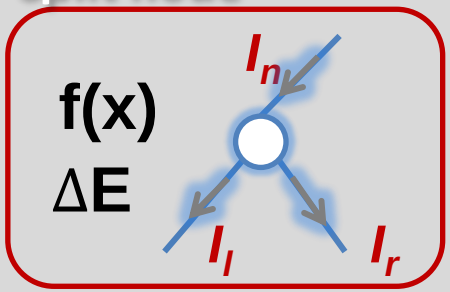
Part II. Decision Forests - Case Studies



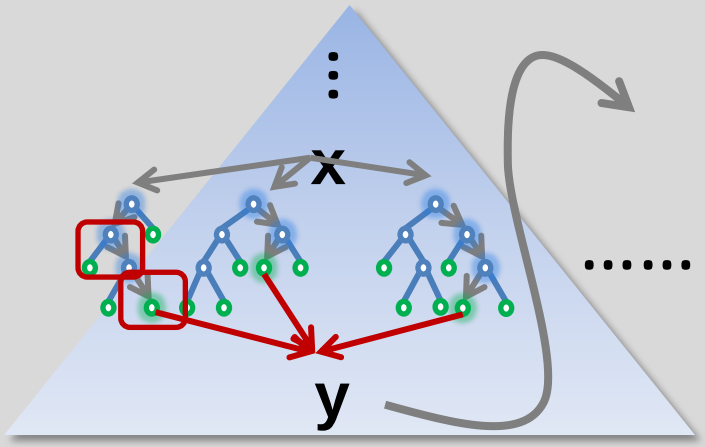
Leaf node



Split node

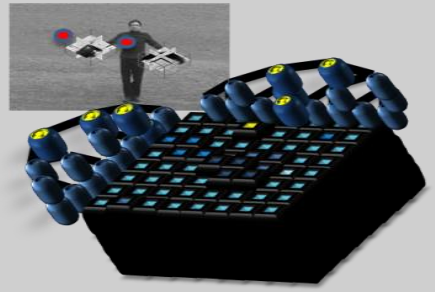


Architecture

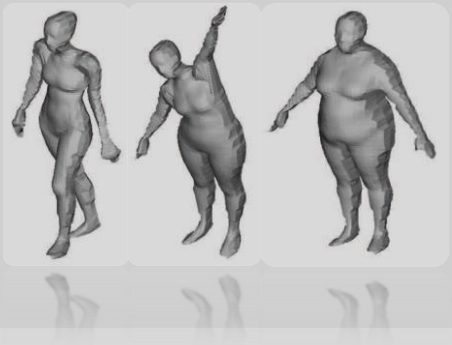


Decision Forests @ ICL

Leaf node

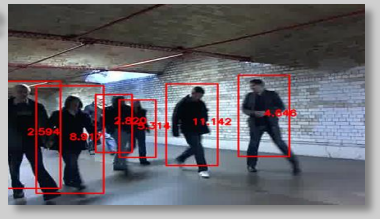


Structural + Semantic info. BMVC10

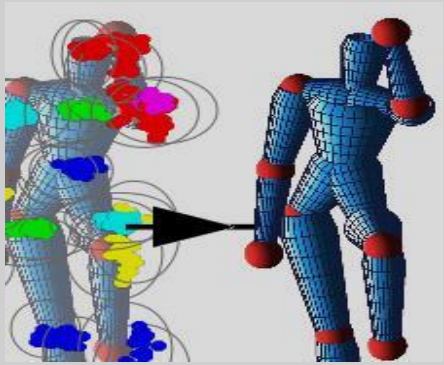


Probabilistic outputs into GPLVM ECCV 10, ICCV 11

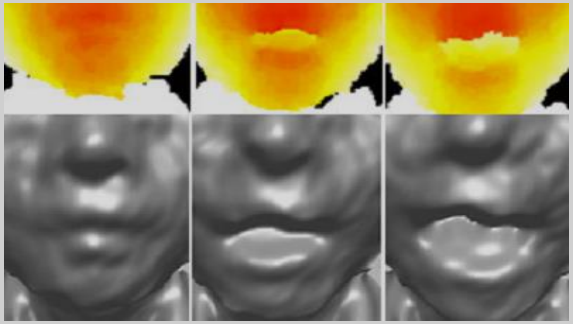
Split node



DOT matching split functions BMVC12



Multiple split criteria / adaptive weighting CVPR13

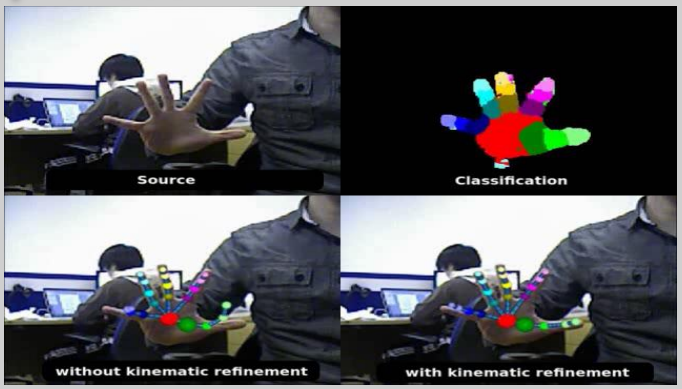


Manifold learning ICCV13

Imperial College London

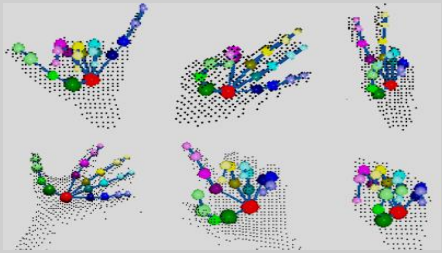
Decision Forests @ ICL

Split node

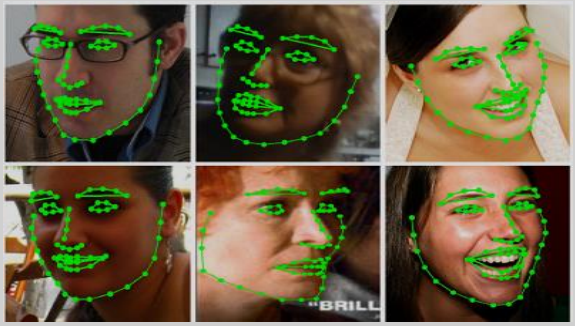


Semi-supervised/ transductive ICCV13 (oral)

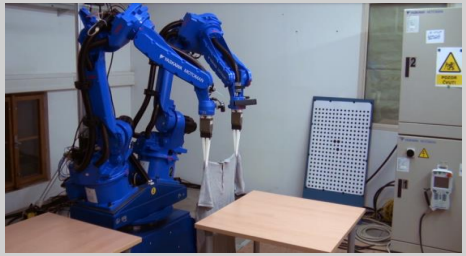
Architecture



Deep architecture by Latent Tree Model CVPR14 (oral)



Iterative multi-output CVPR14

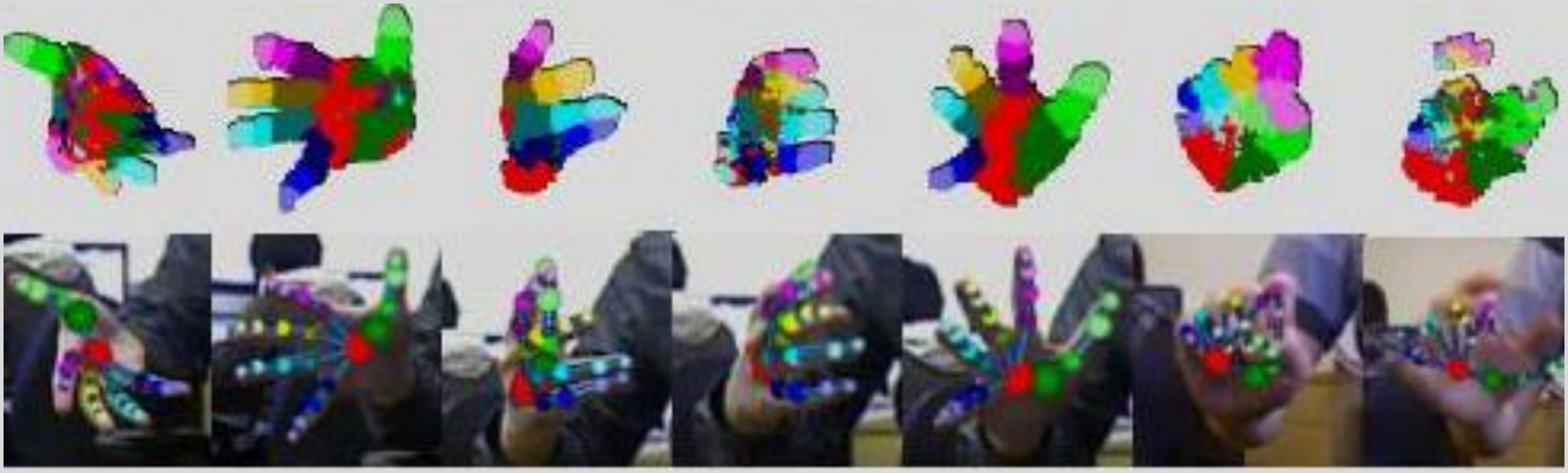


Active Forest ECCV14

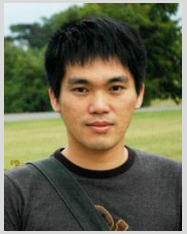
Latent Hough ECCV14

Split Criteria:
Semi-supervised Transductive Regression Forests

Imperial College London



Real-time Articulated Hand Pose Estimation using Semi-supervised Transductive Regression Forests



Danhang Tang

Imperial College London

Tsz-Ho Yu

T-K Kim

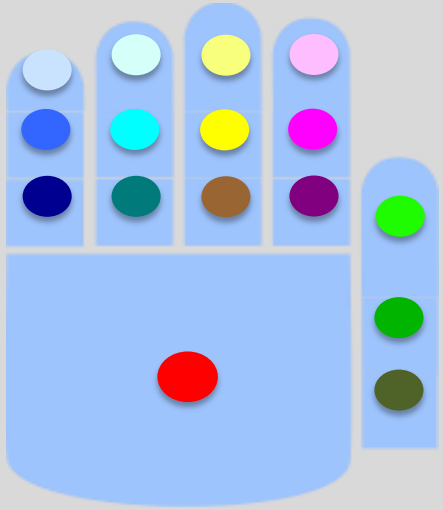
Imperial College London

ICCV 2013 (oral presentation)

Imperial College London

Problem Definition

Input: Given a segmented point cloud (or depth image)



Output: 3D location of 16 joints

$(x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_{16}, y_{16}, z_{16})$

Motivation

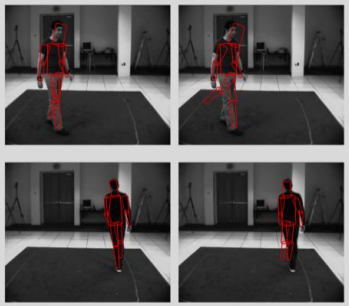
- Bare Hand-based Interaction
  - Natural User Interface (NUI) for Wearable AR

Microsoft – Hololense
http://www.microsoft.com/microsoft-hololens/en-us

Meta – SpaceGlasses
https://www.spaceglasses.com

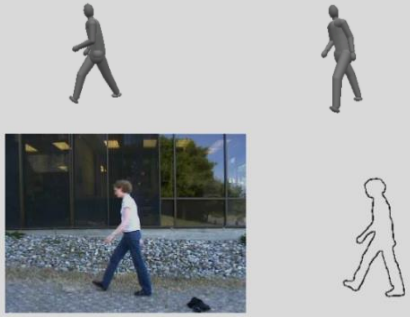
Imperial College London

Motivation



Multiple cameras with invserse kinematics
[Bissacco et al. CVPR2007]
[Yao et al. IJCV2012]
[Sigal IJCV2011]

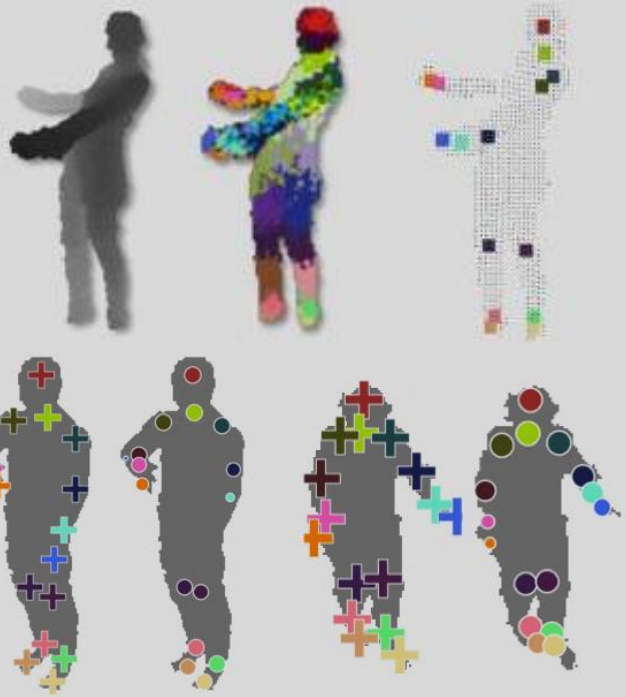
Specialized hardware
(e.g. structured light sensor, TOF camera)
[Shotton et al. CVPR'11]
[Baak et al. ICCV2011]
[Ye et al. CVPR2011]
[Sun et al. CVPR2012]

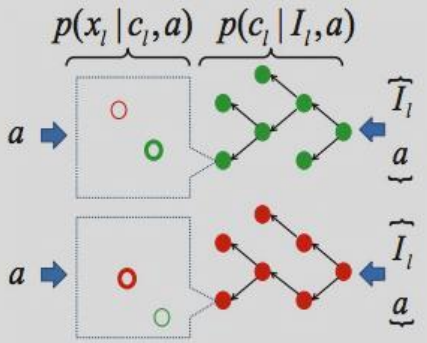


Learning-based (regression)
[Navaratnam et al. BMVC2006]
[Andriluka et al. CVPR2010]

Motivation

- Discriminative approaches (RF) have achieved great success in human body pose estimation.
  - ❖ Efficient – real-time
  - ❖ Accurate – frame-basis, not rely on tracking
  - ❖ Require a large dataset to cover many poses
  - ❖ Train on synthetic, test on real data
  - ❖ Didn't exploit kinematic constraints



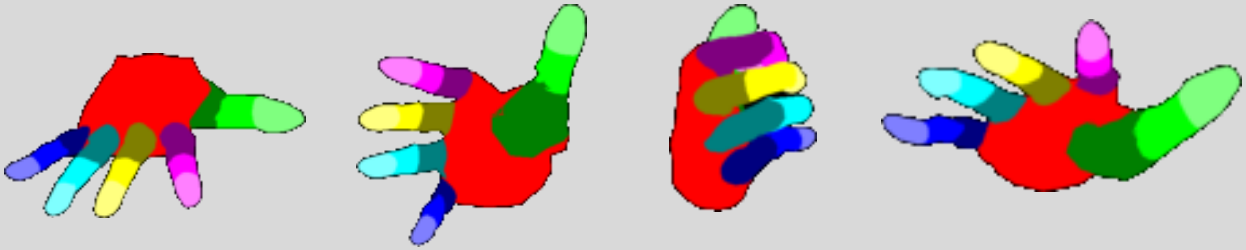


Examples:

Shotton et al. CVPR'11, Girshick et al. ICCV'11, Sun et al. CVPR'12

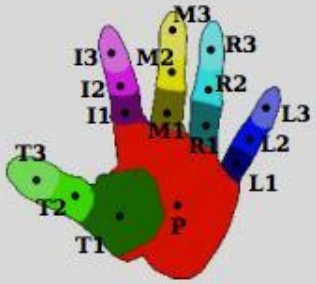
Challenges for Hand?

- Viewpoint changes and self occlusions



- Discrepancy between synthetic and real data is larger than human body

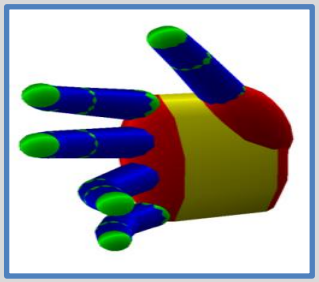
- Labeling is difficult and tedious!



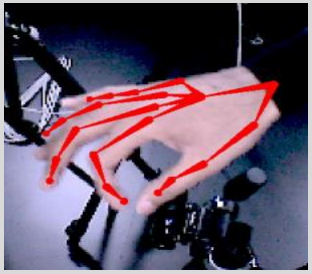
Related Work

Generative

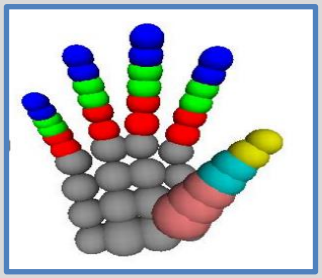
Model-based
Iterative optimisation (PSO)
Need tracking/Intialisation
Robust



Oikonomidis et al.
BMVC'11



Sridhar et al.
ICCV'13

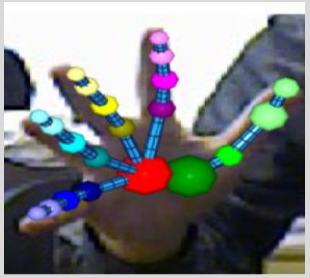


Qian et al.
CVPR'14

Discriminative

Learning-based
No tracking/Intialisation
Efficient
Annotations hard

Wang et al.
SIGGRAPH'09



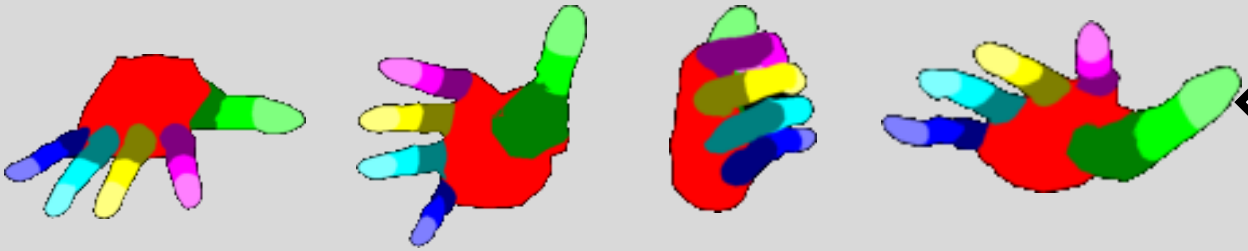
Tang et al.
ICCV'13, CVPR14

Chi et al.
ICCV'13

Imperial College London

Our method

- Viewpoint changes and self occlusions

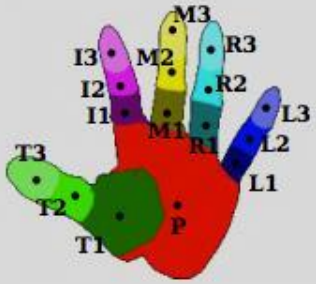


Hierarchical Hybrid Forest

- Discrepancy between synthetic and real data is larger than human body

Transductive Learning

- Labeling is difficult and tedious!

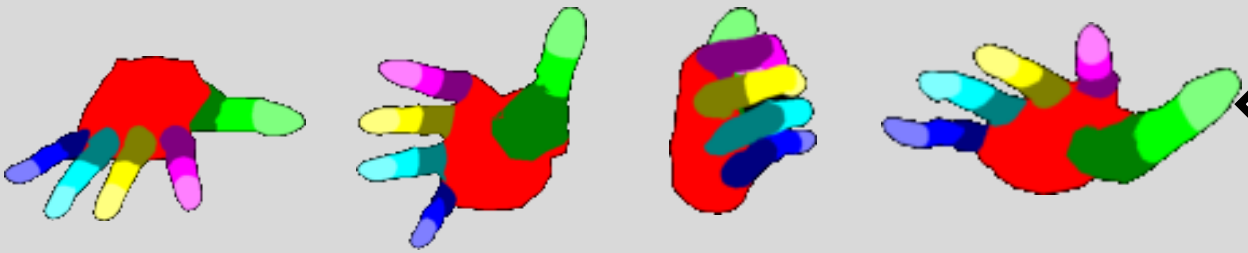


Semi-supervised Learning

Imperial College London

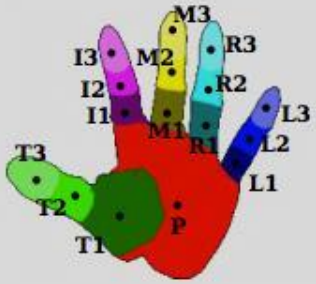
Our method

- Viewpoint changes and self occlusions



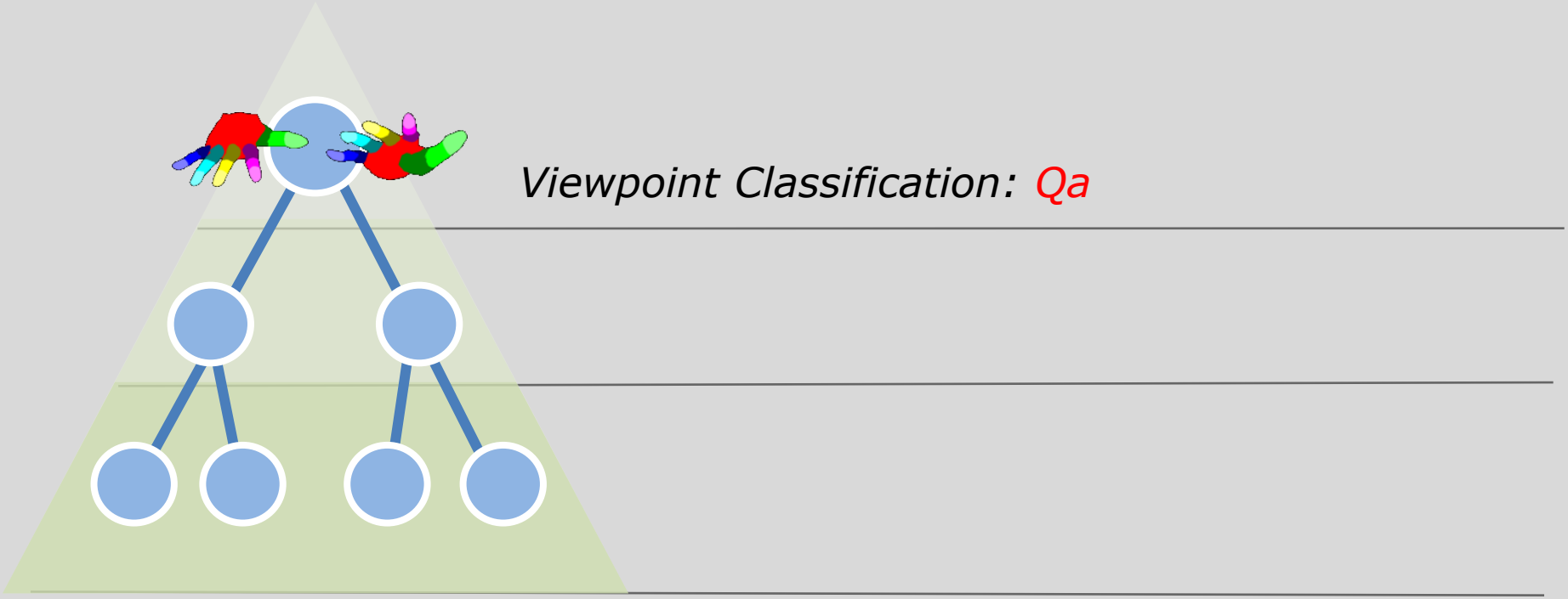
- Discrepancy between synthetic and real data is larger than human body

- Labeling is difficult and tedious!



