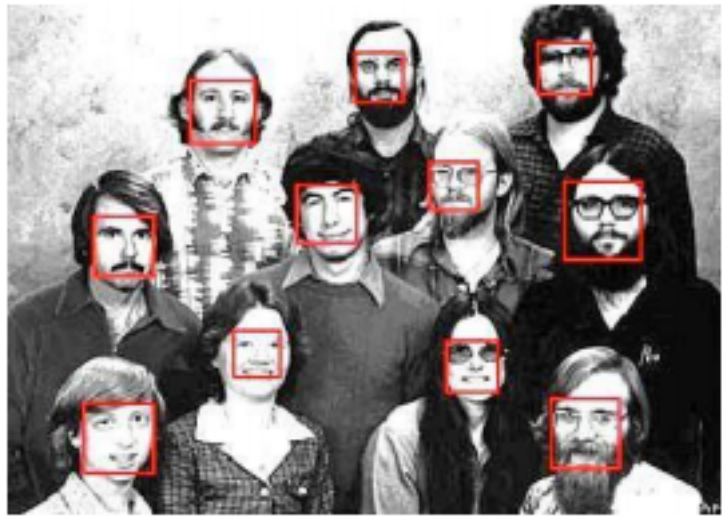
Lecture 11: Data Mining III
In Computer Vision Systems

http://cs.nju.edu.cn/yuy/course_dm14ms.ashx

Face detection

find faces in a given photo

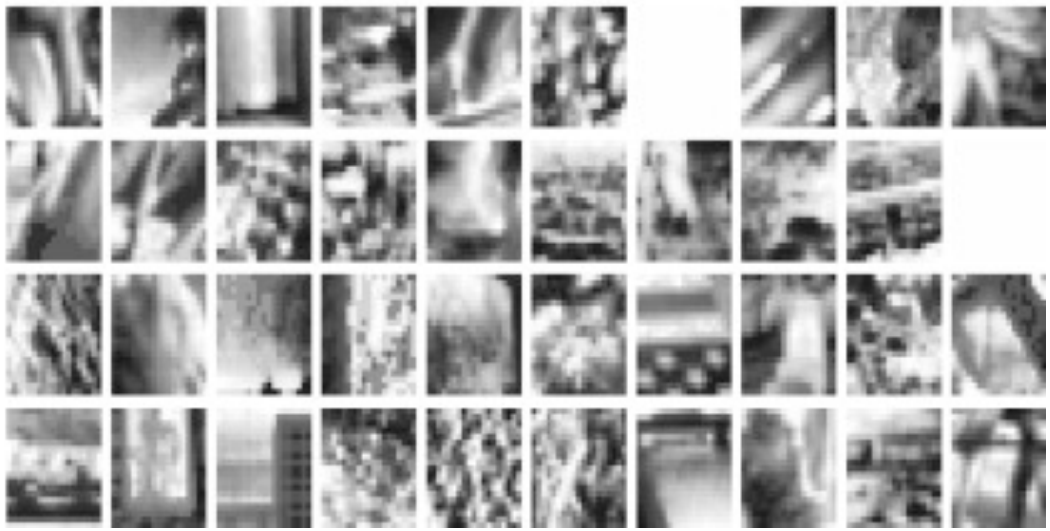


sliding window

What is a face?

What is a face?

What is a face?