Imperial College London

Hierarchical Hybrid Forest



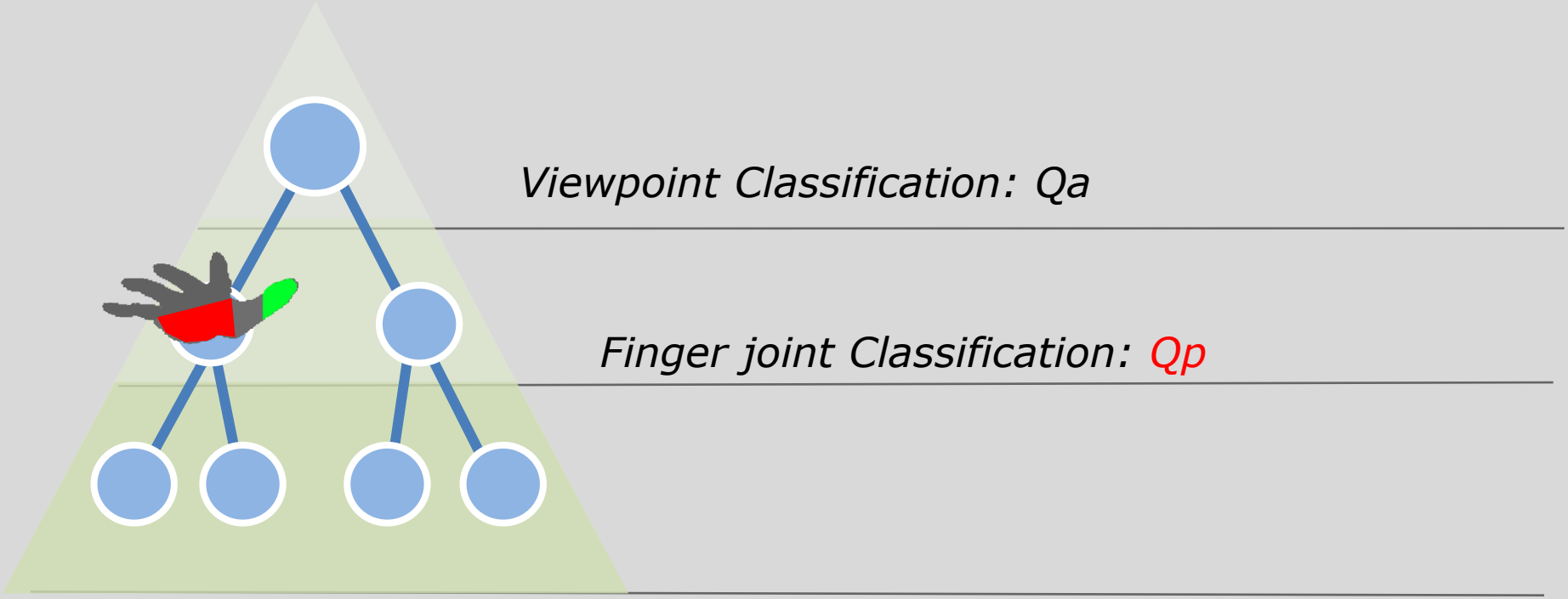
STR forest:

$$Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_P + (1-\alpha)(1-\beta)Q_V$$

- Qa – View point classification quality (Information gain)

Imperial College London

Hierarchical Hybrid Forest



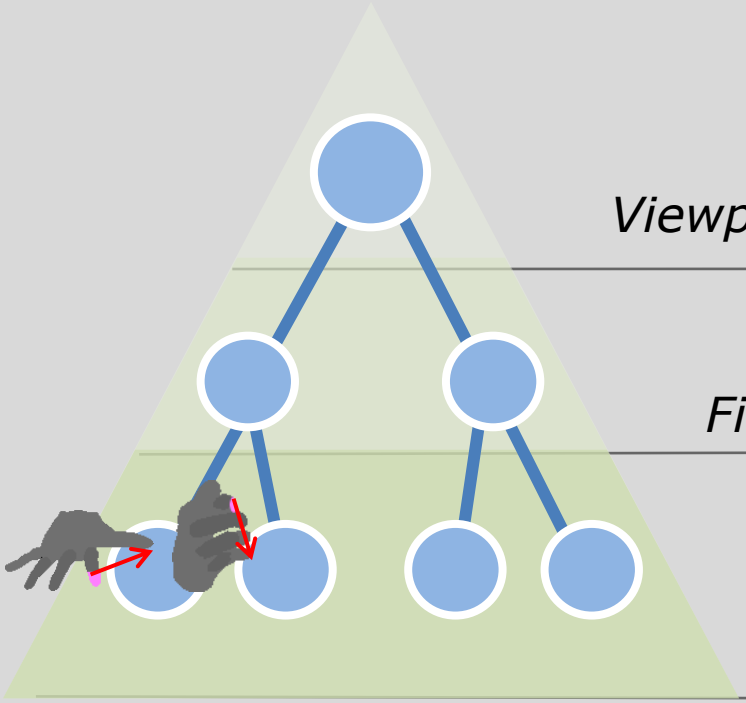
STR forest:

$$Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_P + (1-\alpha)(1-\beta)Q_V$$

- Qa – View point classification quality (Information gain)
- Qp – Joint label classification quality (Information gain)

Imperial College London

Hierarchical Hybrid Forest



Viewpoint Classification: Qa

Finger joint Classification: Qp

Pose Regression: Qv

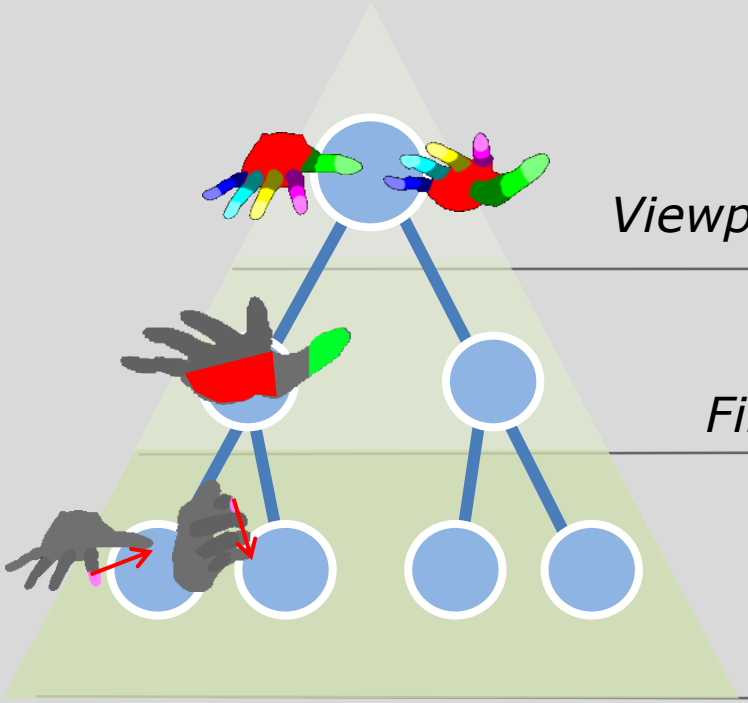
STR forest:

$$Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_P + (1-\alpha)(1-\beta)Q_V$$

- Qa – View point classification quality (Information gain)
- Qp – Joint label classification quality (Information gain)
- Qv – Compactness of voting vectors (Determinant of covariance trace)

Imperial College London

Hierarchical Hybrid Forest



Viewpoint Classification: Qa

Finger Joint Classification: Qp

Pose Regression: Qv

STR forest:

$$Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_P + (1-\alpha)(1-\beta)Q_V$$

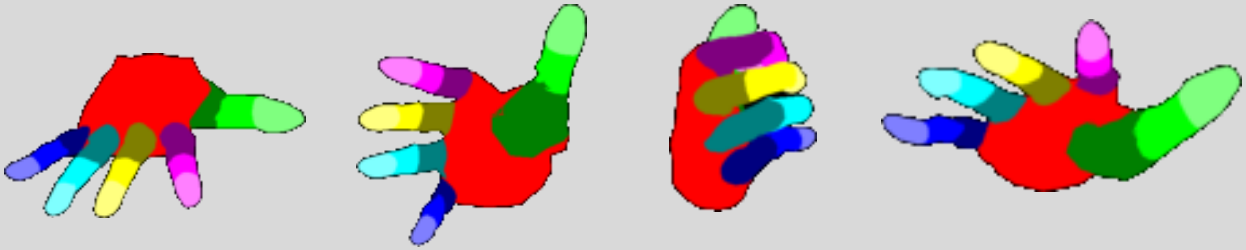
$$\alpha = \begin{cases} 1 \text{ if } \Delta_{\mathbf{a}}(\mathcal{L}) < t_\alpha \\ 0 \text{ otherwise} \end{cases} \qquad \beta = \begin{cases} 1 \text{ if } \Delta_p(\mathcal{L}) < t_\beta \\ 0 \text{ otherwise} \end{cases}$$

- Qa – View point classification quality (Information gain)
- Qp – Joint label classification quality (Information gain)
- Qv – Compactness of voting vectors (Determinant of covariance trace)
- (α,β) – Margin measures of view point labels and joint labels

Imperial College London

Our method

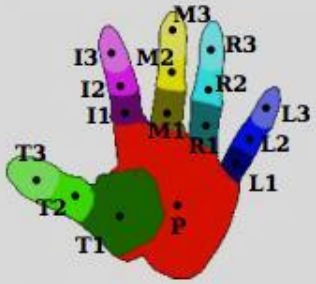
- Viewpoint changes and self occlusions



- Discrepancy between synthetic and real data is larger than human body

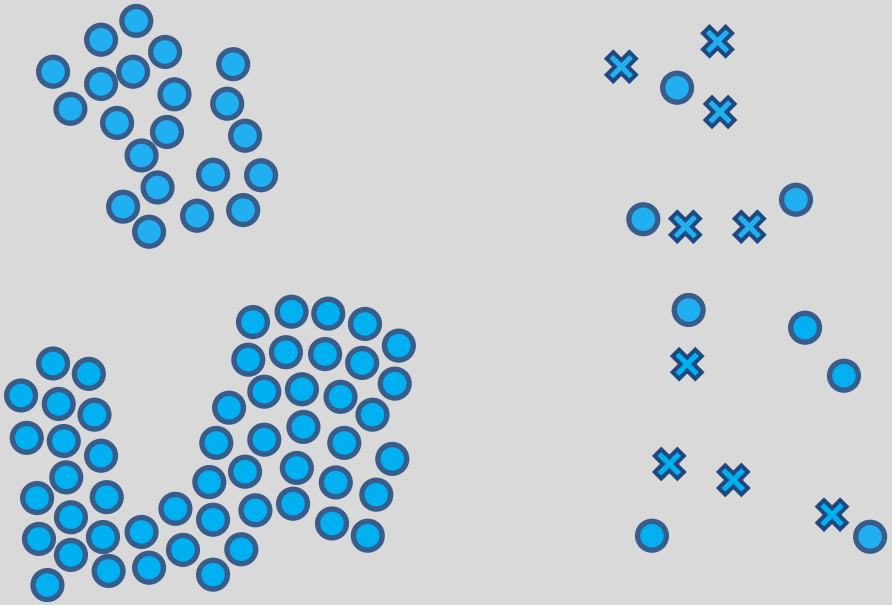
Transductive Learning

- Labeling is difficult and tedious!



Semi-supervised Learning

Transductive learning



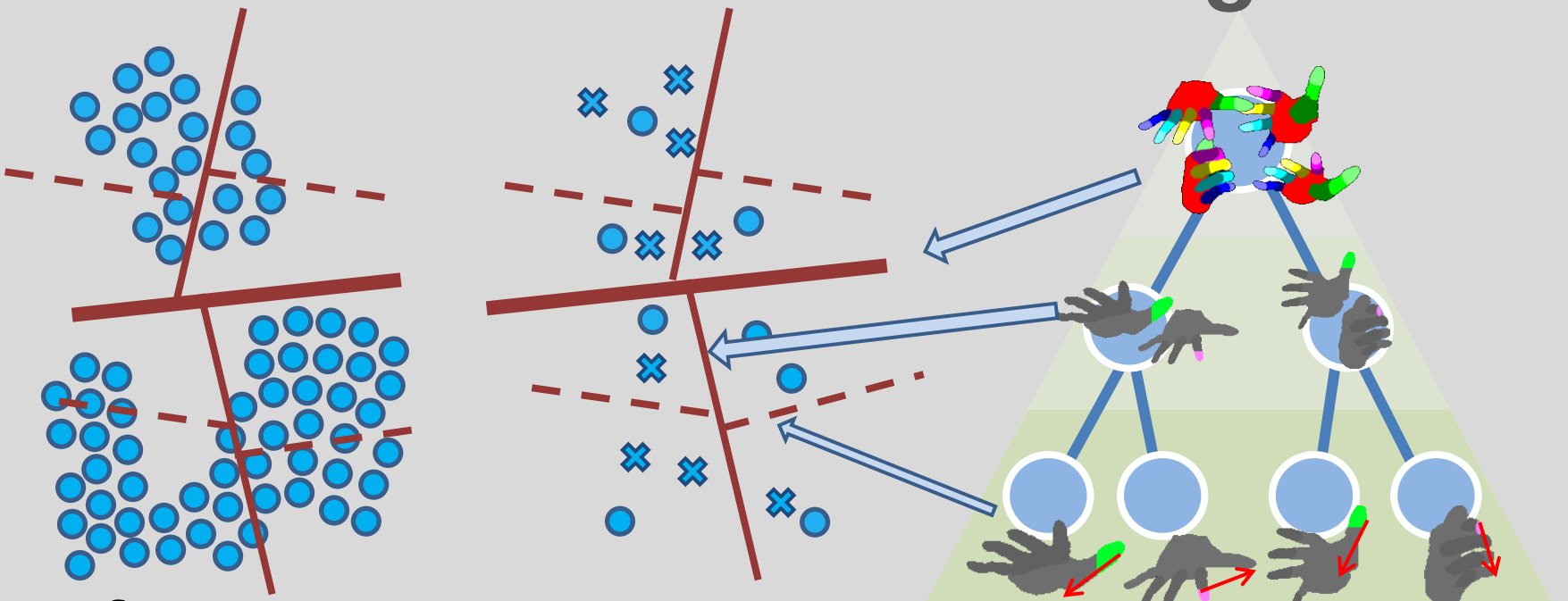
Source space
(Synthetic data *S*)

Target space
(Realistic data *R*)

Training data D = {R_l, R_u, S}: labeled unlabeled

- Synthetic data **S**:
  - »Generated from an articulated hand model. All labeled.
- Realistic data **R**:
  - »Captured from Primesense depth sensor
  - »A small part of **R**, $\mathbf{R_l}$ are labeled manually (unlabeled set $\mathbf{R_u}$)

Transductive learning



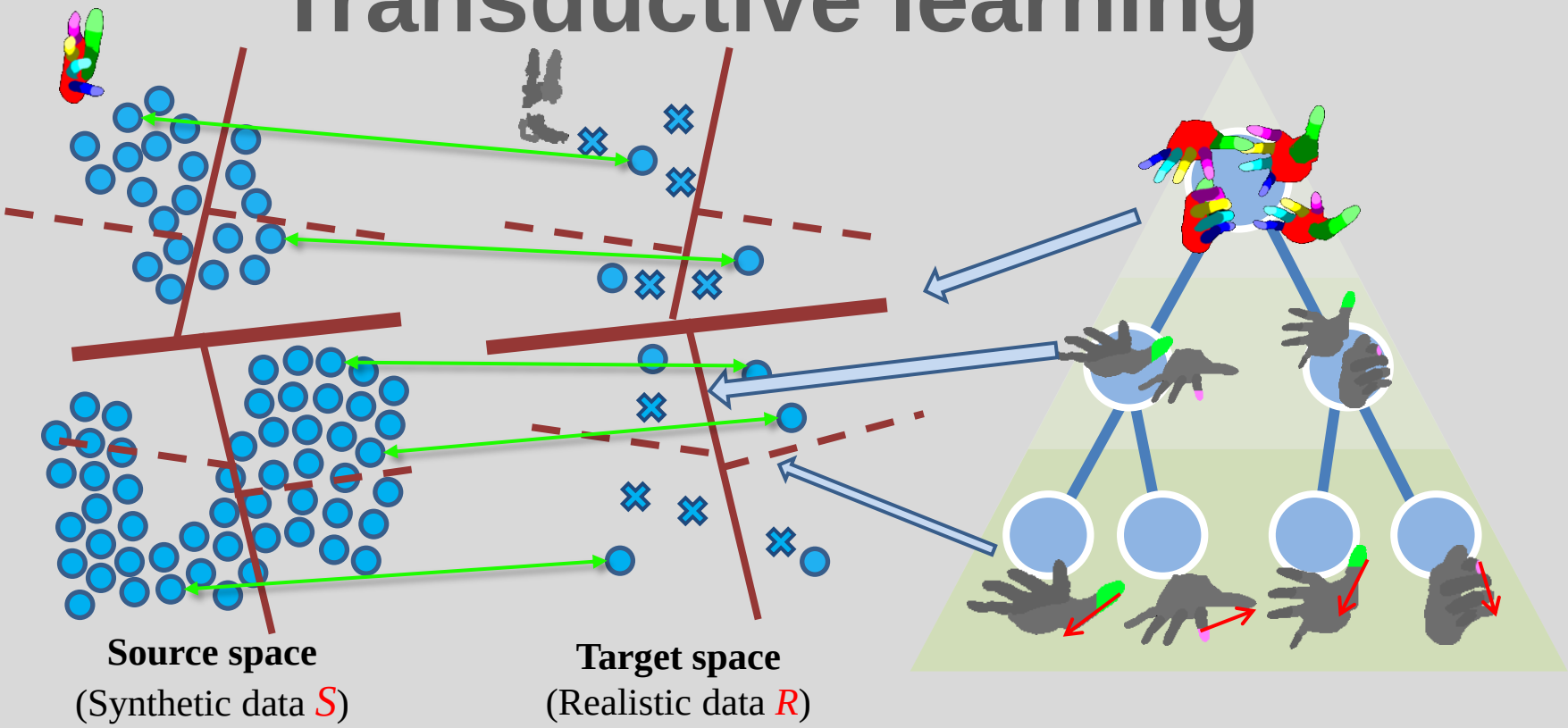
Source space
(Synthetic data *S*)

Target space
(Realistic data *R*)

Training data D = {R_l, R_u, S}:

- Realistic data **R**:
  - » Captured from Kinect
  - » A small part of **R**, $\mathbf{R_l}$ are labeled manually (unlabeled set $\mathbf{R_u}$)
- Synthetic data **S**:
  - » Generated from a articulated hand model, where |**S**| >> |**R**|

Transductive learning



Source space
(Synthetic data *S*)

Target space
(Realistic data *R*)

Training data D = {R_l, R_u, S}:

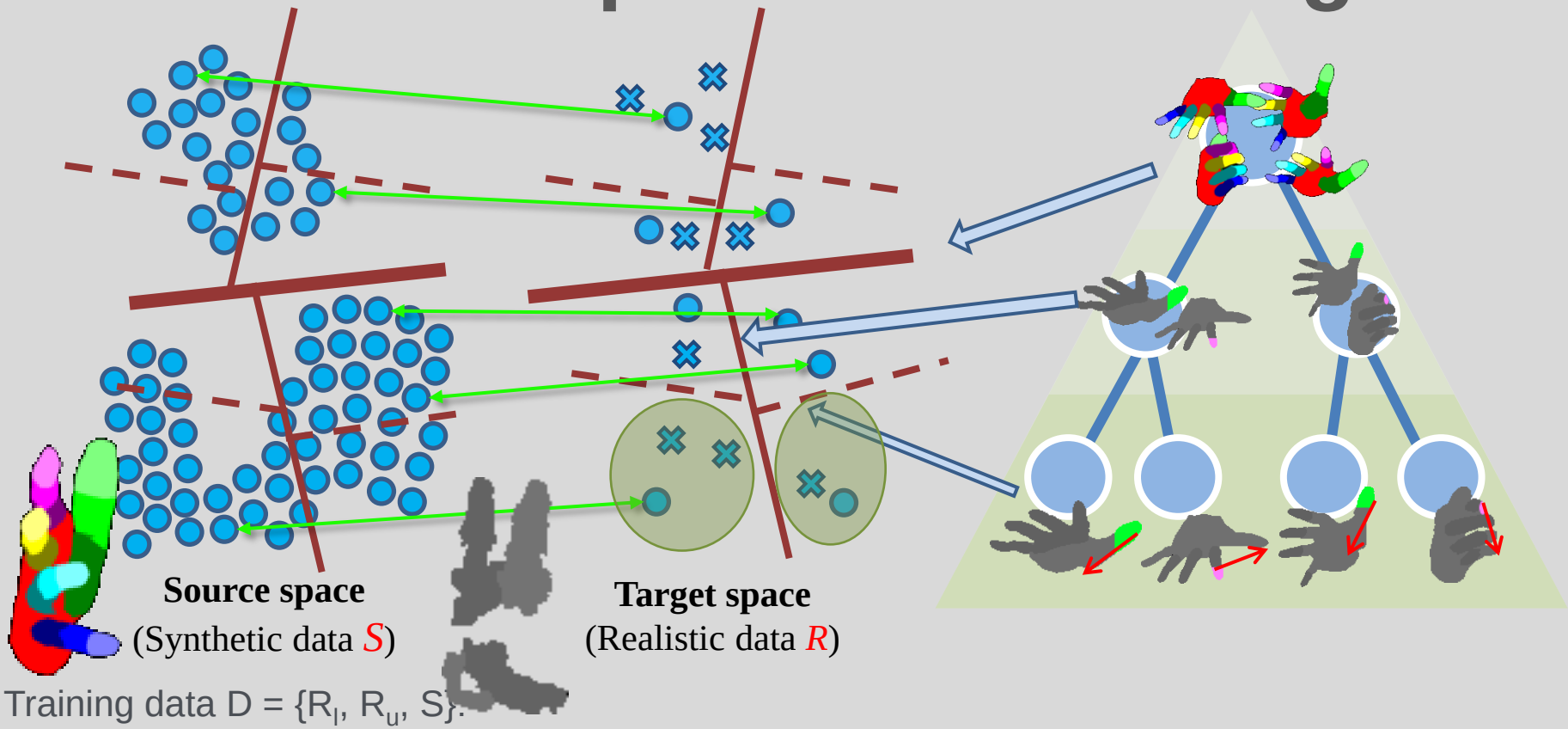
- Similar data-points in $\mathbf{R_l}$ and **S** are paired(if separated by split function give penalty)

$$Q_t = \frac{|\{r,s\} \subset \mathcal{L}_{lc}| + |\{r,s\} \subset \mathcal{L}_{rc}|}{|\{r,s\} \subset \mathcal{L}|}$$

$$\forall \{r,s\} \subset \mathcal{L} \text{ where } \Psi(r,s) = 1$$

Imperial College London

Semi-supervised learning



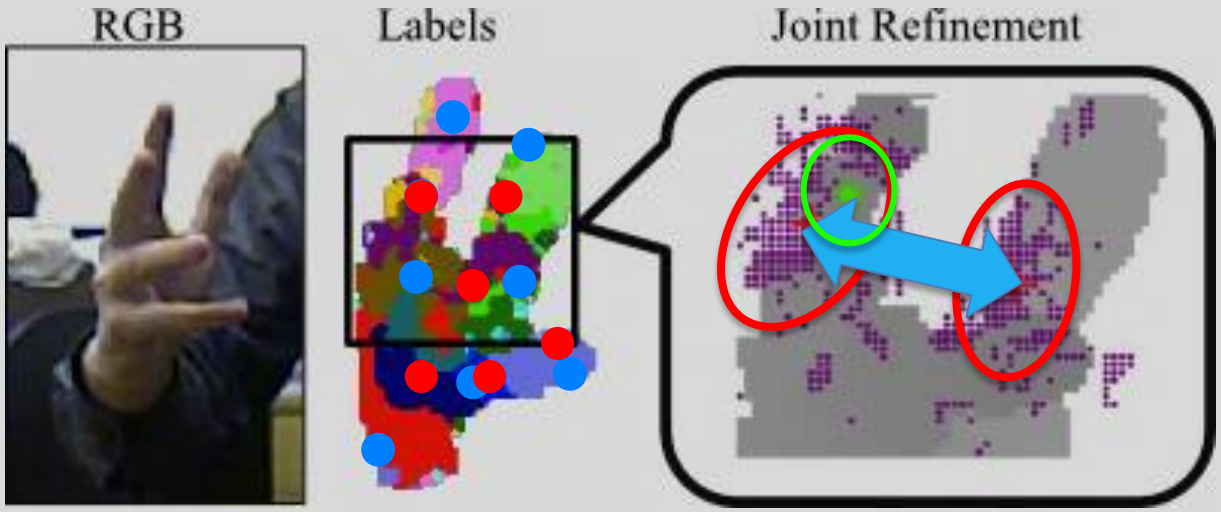
Training data D = {R_l, R_u, S}.

- Similar data-points in $\mathbf{R_l}$ and **S** are paired(if separated by split function give penalty)
- Introduce a semi-supervised term to make use of unlabeled real data when evaluating split function

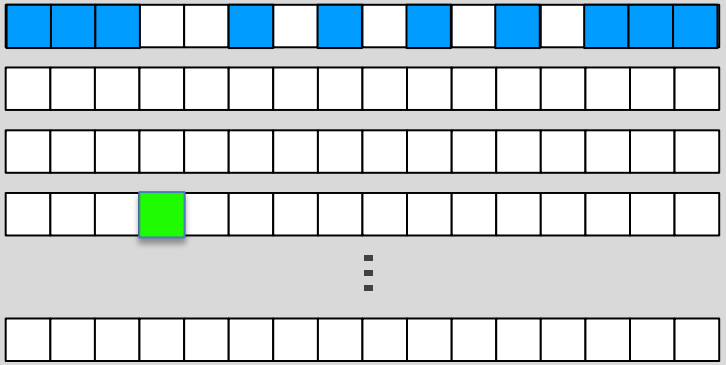
$$Q_u = \left[1 + \frac{|\mathcal{R}_{lc}|}{|\mathcal{R}|}\Lambda(\mathcal{R}_{lc}) + \frac{|\mathcal{R}_{rc}|}{|\mathcal{R}|}\Lambda(\mathcal{R}_{rc})\right]^{-1}$$

Imperial College London

Kinematic refinement



Joint position database



Experiment settings

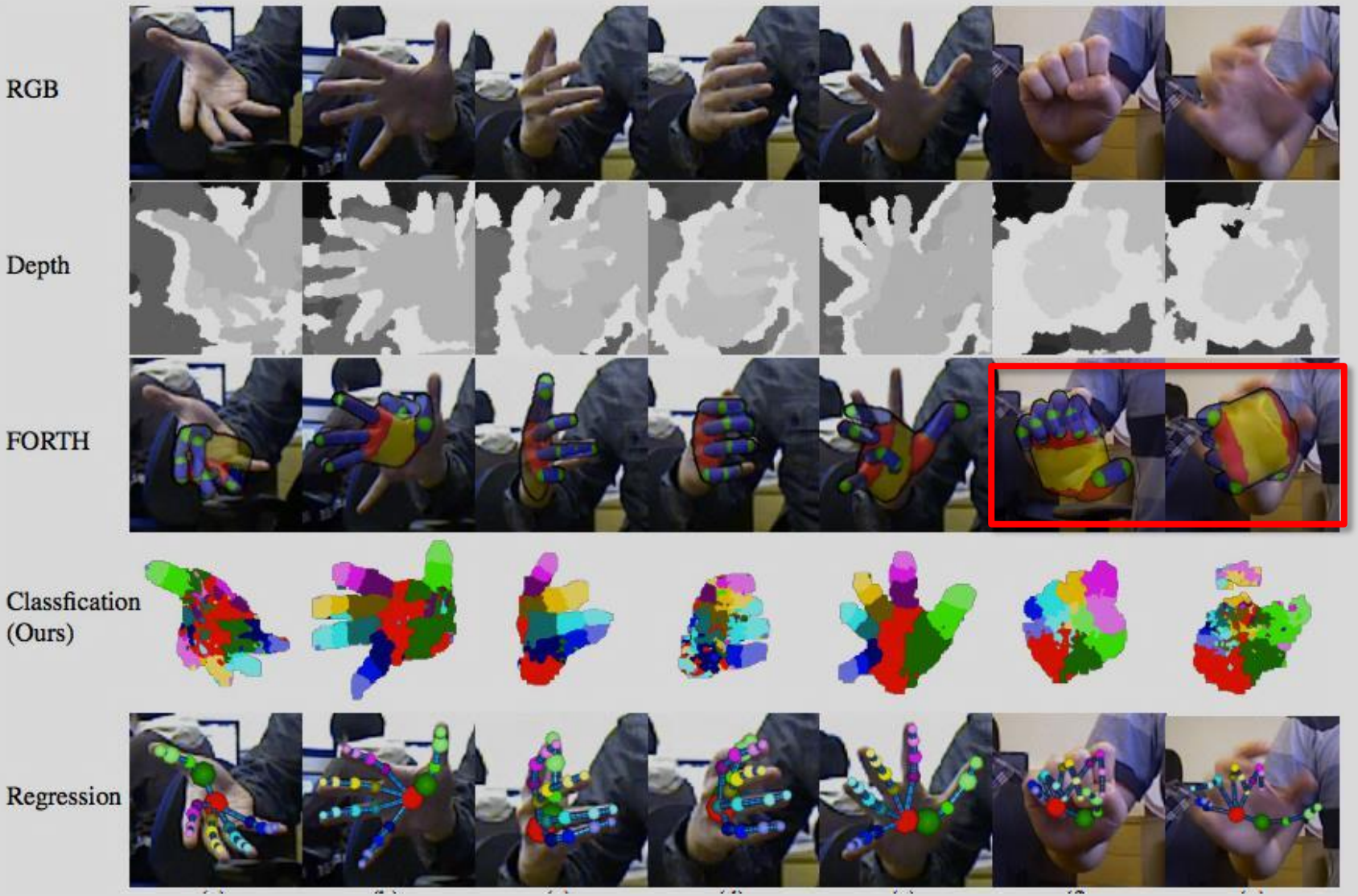
Training data:

- *Synthetic data(337.5K images)*
- *Real data(81K images, <1.2K labeled)*

Evaluation data:

- *Three different testing sequences*
  1. *Sequence A --- Single viewpoint(450 frames)*
  2. *Sequence B --- Multiple viewpoints, with slow hand movements(1000 frames)*
  3. *Sequence C --- Multiple viewpoints, with fast hand movements(240 frames)*

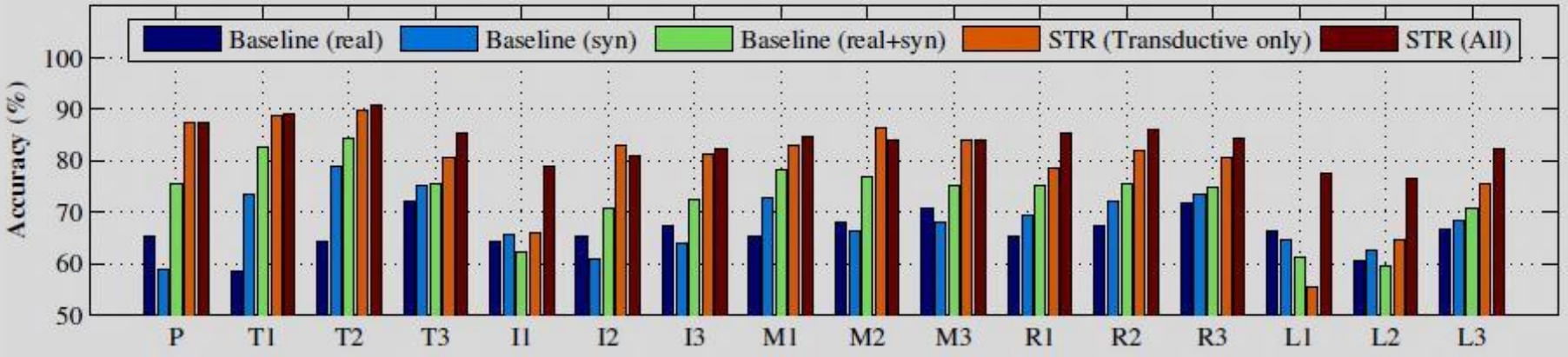
Imperial College London



Imperial College London

Self comparison experiment

Self comparison(Sequence A):



» This graph shows the joint classification accuracy of Sequence A.

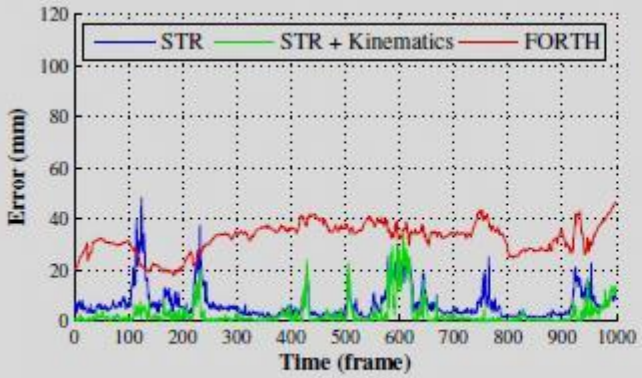
» Realistic and synthetic baselines produced similar accuracies.

» Using the transductive term is better than simply augmented real and synthetic data.

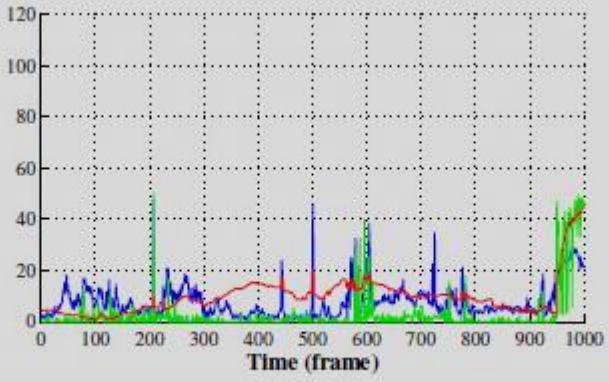
» All terms together achieves the best results.

Imperial College London

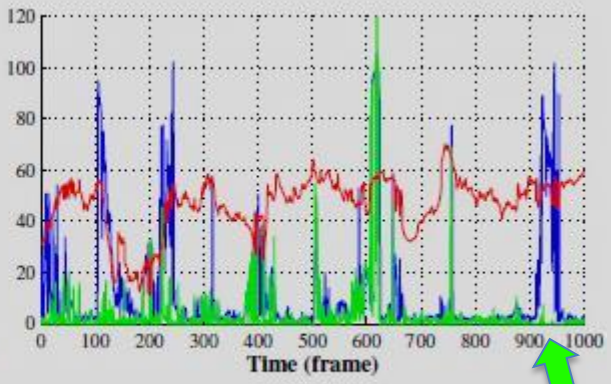
Multiview experiments



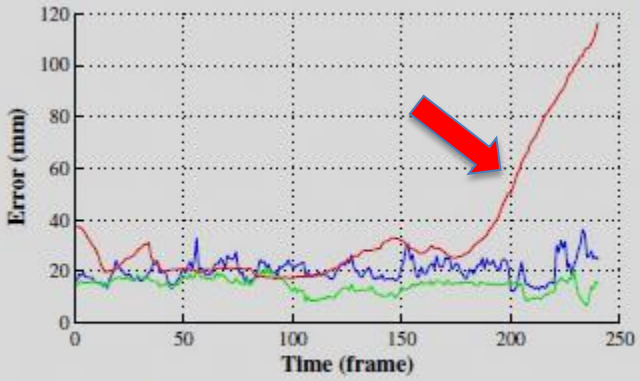
(a) Test sequence B (Average error)



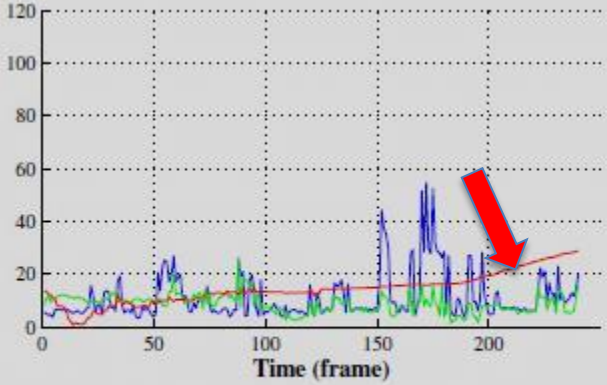
(b) Test sequence B (Palm)



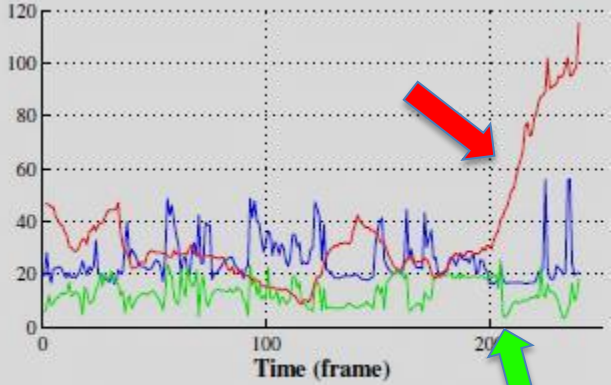
(c) Test sequence B (Index finger tip)



(d) Test sequence C (Average error)



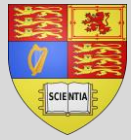
(e) Test sequence C (Palm)



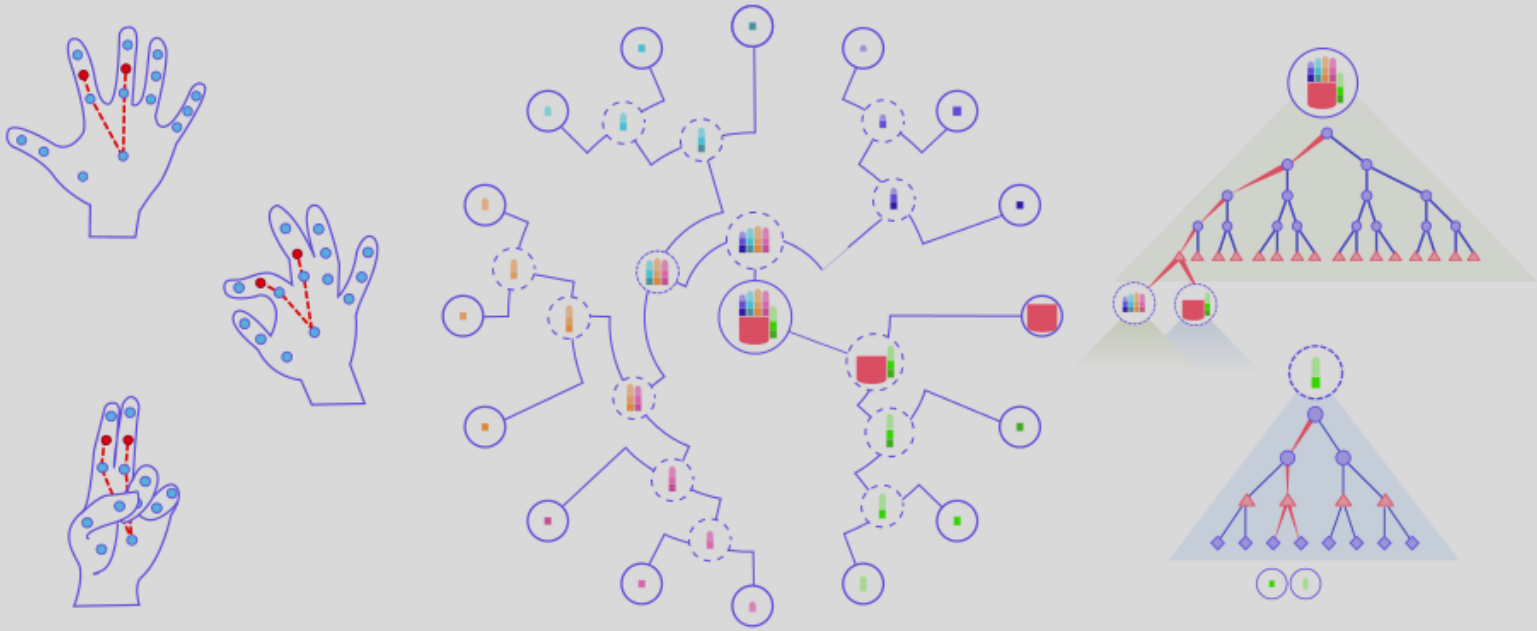
(f) Test sequence C (Index finger tip)

Self comparison

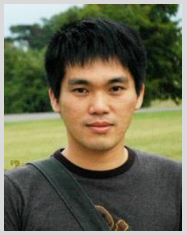
Deep Architecture:
Latent Regression Forest



Imperial College
Londc



Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture



Danhang Tang

Hyung Jin Chang

Alykhan Tejani

T-K Kim

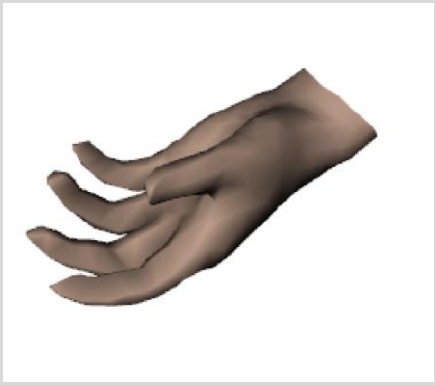
Imperial College London

CVPR 2014 (oral presentation)

Motivation

- Existing discriminative approaches

- Holistic
- Direct mapping between X and Y
- Efficient
- Less flexible – need refinement
- Kinematic contrained



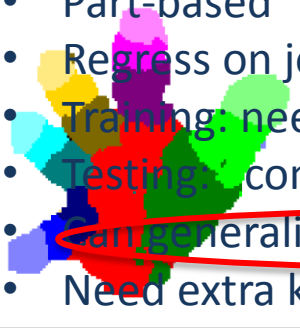
Real-Time Hand-Tracking with a Color Glove
Wang and Popovic, SIGGRAPH '09

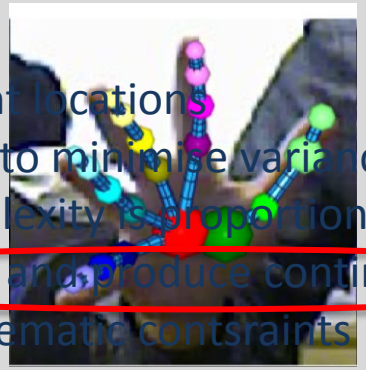
Motivation

- Existing discriminative approaches

- Holistic
- Direct mapping between X and Y
- Efficient
- Less flexible – need refinement
- Kinematic contrained

Real-Time Hand-Tracking with a Color Glove
Wang and Popovic, SIGGRAPH '09





- Part-based
- Regress on joint locations
- Training: need to minimise variance of 48 dimensional vectors
- Testing: complexity is proportional to number of patches
- Can generalise and produce continuous results
- Need extra kinematic contsraints

Real-time Hand Pose Estimation using Transductive Regression Forests
Tang et al., ICCV '13

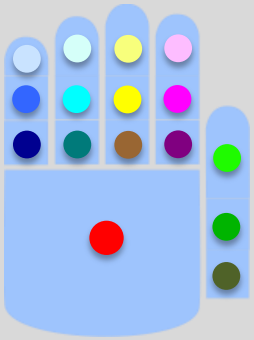
Our Method

- What if?
  - We can predict the possible locations of joints.
  - Only apply a “sub-detector” that trained with possible samples to that area.
- Our solution - model the inference process as a topology-guided search,in order to maintain both efficiency and flexibility.
  - Represent the topology with Latent Tree Model*, and learn it in an unsupervised manner.
  - Propose a novel algorithm called Latent Regression Forest, which achieves 62.5 fps without CPU/GPU optimisation.
  - Propose a 330K fully annotated,multi-subject dataset.

*“Learning Latent Tree Graphical Models” ,
Choi et al., JMLR, 2011

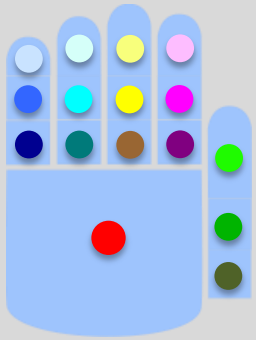
Learning the hand topology

- A representation for hand topology is needed.



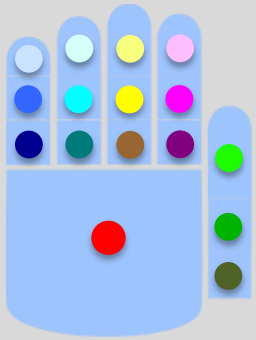
Learning the hand topology

- A representation for hand topology is needed.
- We model hand topology as a binary Latent Tree Model(LTM) in a coarse-to-fine manner.



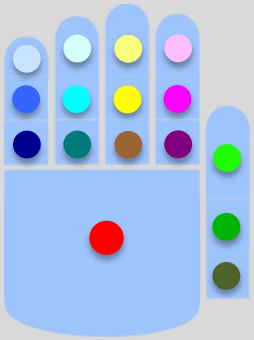
Learning the hand topology

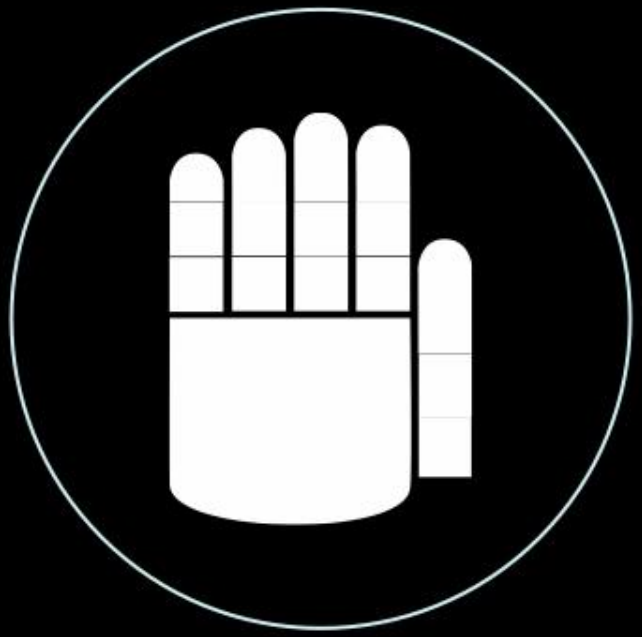
- A representation for hand topology is needed.
- We model hand topology as a binary Latent Tree Model(LTM) in a coarse-to-fine manner.
  - Manually define it with prior knowledge?



Learning the hand topology

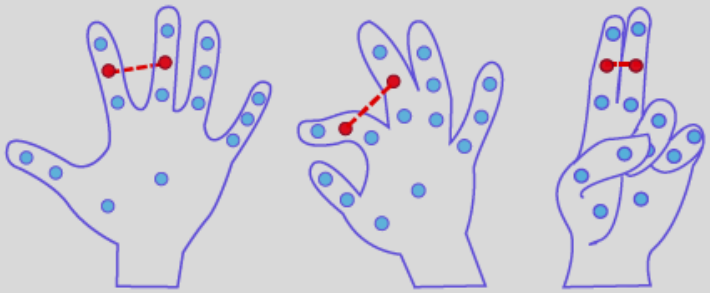
- A representation for hand topology is needed.
- We model hand topology as a binary Latent Tree Model(LTM) in a coarse-to-fine manner.
  - Manually define it with prior knowledge?
  - Learn it Chow-Liu Neighbour-Joining (CLNJ)
    - “Learning Latent Tree Graphical Models” , Choi et al., JMLR, 2011
    - No need for prior knowledge.
    - Can be applied to any articulated object.

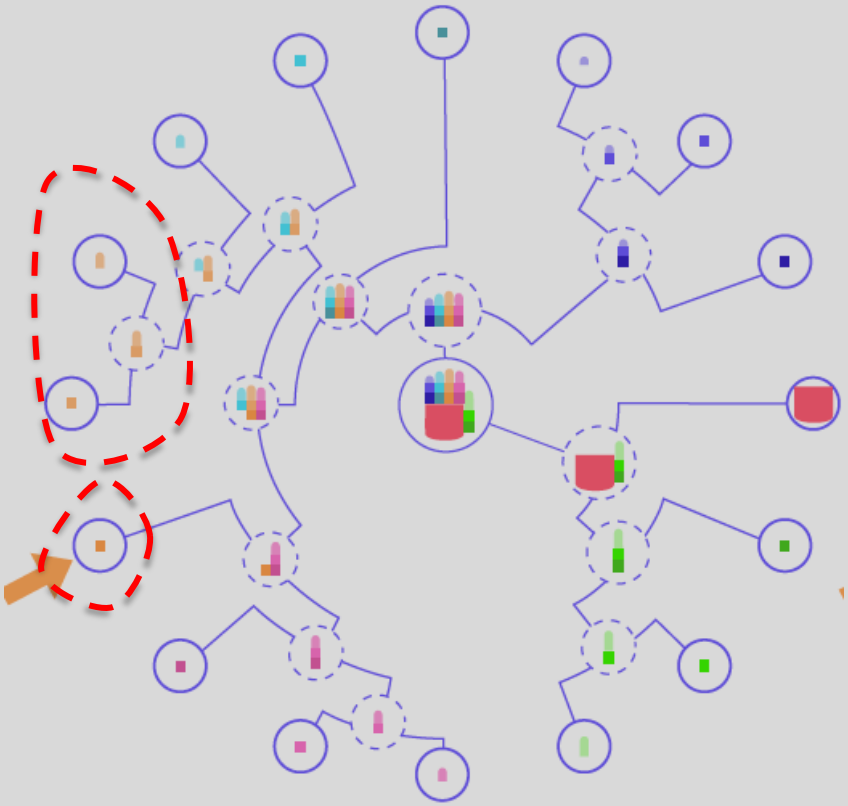




Define Distance Metric

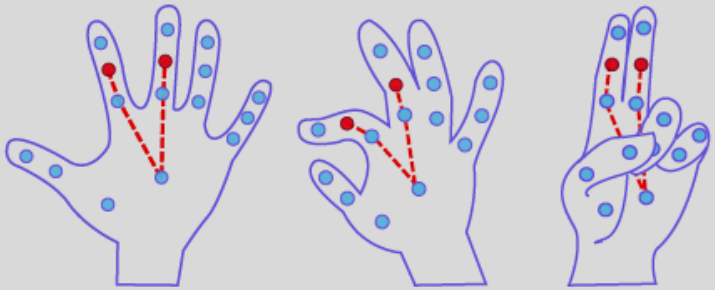
- 3D Euclidean distance
  - Not pose invariant

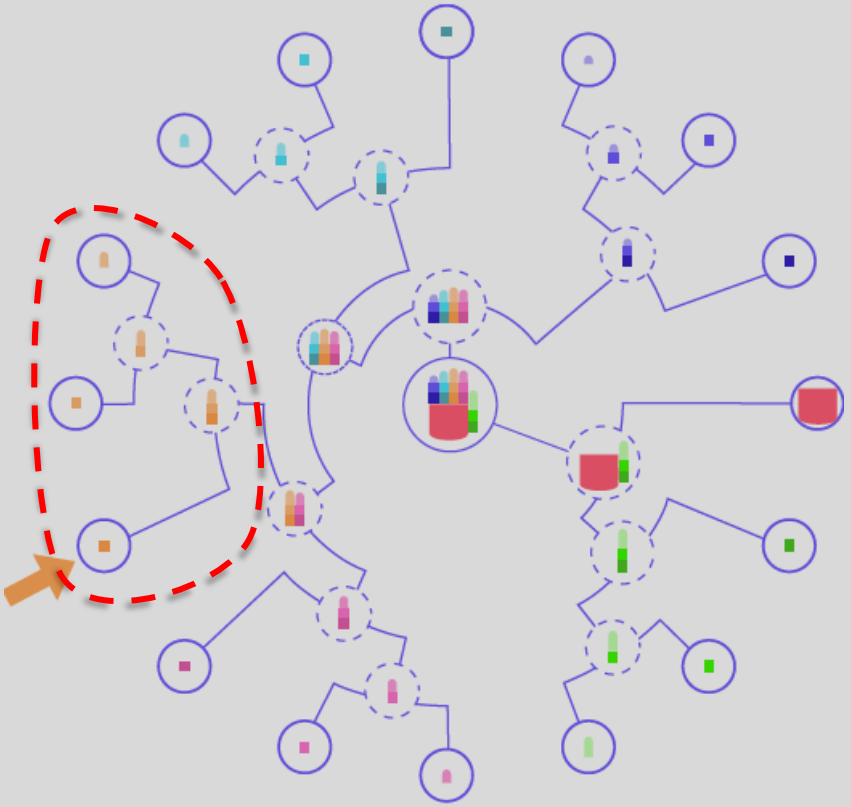




Define Distance Metric

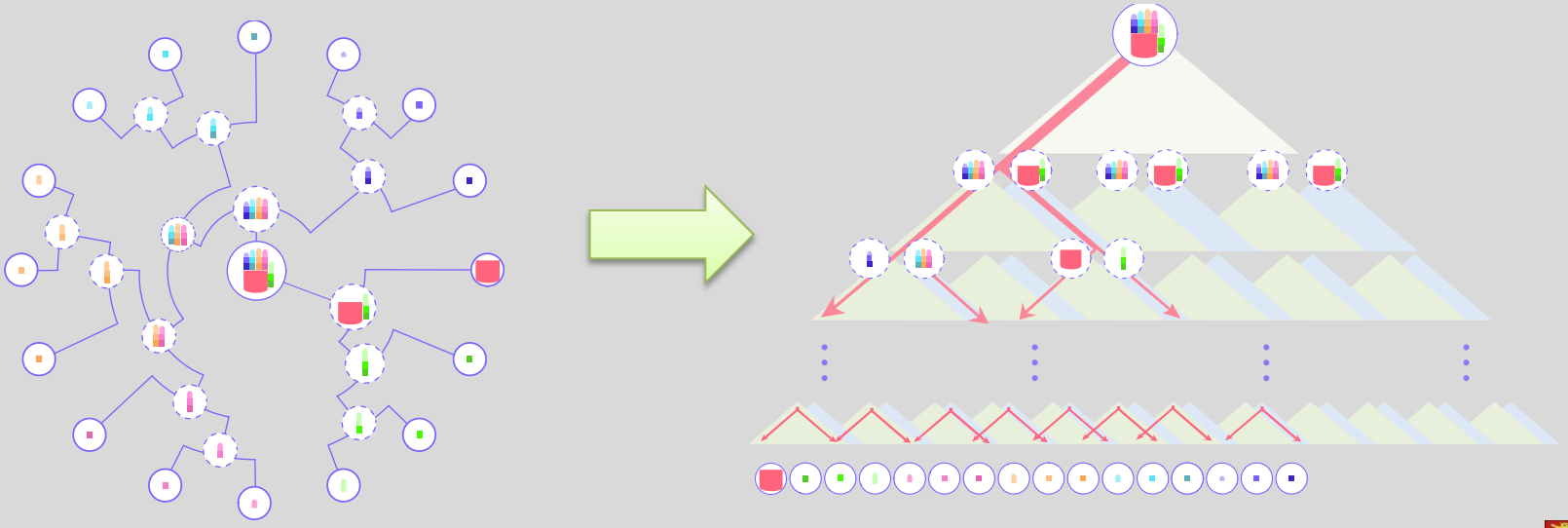
- Geodesic distance
  - Consider all joints as an undirected graph
  - Remove edges with discontinuous z-values

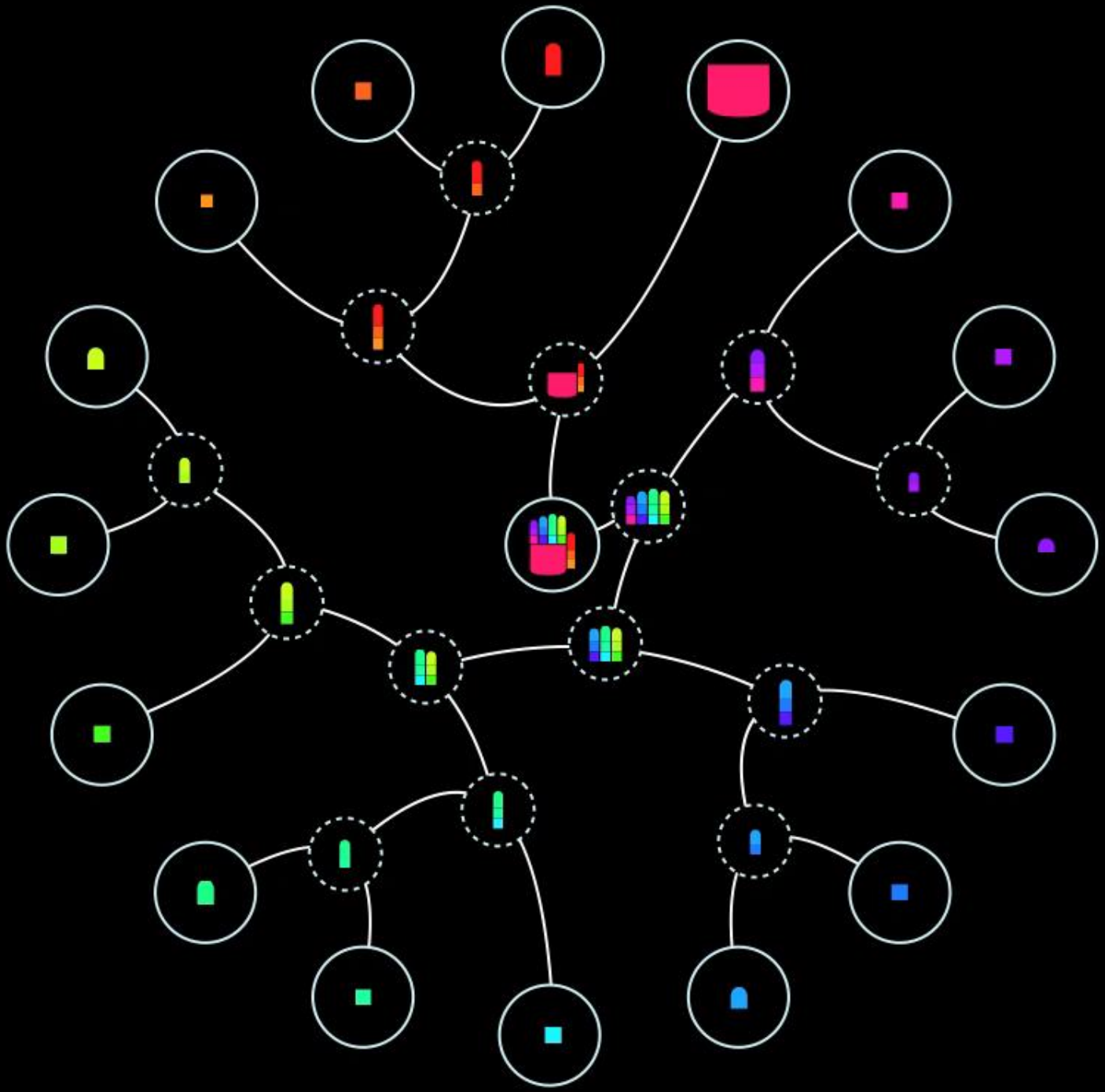




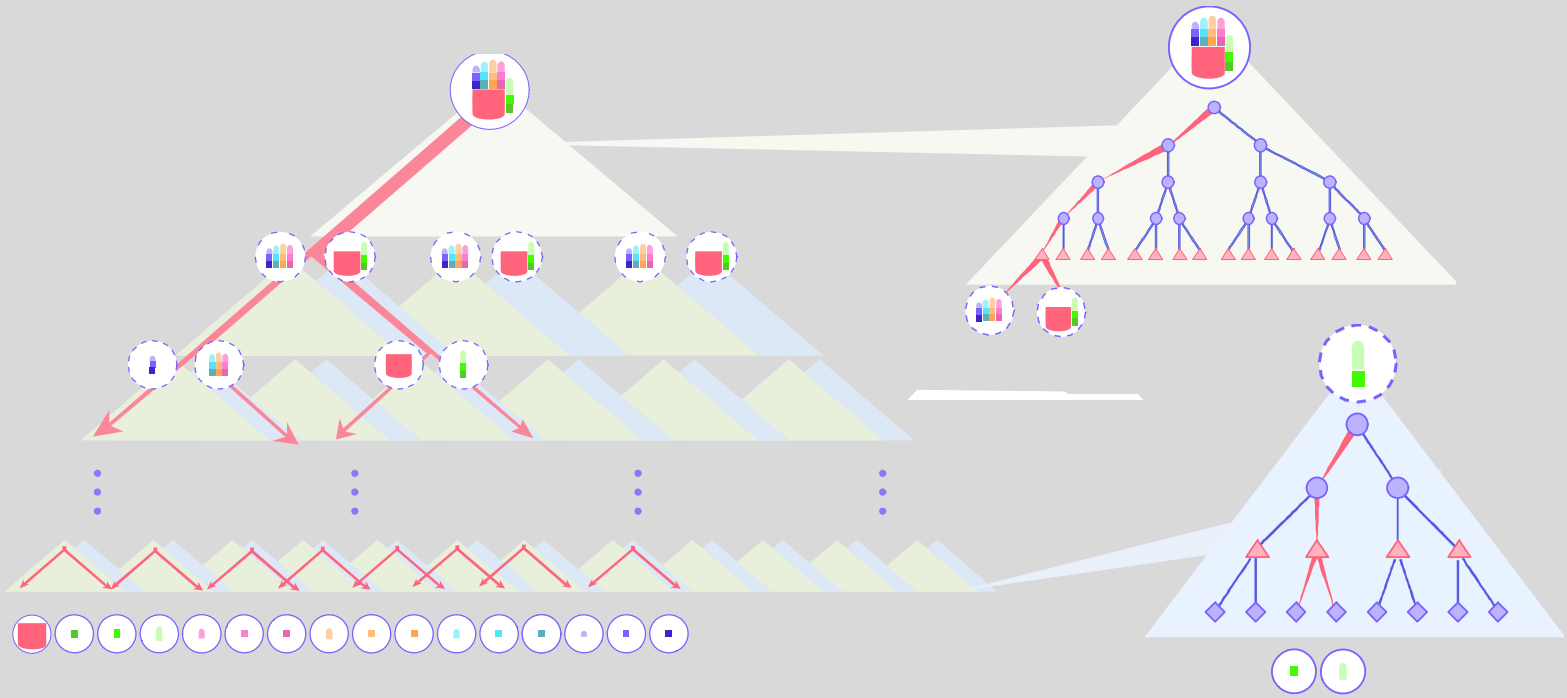
Latent Regression Forest

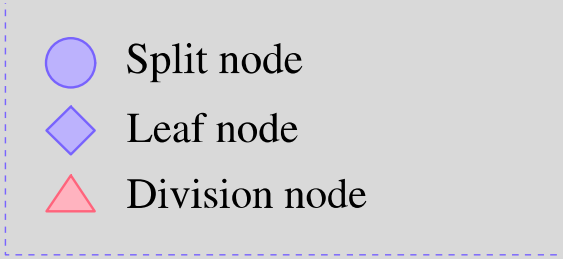
- A Latent Tree Model represents a coarse-to-fine topology of hand.
- We still need a regressor to perform the regression.
- Solution: Guided by the LTM, we train a multi-layer regression forest, termed as Latent Regression Forest



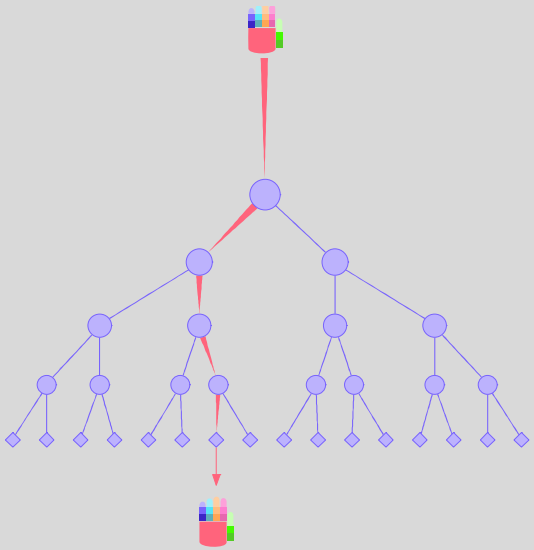


Latent Regression Forest

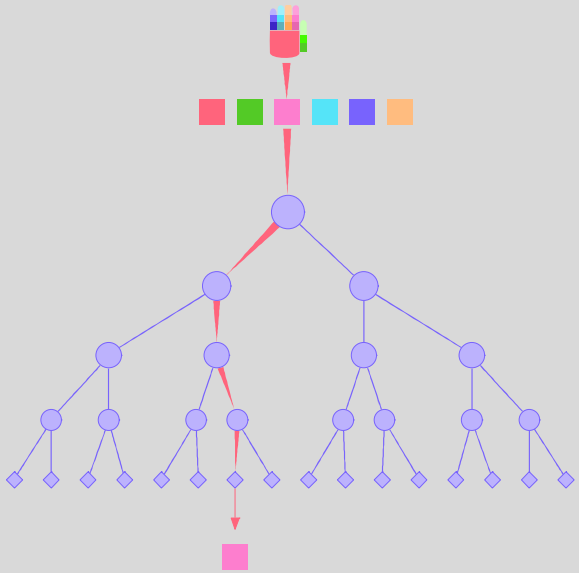




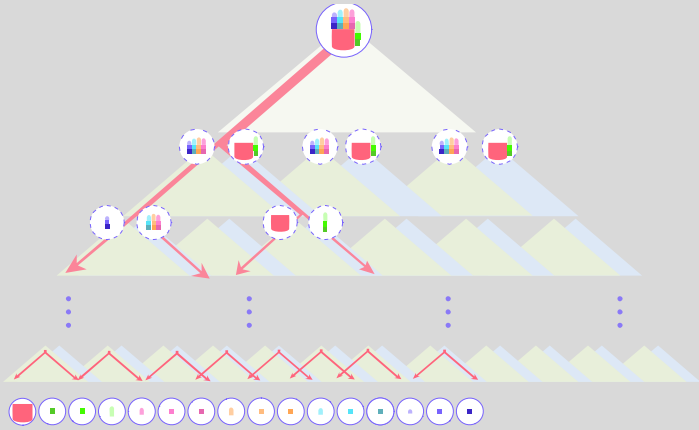
Comparison



Holistic:
1 sample(whole hand)
1 leaf node



Part-based:
N samples(N=num of pixels)
N leaf nodes
Regress on 48 dim vectors

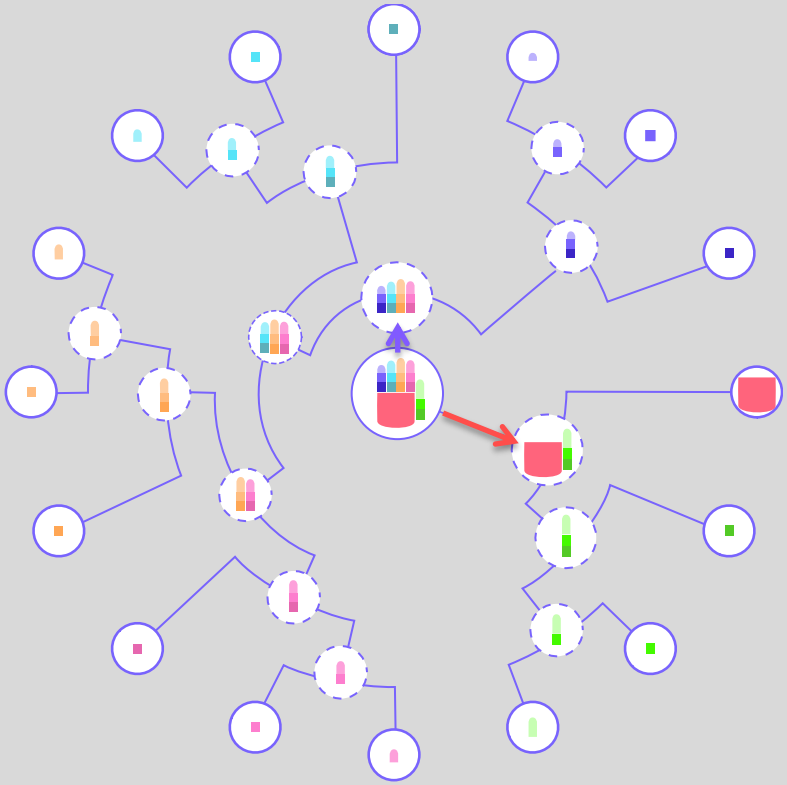


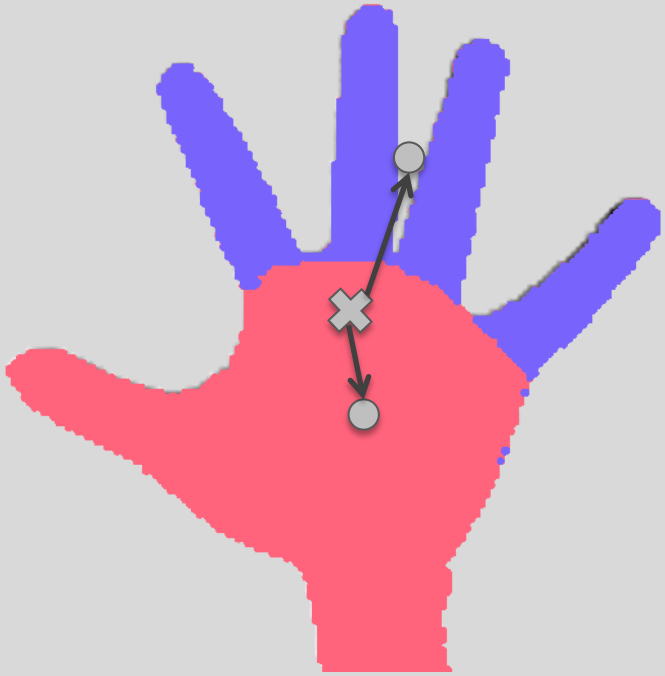
LRF:
1 sample
16 leaf nodes(skeletal parts)
Regress on 6 dim vectors

Complexity(proportional to num of samples): Part-based >> LRF ≈ Holistic

Inference

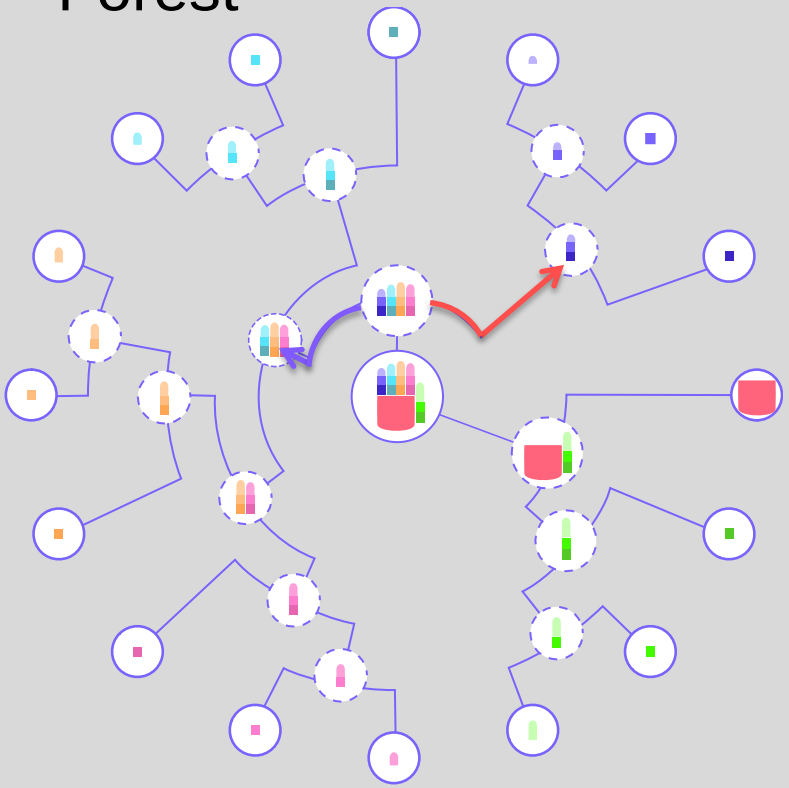
- A search process, guided by a binary Latent Tree Model, performed by the Latent Regression Forest.

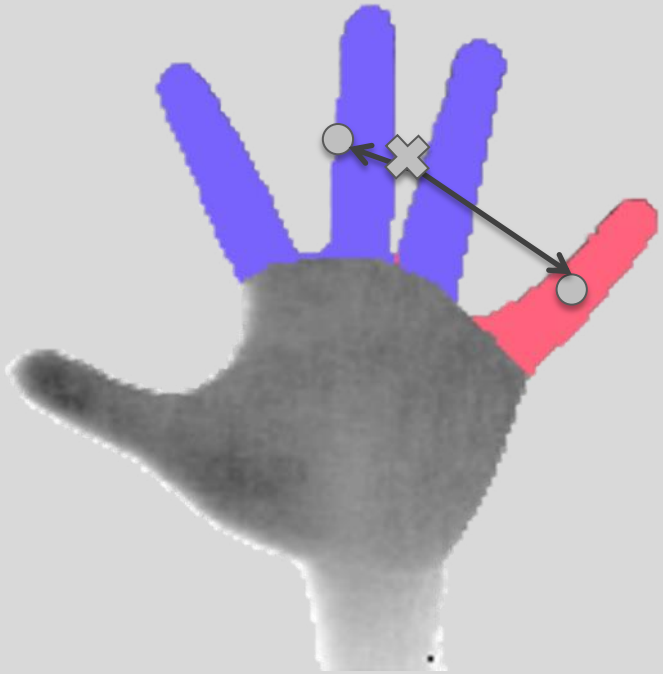




Inference

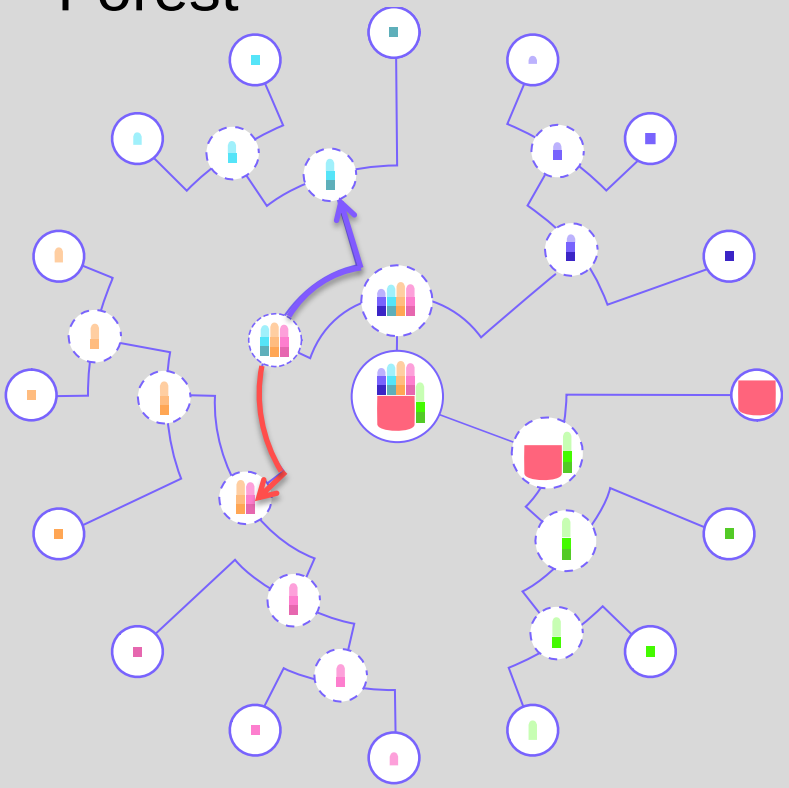
- A search process, guided by a binary Latent Tree Model(LTM), performed by the Latent Regression Forest





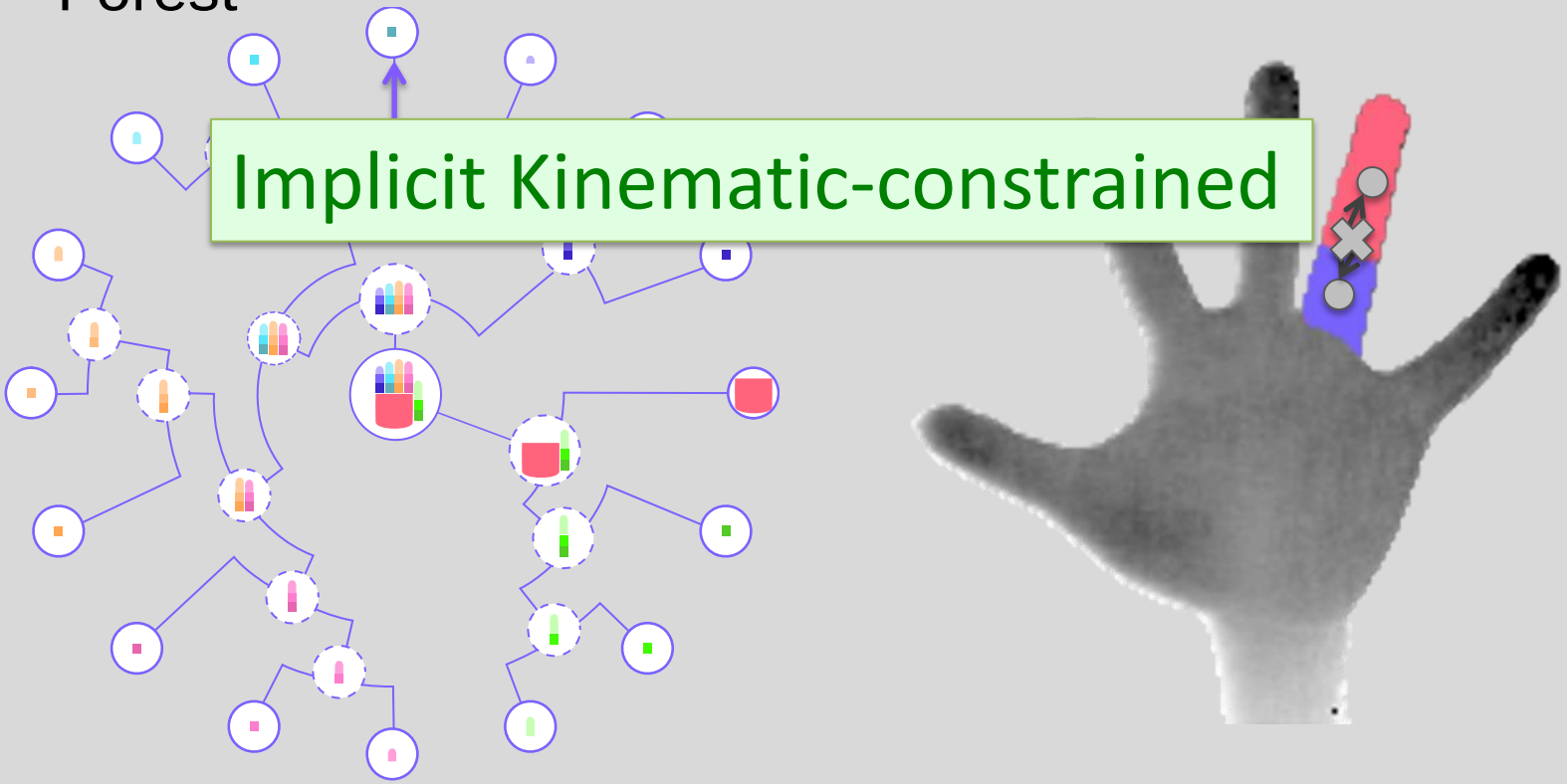
Inference

- A search process, guided by a binary Latent Tree Model(LTM), performed by the Latent Regression Forest



Inference

- A search process, guided by a binary Latent Tree Model(LTM), performed by the Latent Regression Forest

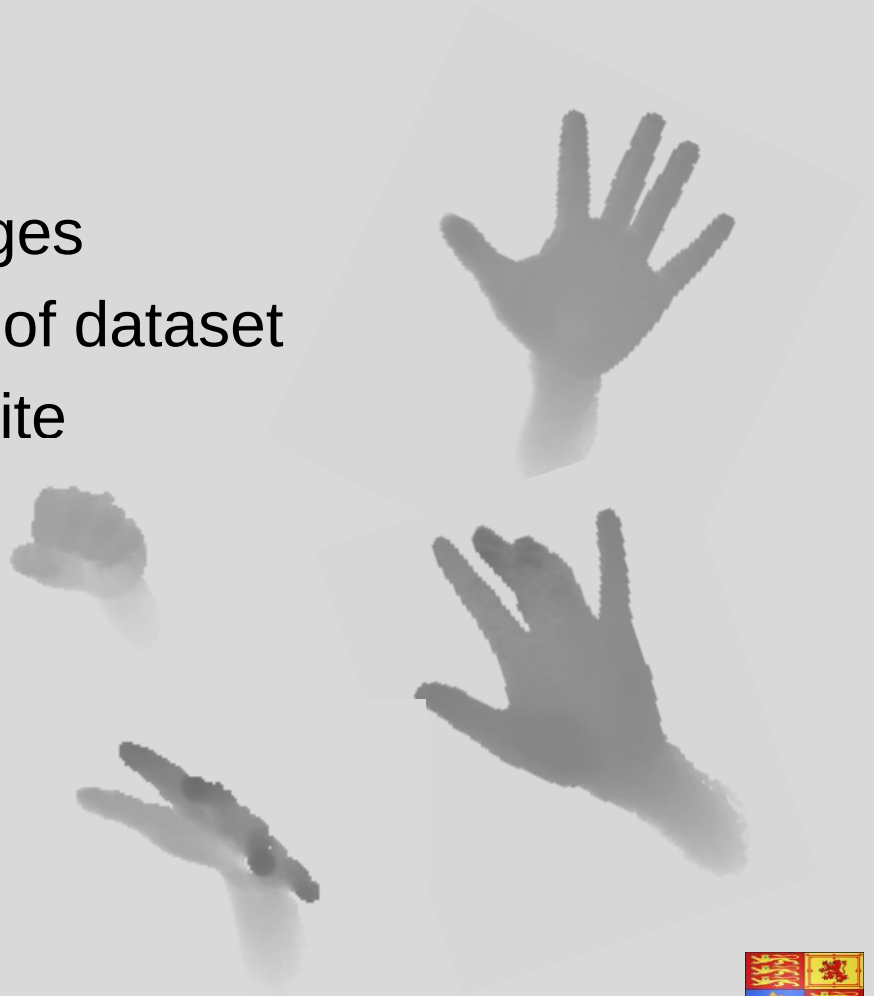


Imperial College London

Dataset and Setting

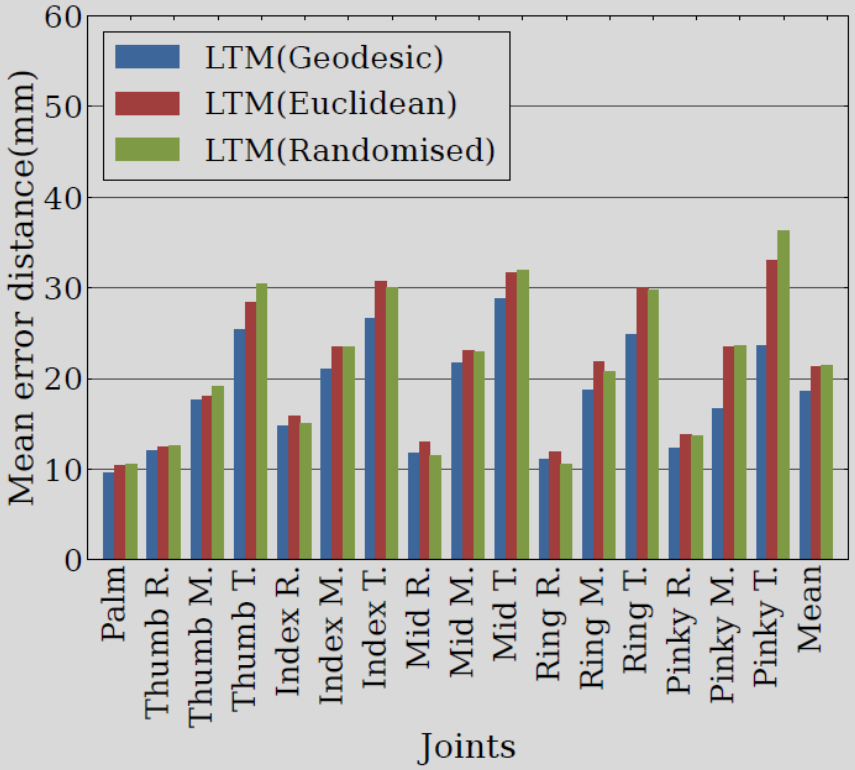
- Intel Creative sensor
- 10 different subjects
- 22K of fully labeled depth images
- In-plane rotated to have 330K of dataset
- Publicly available on our website

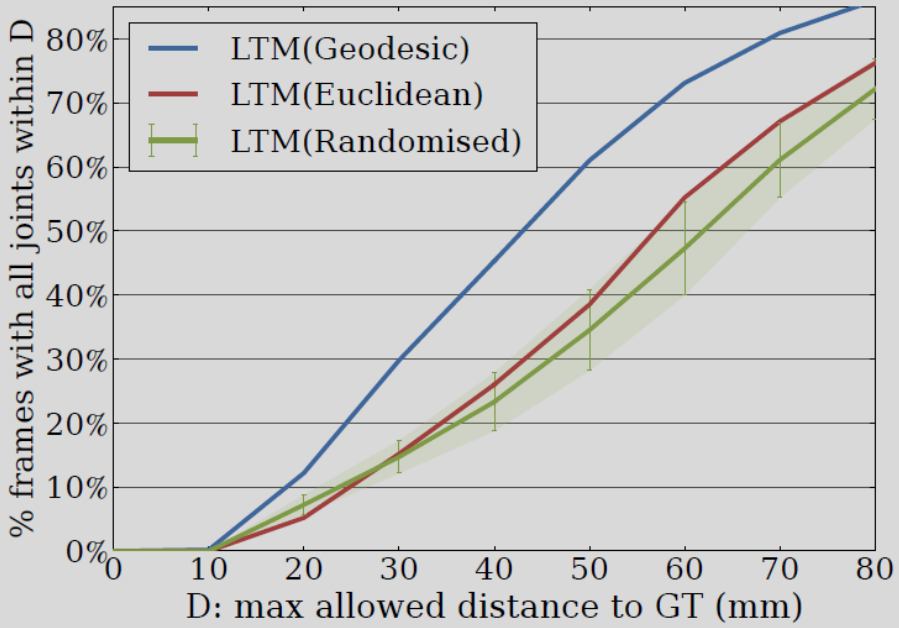
- 16 trees
- Stop: variance < 10mm



Experimental Results

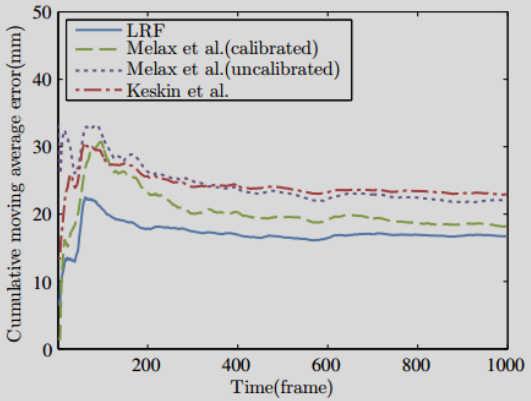
- Self Comparisons



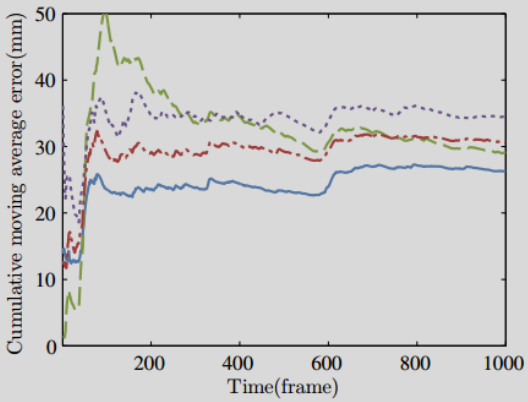


Experimental Results

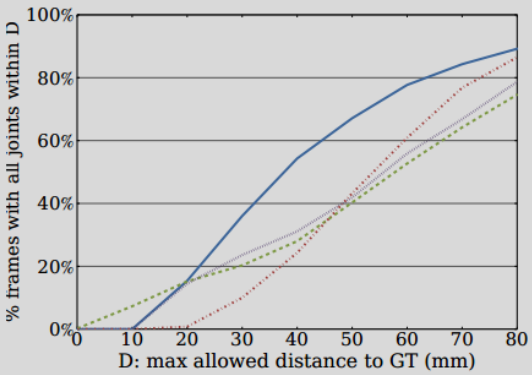
- Quantitative comparison with state-of-the-art methods



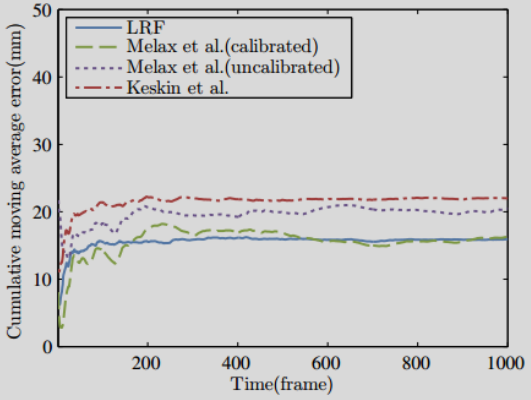
(a) Test sequence A (average error)



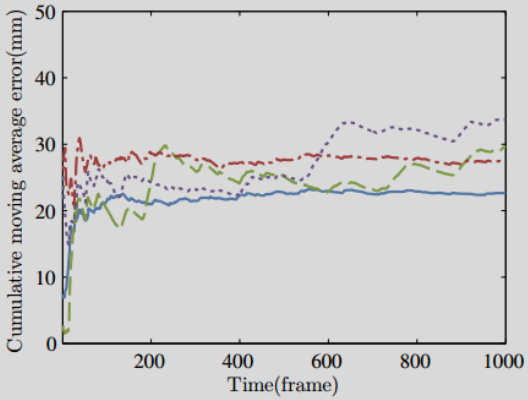
(b) Test sequence A(index tip)



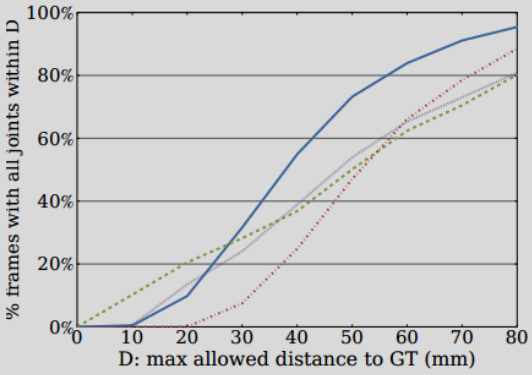
(c) Worst case accuracy [18] of sequence A



(d) Test sequence B(average error)



(e) Test sequence B(index tip)

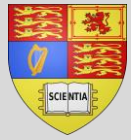


(f) Worst case accuracy [18] of sequence B

Experimental Results

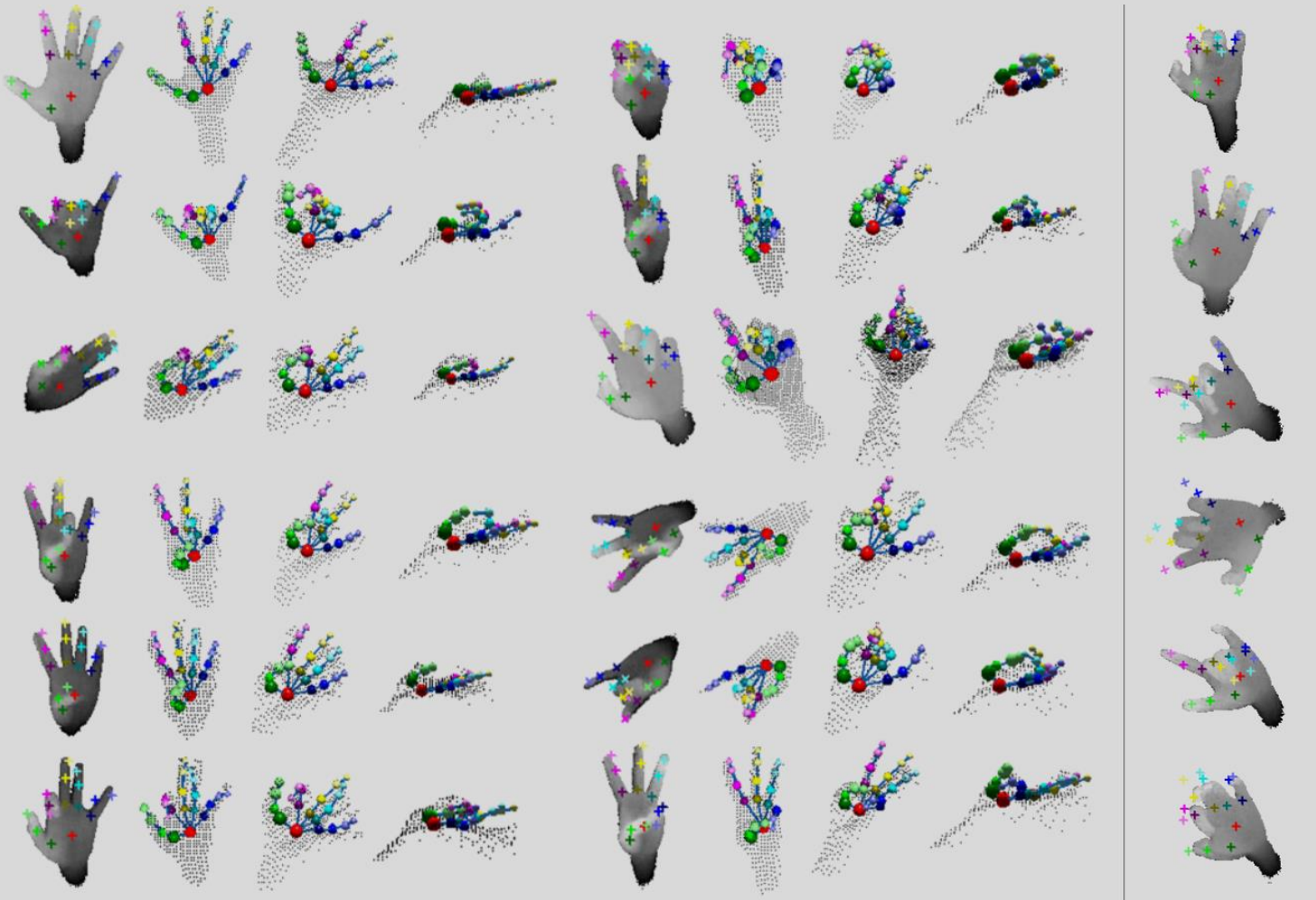
- Efficiency comparison

	Speed	Optimisation
Our method	62.5 fps	No
Keskin et al.(our impl.)	8.6 fps	No
Melax et al.	60 fps	CPU
FORTH	5 fps	GPU(GT 640)



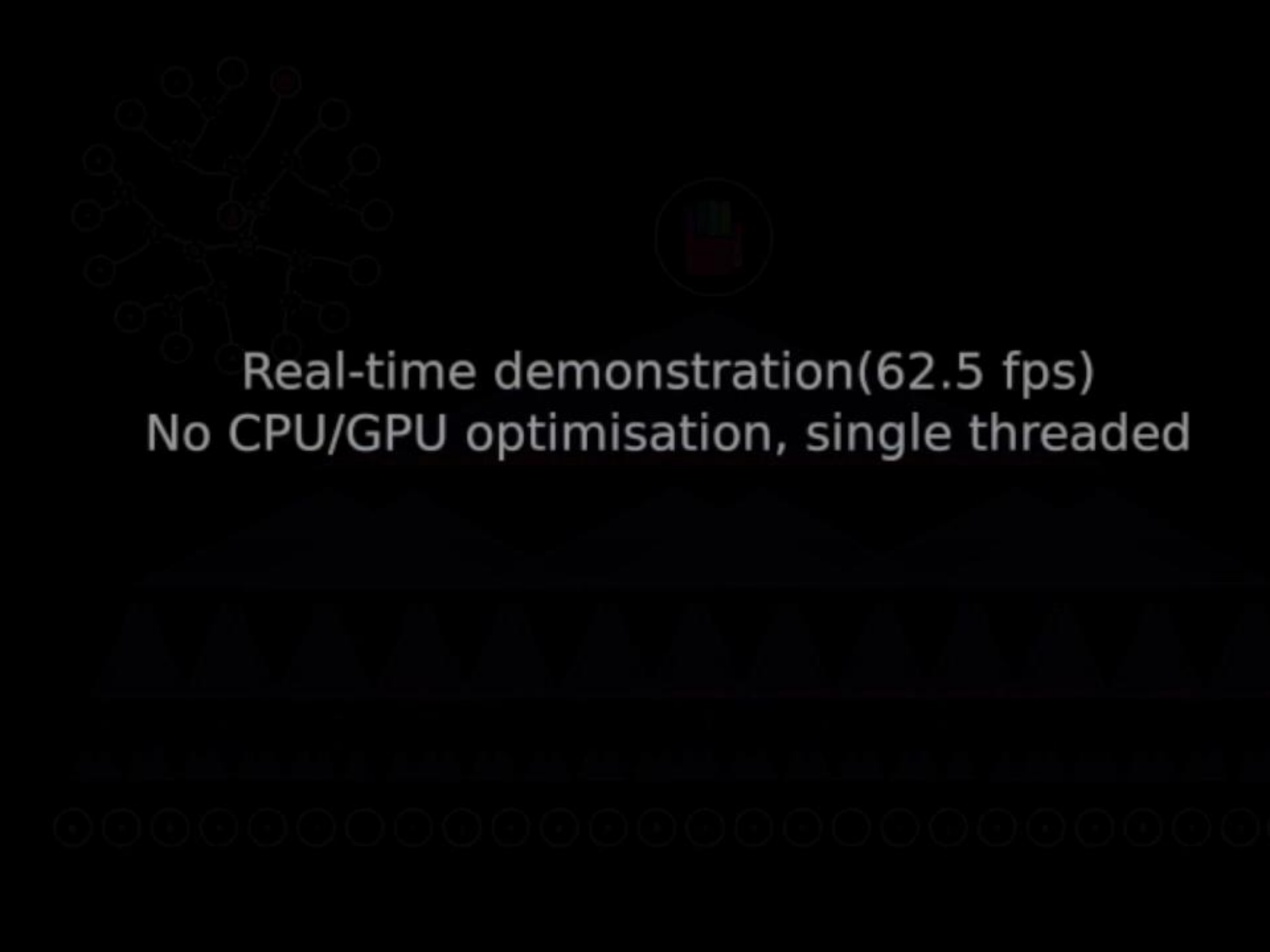
Imperial College London

Experimental Results



(a) Success case

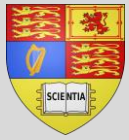
(b) Failure cases


Real-time demonstration(62.5 fps)
No CPU/GPU optimisation, single threaded

Imperial College
London

SD Gesture:
Static and Dynamic Gesture Estimation for Manipulating a Function-equipped AR Object

ISMAR 2015

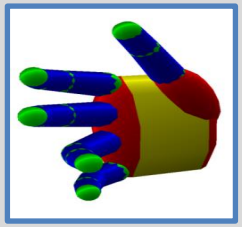


Summary

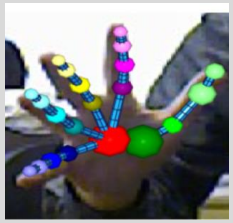
- The unique challenges in articulated hand pose estimation have been tackled.
  - Semi-supervised learning
  - Synthetic and real data discrepancy
  - High degree of freedom (16joints x 3d positions)
  - Kinematics
  - Real-time performance
- The proposed methods based on Decision Forests deliver real-time speed and high accuracy (frame-basis).
  - STR forest: 25Hz on Intel i7 PC without CPU/GPU optimization
  - Latent forest: 62.5 fps

Directions

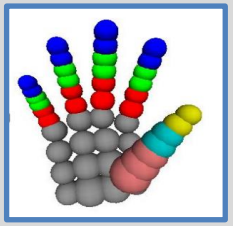
- Hybrid method that combines discriminative and generative
- Hand model
- Hand shape variations
- Hierarchical joint estimation



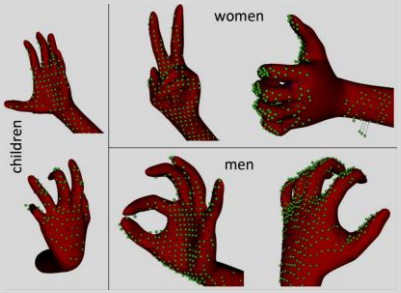
Oikonomidis et al.
BMVC'11



Tang et al.
ICCV'13, CVPR14



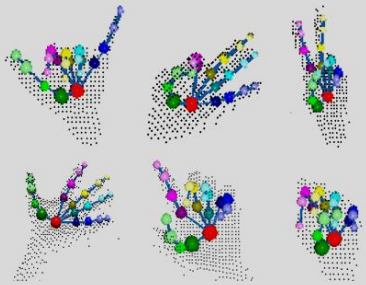
Qian et al.
CVPR'14



Khamis et al.
CVPR'15

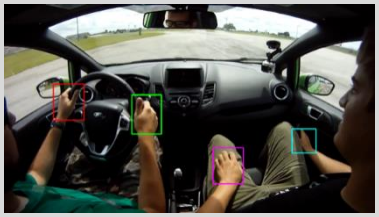
Directions

- Distant/Egocentric views
- Multiple hands
- Using RGB sequences
- Hands interacting objects
- Grasping (force estimation)

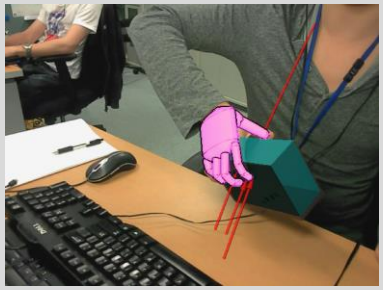


Tang et al. CVPR'14

Rogez et al. CVPR'15



VIVA Challenge http://cvrr.ucsd.edu/vivachallenge/

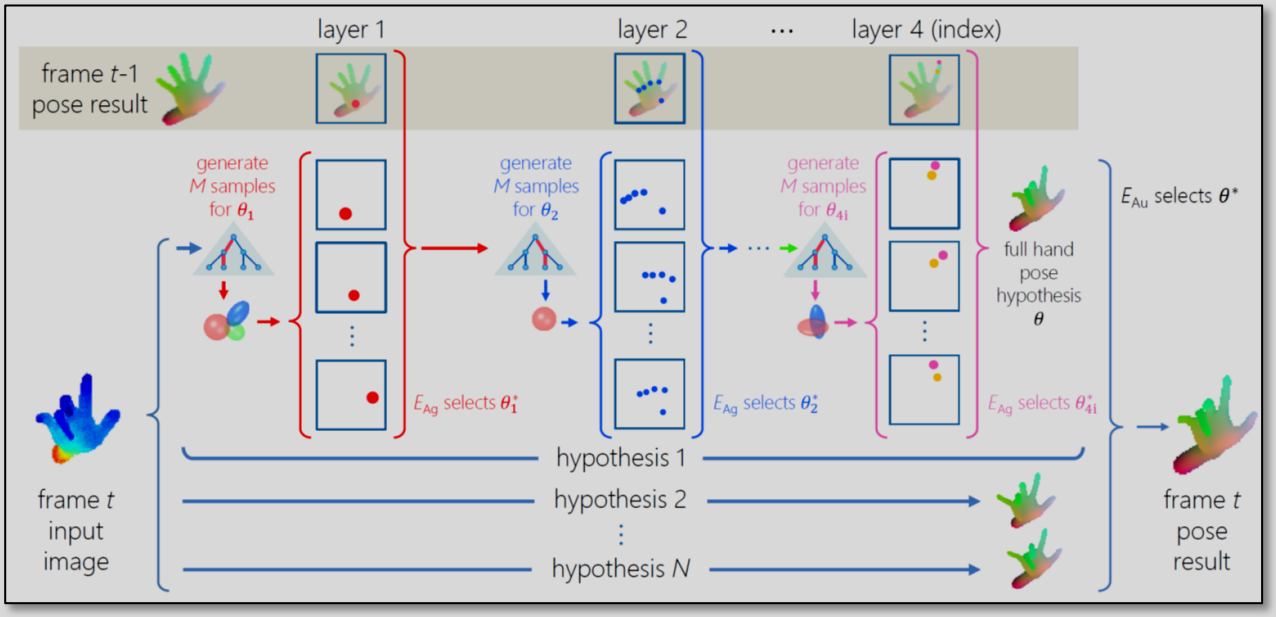


Pham et al. CVPR'15

Imperial College London

Hierarchical Sampling Optimization for Estimating Human Hand Pose (under review)

- A sequence of predictors organized into a kinematic hierarchy.
- Each predictor is conditioned on its ancestors, and generates a set of samples.
- Highly-efficient surrogate energy to select among samples.
- Via the full hierarchy, the partial pose samples are concatenated to a full-pose hypothesis.
- Finally the original full energy function selects the best result.

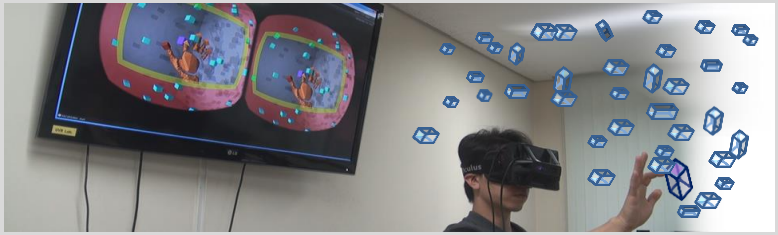


Resources

- Project pages @ Imperial College

http://www.iis.ee.ic.ac.uk/~dtang/hand.html

https://sites.google.com/site/fingergesture/



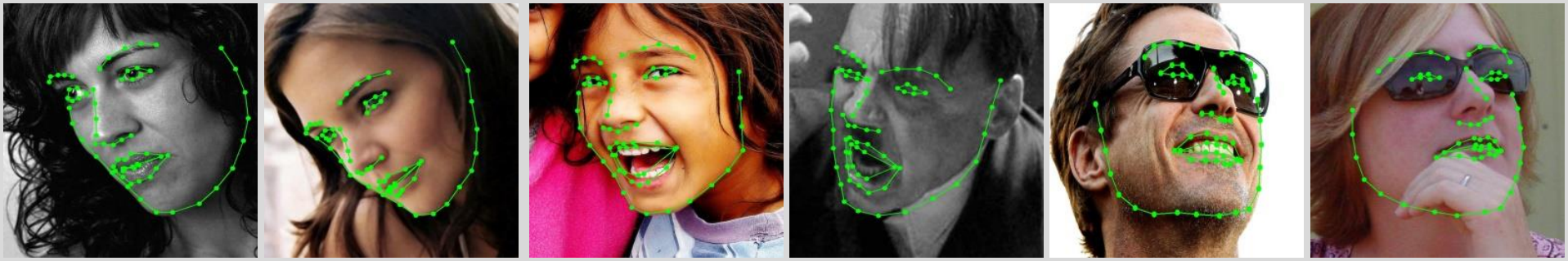
(IEEE TVCG15)

- CVPR15 workshop on observing and understanding hands in action (HANDS 2015): organised by T-K. Kim, G. Rogez, A. Agyros, J. Shotton, D. Ramanan, M. Trivedi et al.

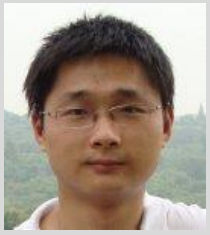
http://www.ics.uci.edu/~jsupanci/HANDS-2015/

Iterative Multi-Output Random Forests

Imperial College London



Unified Face Analysis by Iterative Multi-Output Random Forests



Xiaowei Zhao

T-K Kim

Wenhan Luo

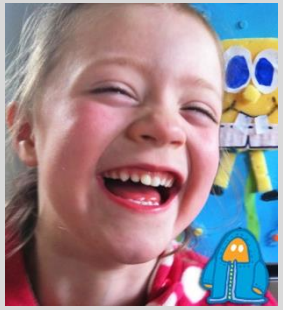
Imperial College London

CVPR 2014

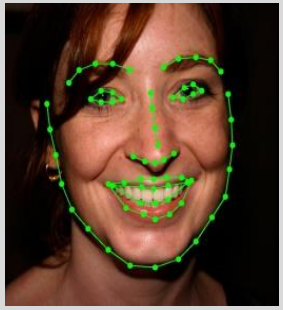
Imperial College London

Motivation

Pose Estimation



Expression Recognition

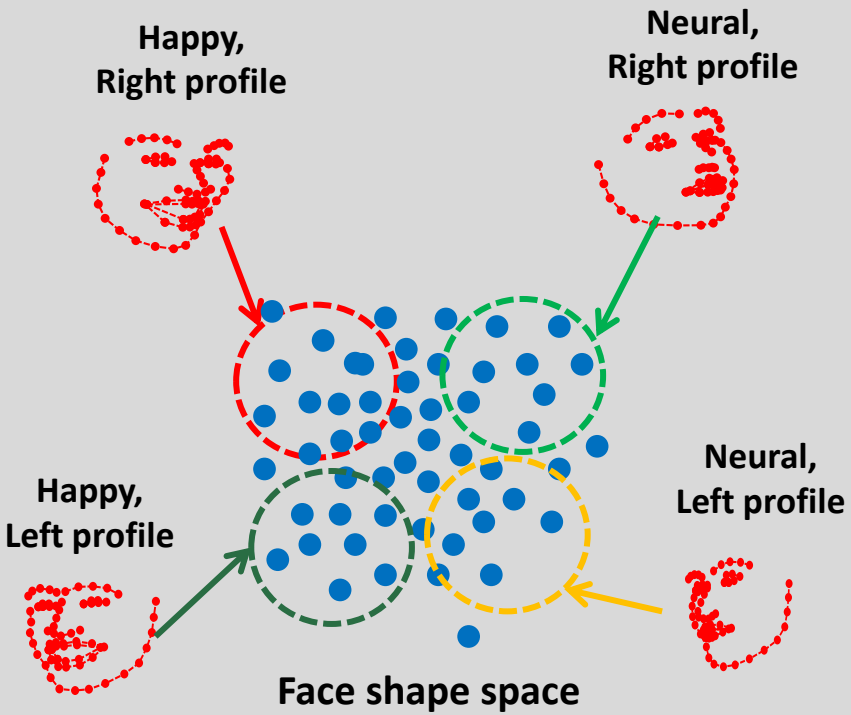


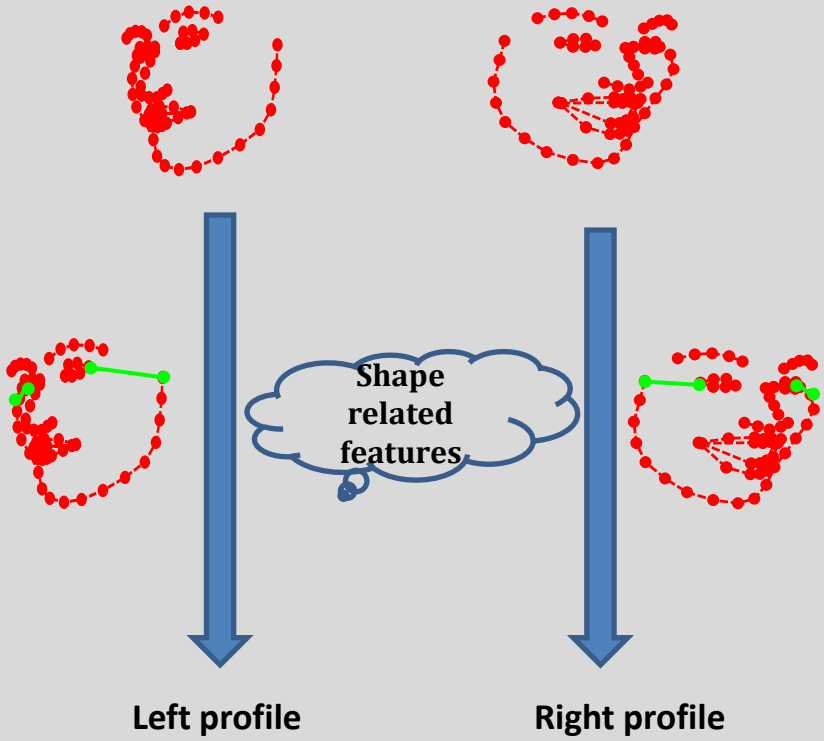
Landmark Localization

Face analysis tasks are *closely related* and *can help each other*.

Imperial College London

What is the relationship?





Facial attributes can provide *strong and compact prior* to constrain shape variation.

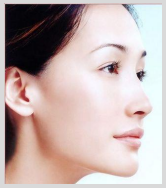
Facial attributes can be estimated by using *shape related features*.

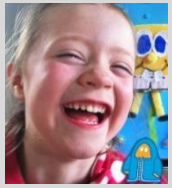
Imperial College London

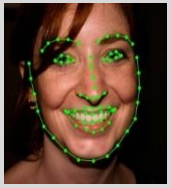
Main idea

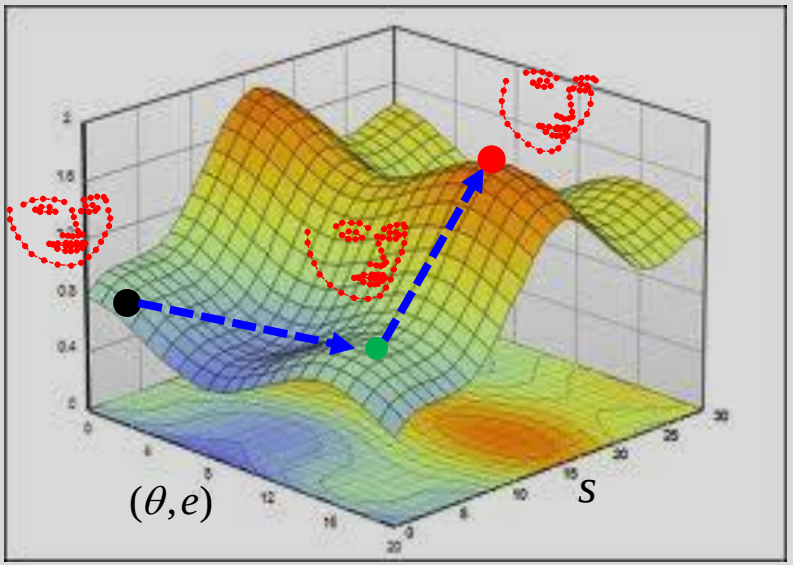
A *unified* framework to *jointly* address all face analysis tasks!

$$(\theta, e, s)^* = \arg\max_{\theta,e,s} p(\theta, e, s \mid I, b)$$

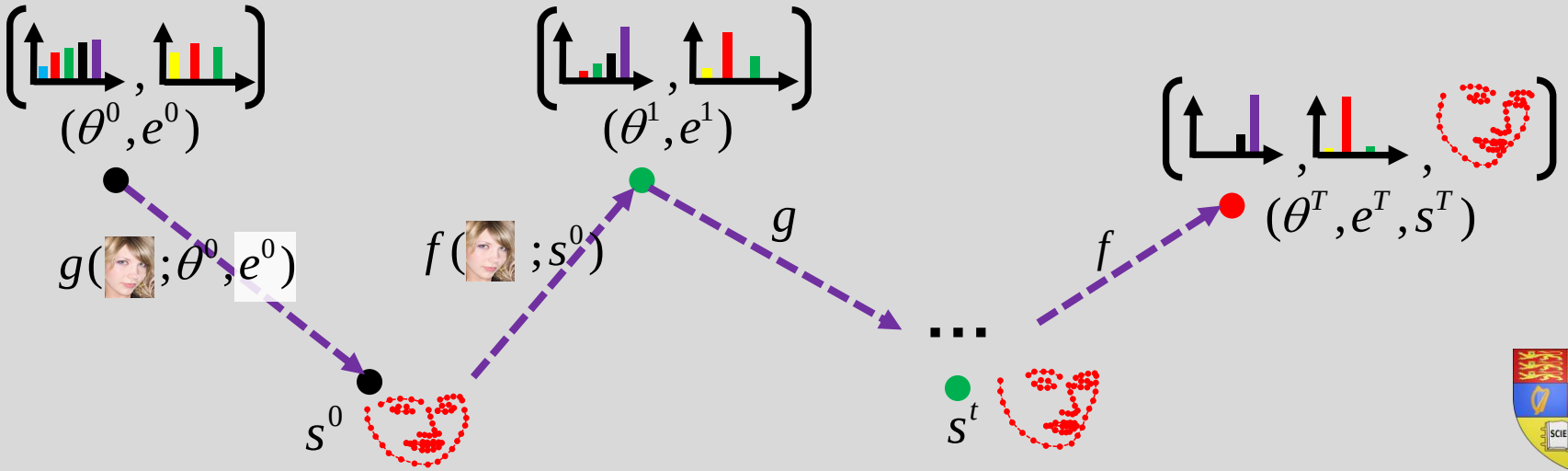
θ 

e 

s 

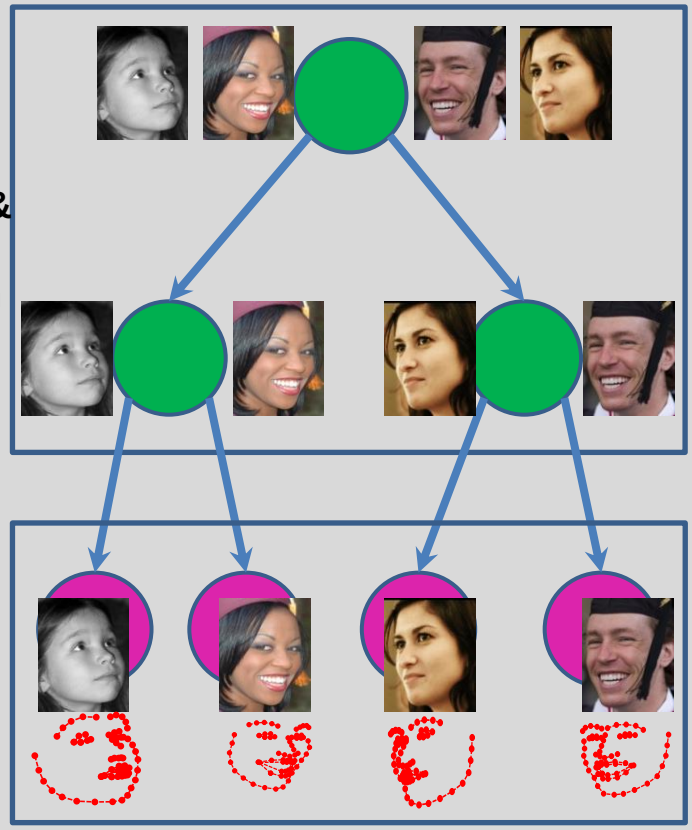


We *iteratively* exploit such *relationship* to approximate the optimal solution.



Imperial College London

Pose & Expression -> Landmark



One Tree in Hierarchical Face Analysis Forest

The shape prior is automatically learned by the hierarchical forest.

Split function:

- **simple patch comparison test**

Hybrid quality function:

$$Q_{face} = \alpha Q_\theta + (1-\alpha)\beta Q_e + (1-\alpha)(1-\beta)Q_s$$

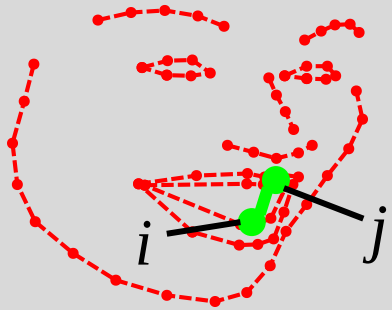
α β **control weights of** Q_θ, Q_e, **and** Q_s

adaptively switch according to the purity of head pose and expression.

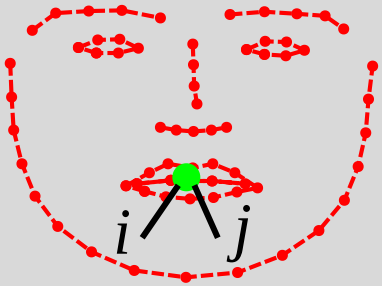
$$Q_s = -\sum_{k=1}^{K} \frac{\sum_i p(c_k \mid P_i)}{|P|} \log\left(\frac{\sum_i p(c_k \mid P_i)}{|P|}\right)$$

Imperial College London

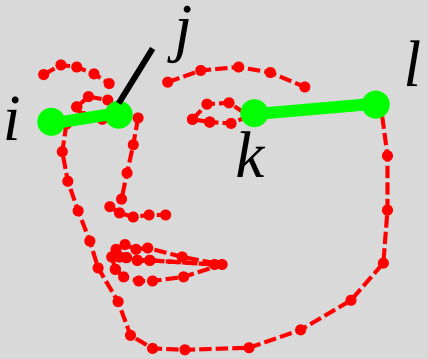
Landmark -> Pose & Expression



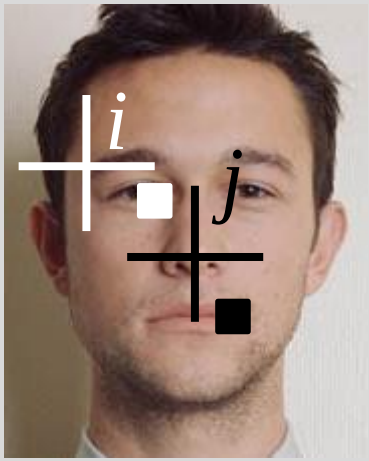
(a)



(b)



(c)



(d)

The shape information is automatically encoded by the shape-related features, which are fed to the forest as richer feature set.

Shape residual regression

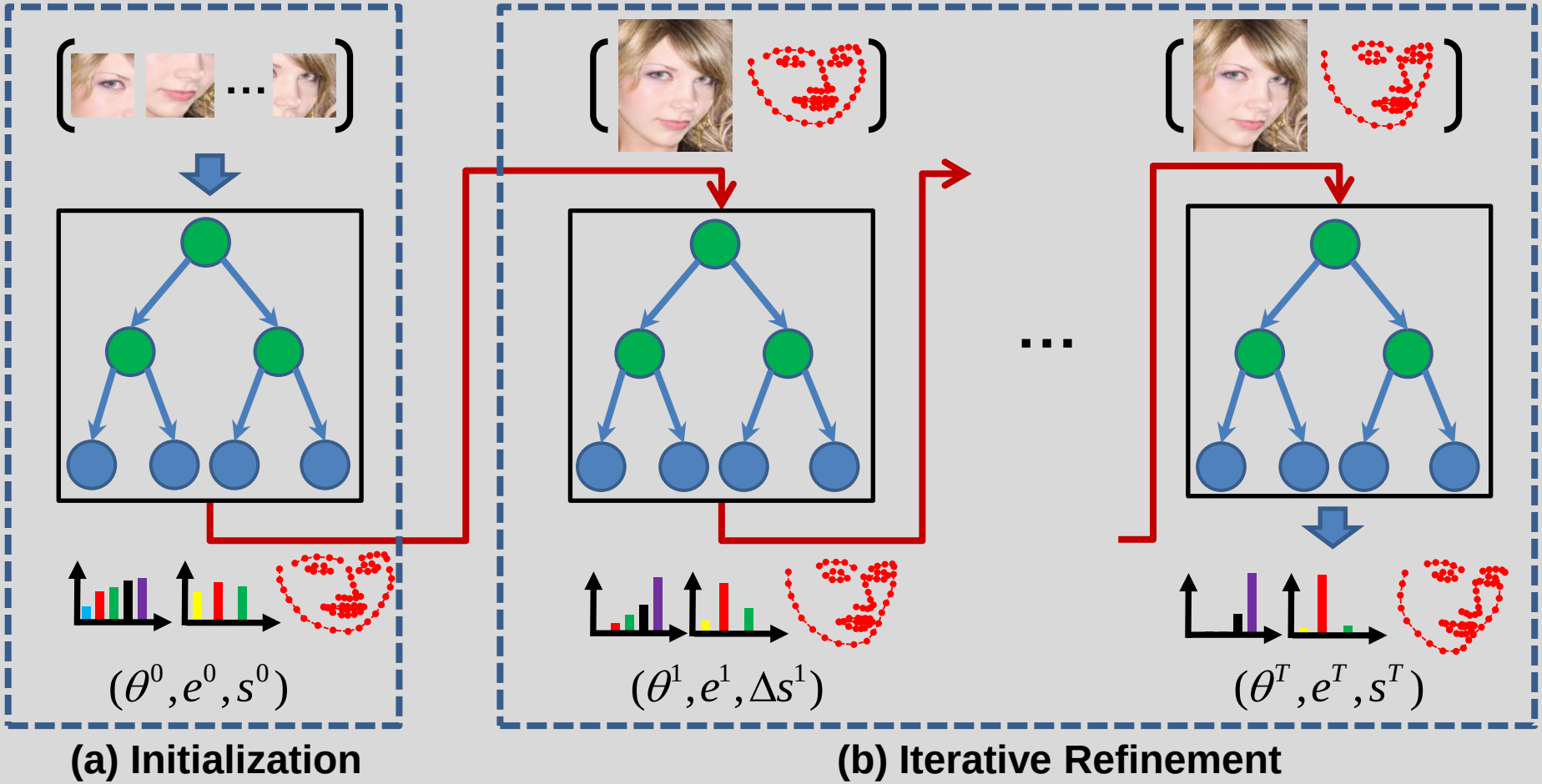
$$Q_s = -\Psi(\sum\nolimits_l(\Delta s^{t-1})) - \Psi(\sum\nolimits_r(\Delta s^{t-1}))$$

$$\Psi(\bullet) = \log(\det(\bullet))$$

$\sum$ is the covariance matrix of shape residue

Imperial College London

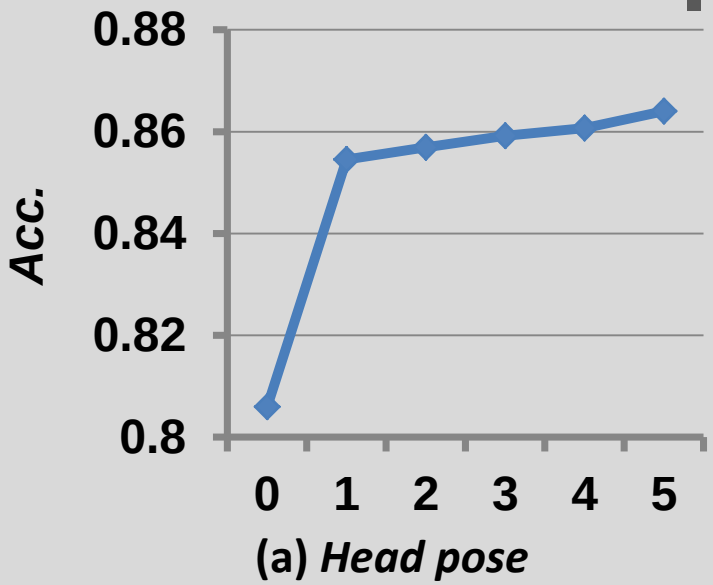
Implementation: iMORF



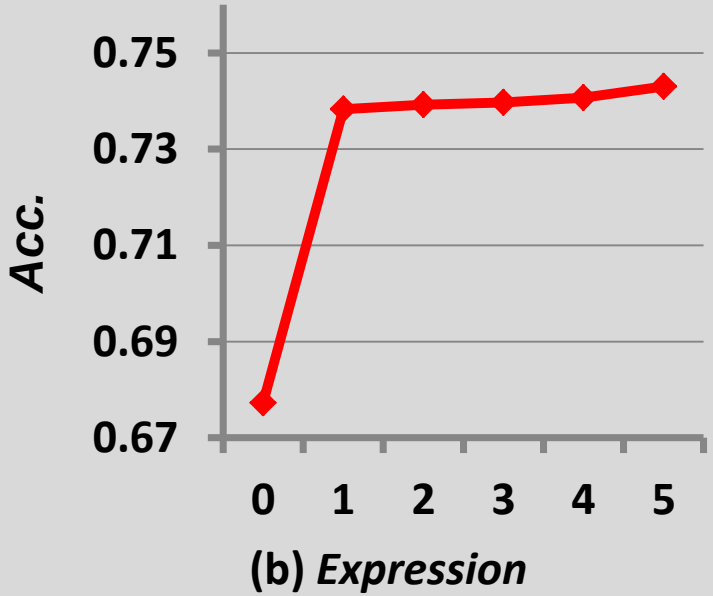
Our idea is naturally implemented by a novel *iterative Multi-Output Random Forests (iMORF)* algorithm.

Imperial College London

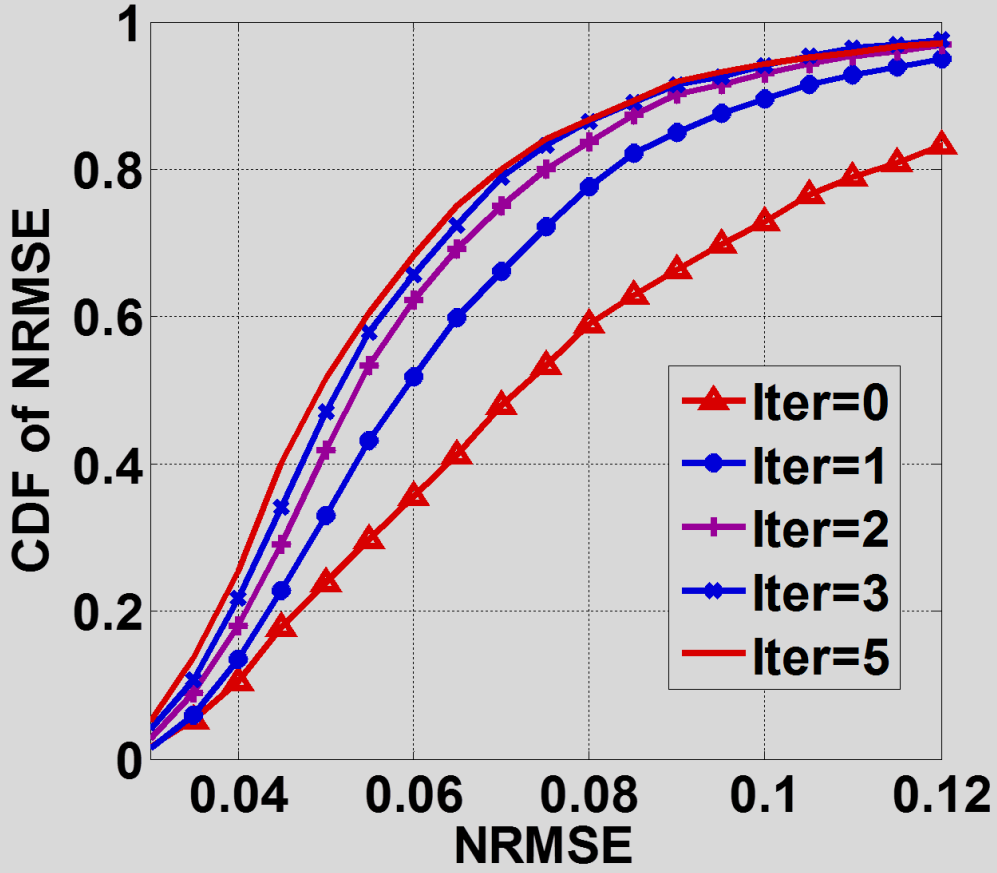
Iterative Performance Improvement



(a) *Head pose*



(b) *Expression*



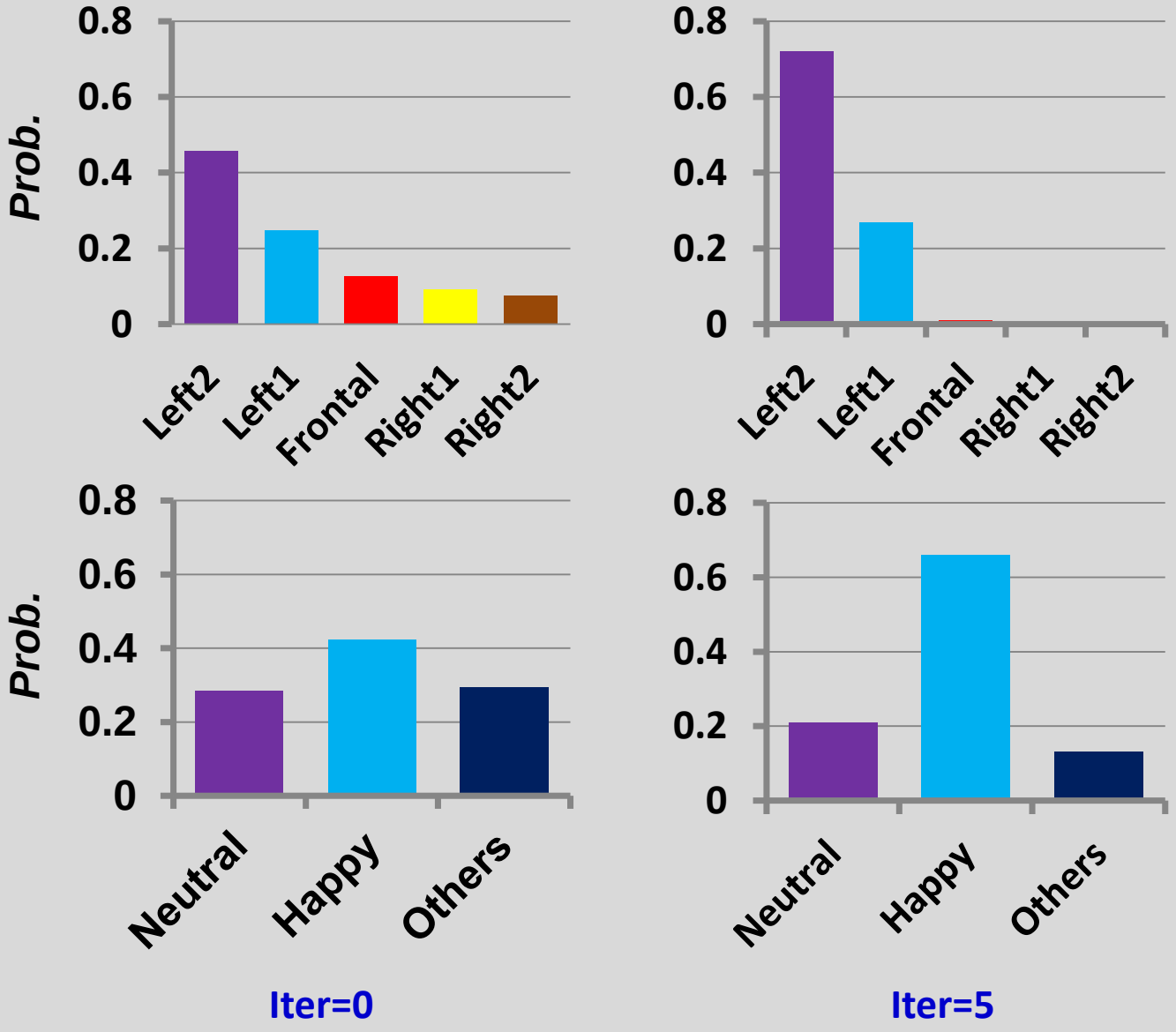
(c) *Landmark positions*

Imperial College London

Estimated probability distribution

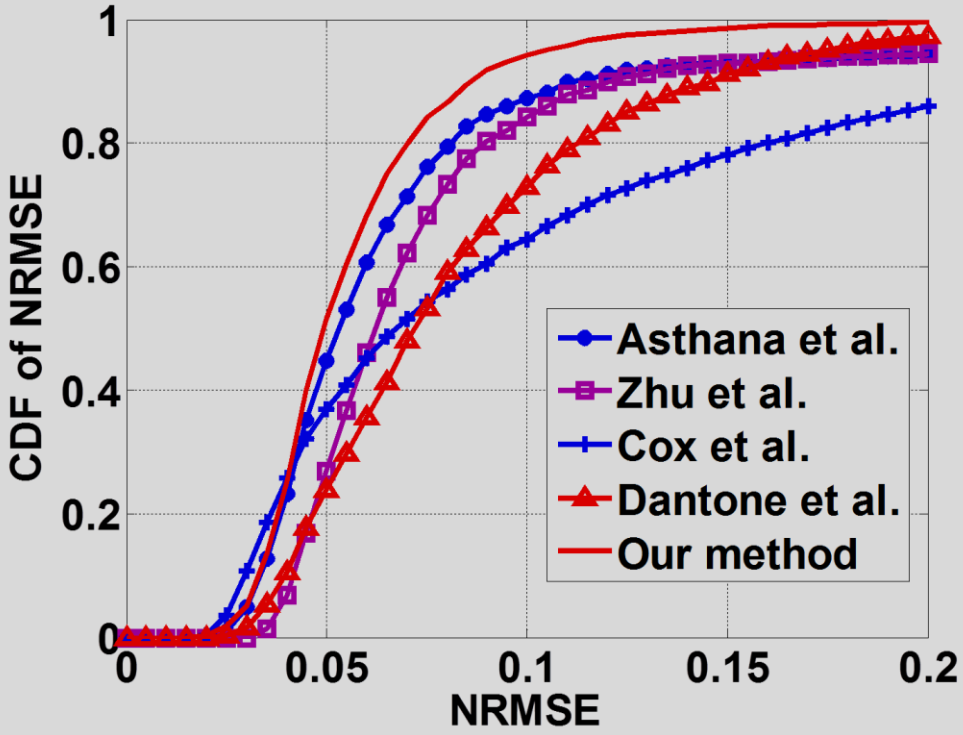
(a) *Head pose*

(b) *Expression*



Imperial College London

Comparison with state of the arts



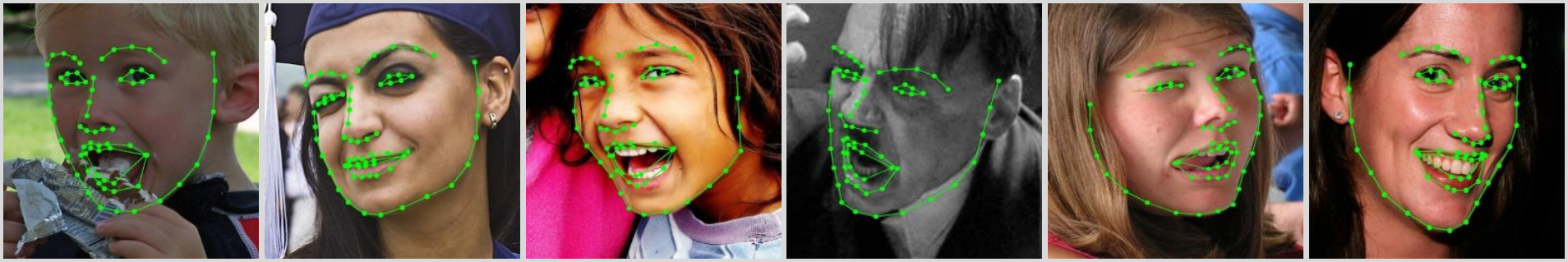
Head pose estimation	
Asthana et al.	80. 38%
Cox et al.	47.09%
Zhu et al.	76.65%
Fanelli et al.	80.59%
Our method	**86.40%**

Expression recognition	
LBP+SVM	83.87%
SIFT+SVM	86.39%
Gabor+SVM	88.61%
CSPL	89.89%
Our method	**90.04%**

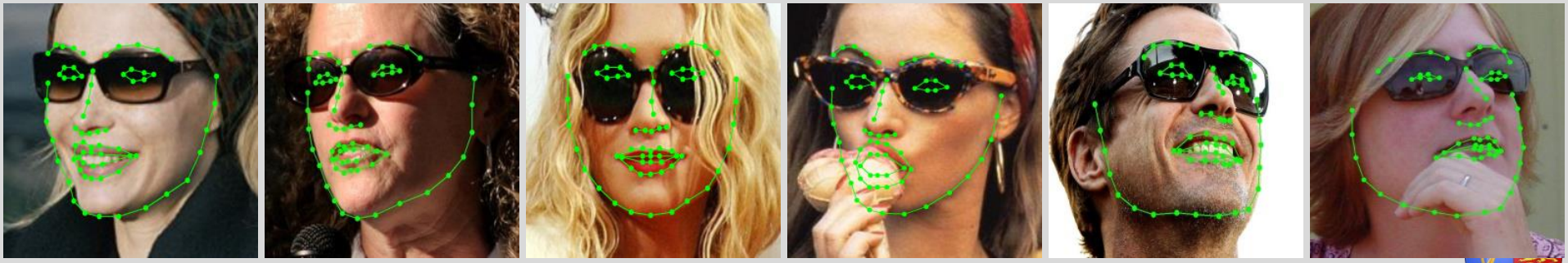
Imperial College London

Results on challenging examples

Pose variation

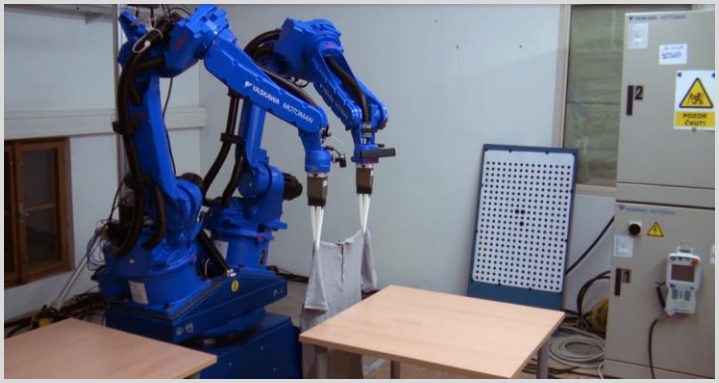


Expression



Occlusion

Active Random Forest



A. Doumanoglou, T-K. Kim, X. Zhao, S. Malassiotis, **Active Random Forests: An application to Autonomous Unfolding of Clothes**, ECCV, 2014

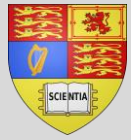
A. Doumanoglou, A. Kargakos, T-K. Kim, S. Malassiotis, **Autonomous Active Recognition and Unfolding of Clothes using Random Decision Forests and Probabilistic Planning**, ICRA, 2014 (KUKA best service robotics paper award).

Autonomous Active Recognition and Unfolding of Clothes using Random Decision Forests and Probabilistic Planning

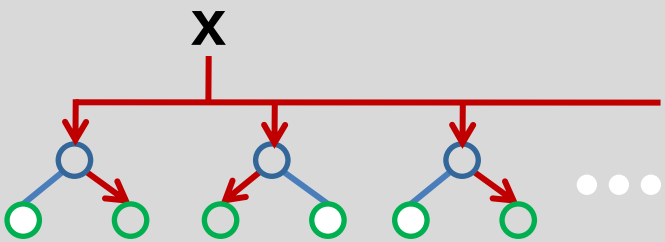
Andreas Doumanoglou, Andreas Kargakos, Tae-Kyun Kim, Sotiris Malassiotis

ICRA, Hong Kong, 2014

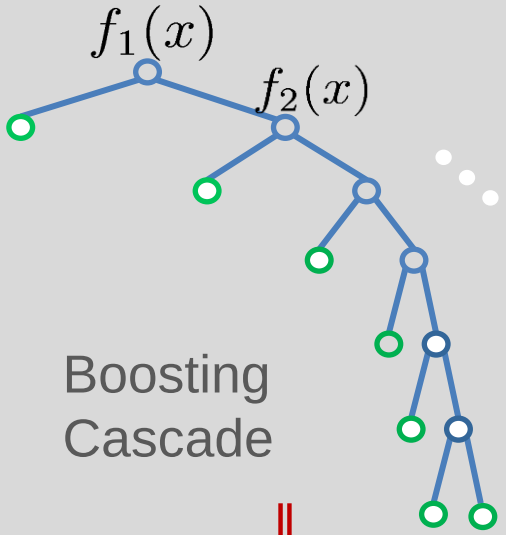
Summary



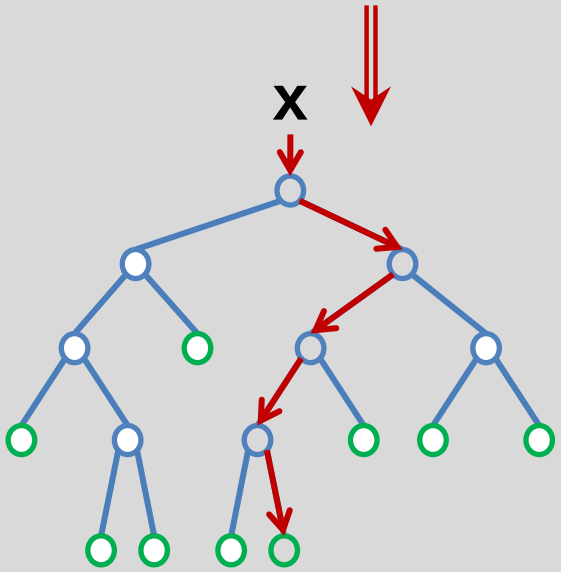
Basics and Motivations



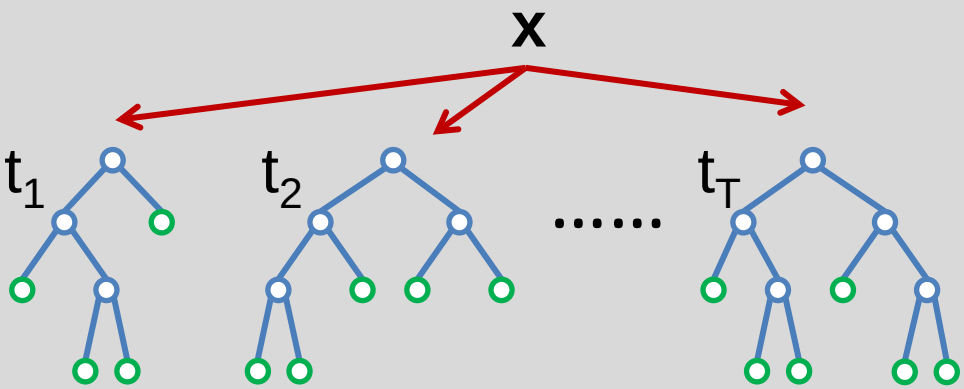
Boosting



Boosting Cascade

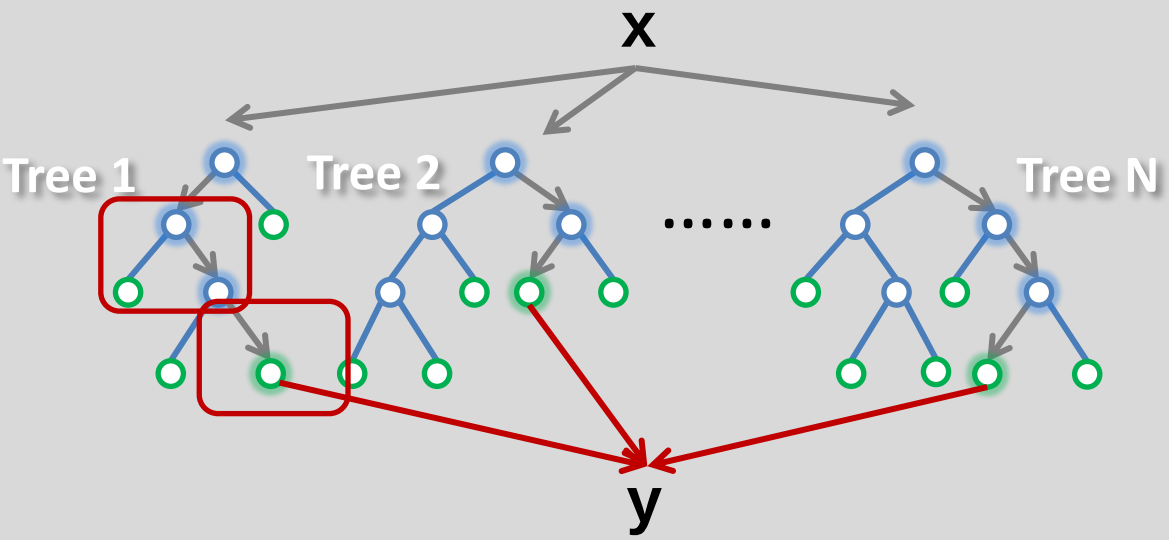


Tree structure boosting

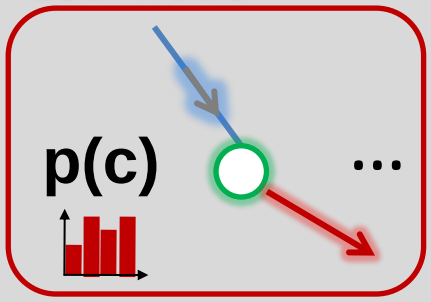


Randomised Decision Forest

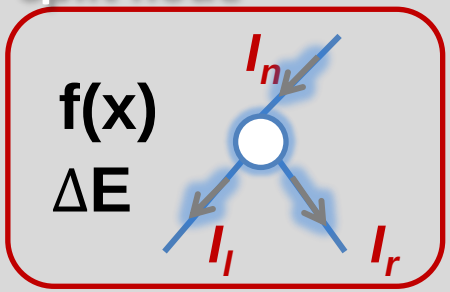
Case Studies @ ICL



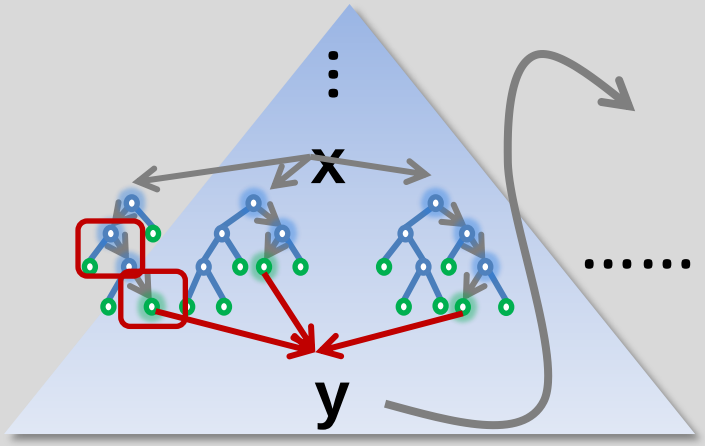
Leaf node



Split node

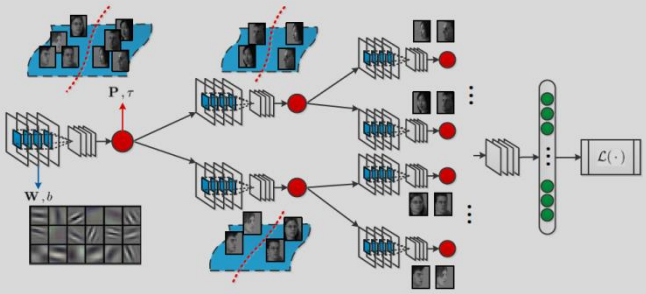


Architecture

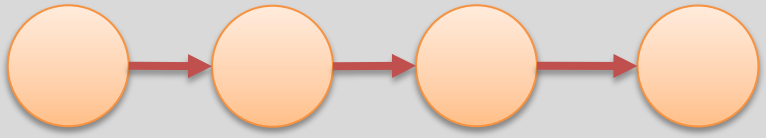


Directions on RF

- Deep learning + RF
  - learning representation, conditional computing, efficiency
- Active RF classifiers
  - action as a learning parameter
- Dynamic + Static
- State transitions
  - Captured under a powerful classifier framework



ICRA14, ECCV14



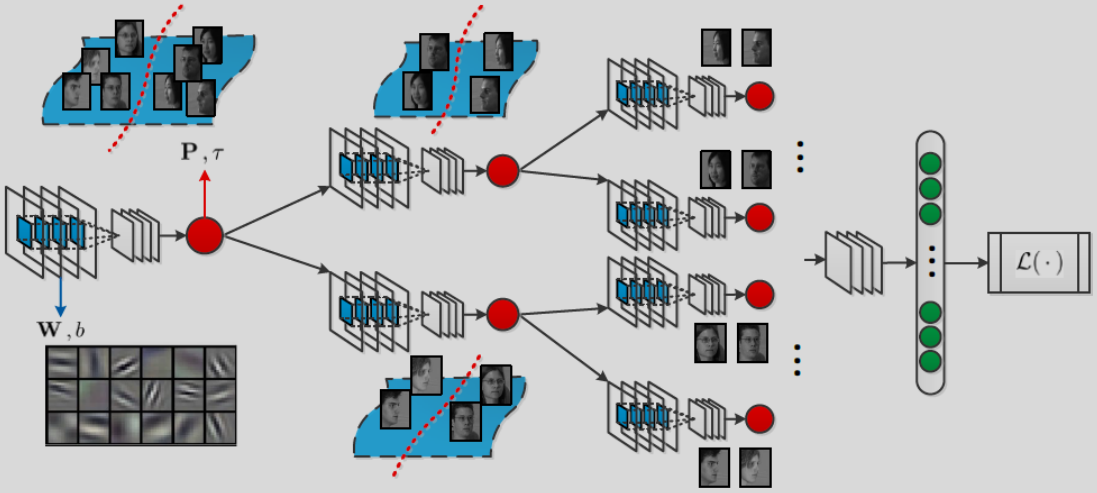
Conditional Convolutional Neural Network

(under review)

- Joint object ftn.

$$\mathcal{L} = \sum_n \mathcal{J}(\boldsymbol{x}_n, y_n) + \beta \sum_i \sum_j \mathcal{L}^{(i,j)}$$

the first term is the softmax loss for classification and the second is the nodewise loss for clustering.



- Conditional forward ftn.

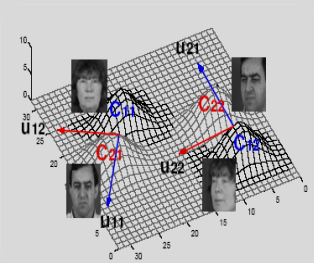
$$\begin{cases} \boldsymbol{X}_n^{(i+1,2j)} & = \mathbb{1}(\varphi(\widetilde{\boldsymbol{X}}_n^{(i,j)}) \geq 0) \cdot \widetilde{\boldsymbol{X}}_n^{(i,j)} \\ \boldsymbol{X}_n^{(i+1,2j+1)} & = \mathbb{1}(\varphi(\widetilde{\boldsymbol{X}}_n^{(i,j)}) < 0) \cdot \widetilde{\boldsymbol{X}}_n^{(i,j)} \end{cases}$$

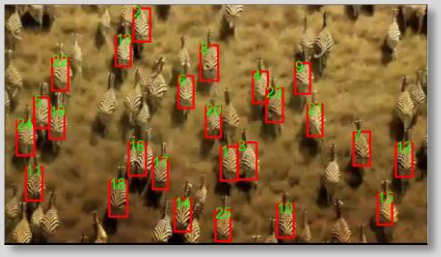
$$\widetilde{\boldsymbol{X}}_n^{(i,j)} = \sigma(\boldsymbol{W}^{(i,j)} * \boldsymbol{X}_n^{(i,j)} + b^{i,j})$$

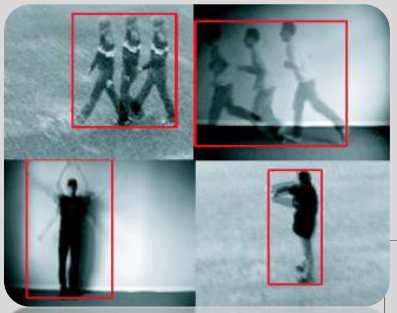
- Optimisation is via back propagation with Stochastic Gradient Descent. Each node has 4 parameters – P(i;j), τ(i;j), W(i;j) and b(i;j).

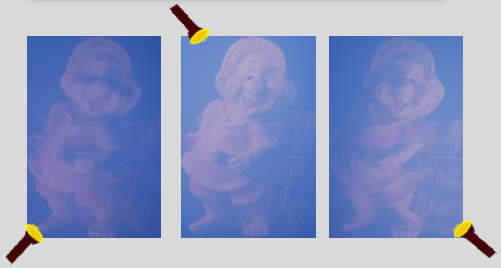
Imperial College London

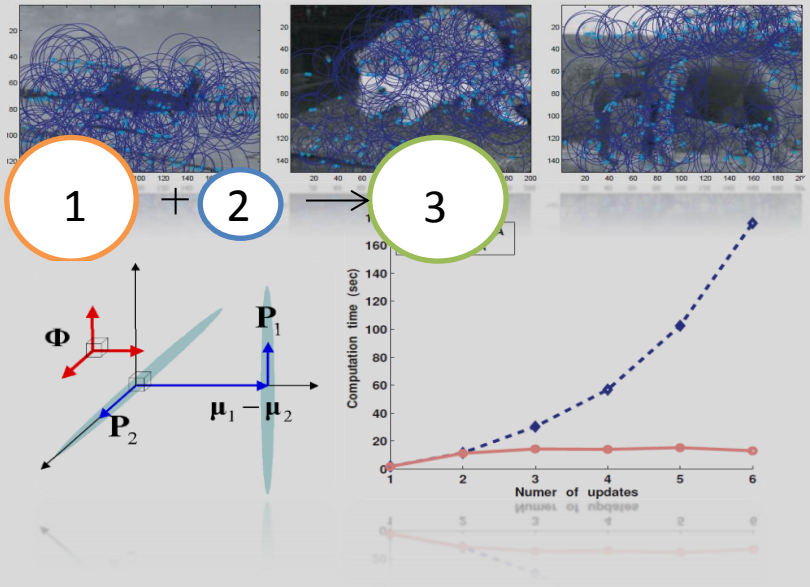
Other research topics @ ICL

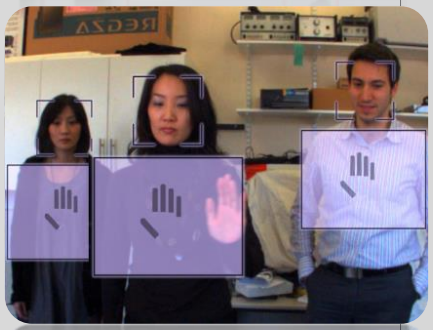








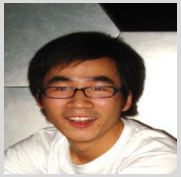


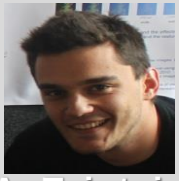


Imperial College London

Thanks to

Imperial College London


W. Luo


A. Tsiotsios


Y. Pei

UNIVERSITY OF SURREY


J. Kittler

TOSHIBA Leading Innovation


B. Stenger

Microsoft Research

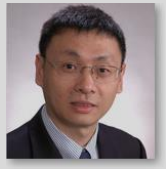

J. Shotton


A. Doumanoglou

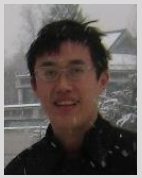

H.J. Chang


C. Xiong

NUS National University of Singapore


S. Yan


T-H. Yu


Y. Chen

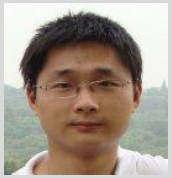

R. Kouskouridas


A. Tejani

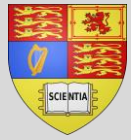

D.Tang


I. Budvytis


R.Cipolla


X. Zhao


A. Davison




Questions?
http://www.iis.ee.ic.ac.uk/ComputerVision