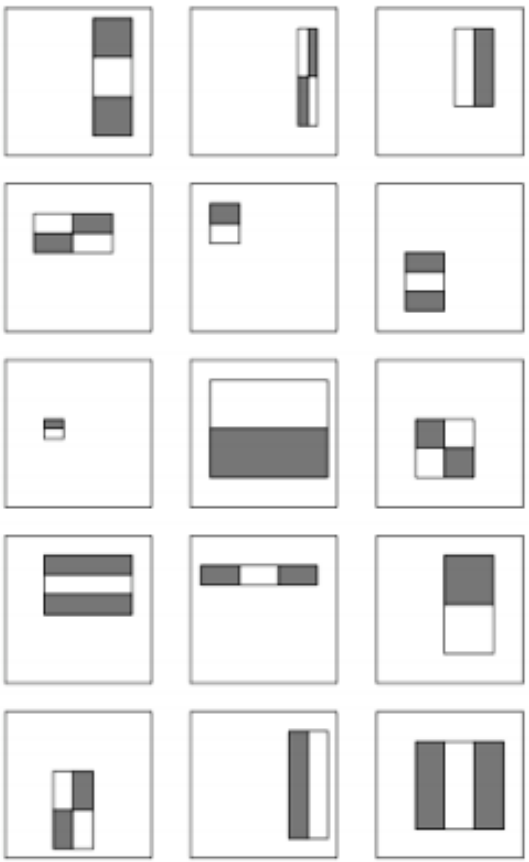


What is a face?

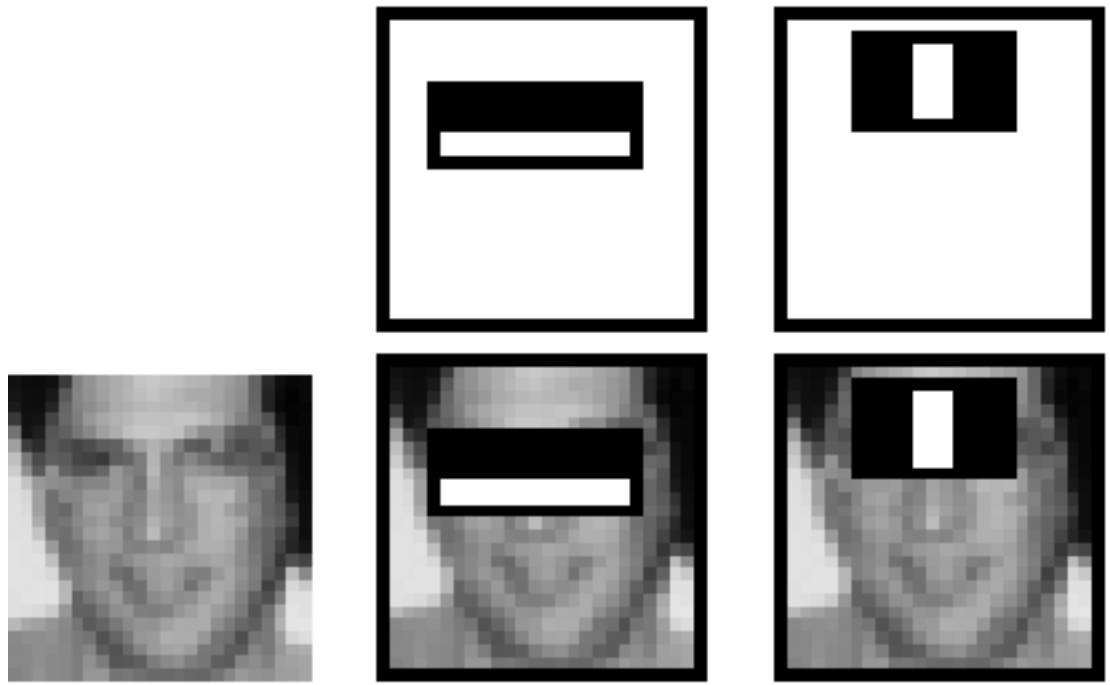
similar to positive face
rather than negative face

Viola&Jones face features [IJCV'01]

features: simple templates



for each sliding window apply temples to calculate features

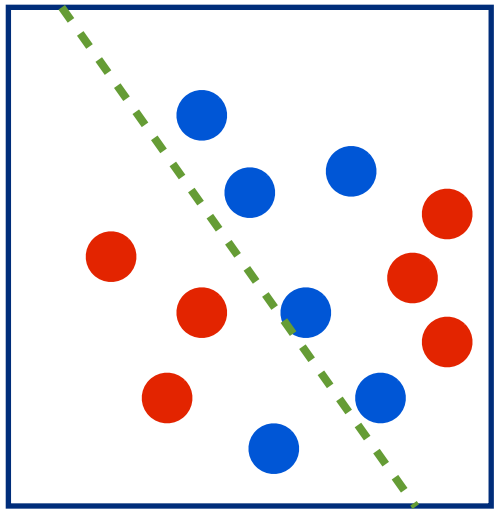


conceptually forms a vector:

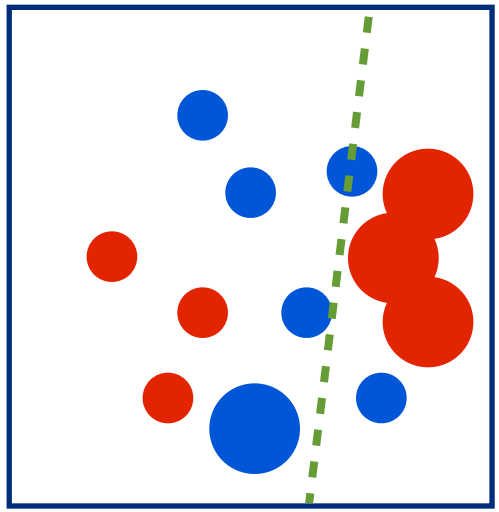
(200, 50, 90, ...)

AdaBoost

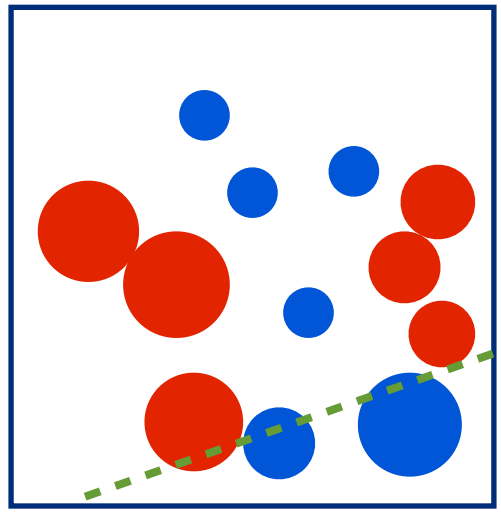
classifier 1



classifier 2



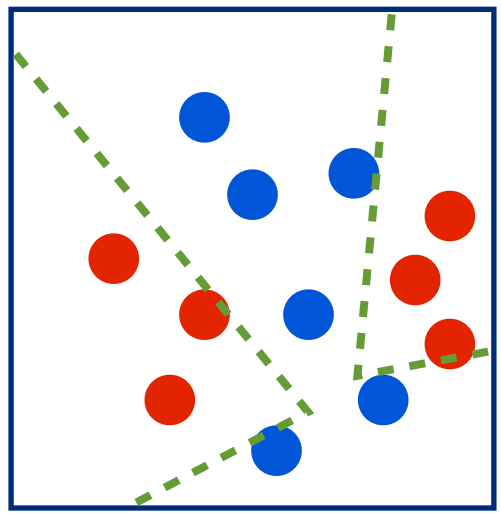
classifier 3



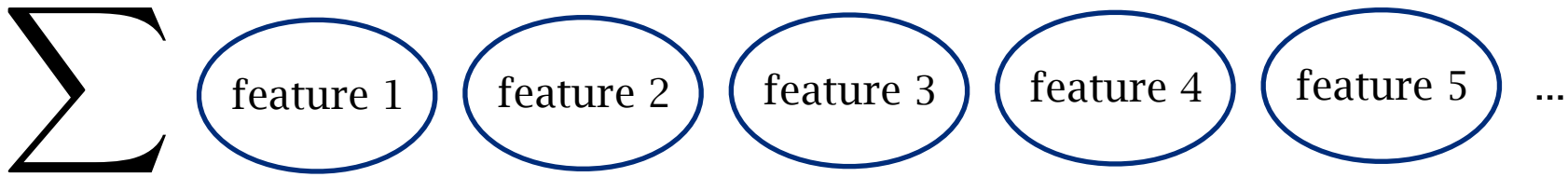
In V&J's system, each classifier is one feature

AdaBoost selects a small subset of features

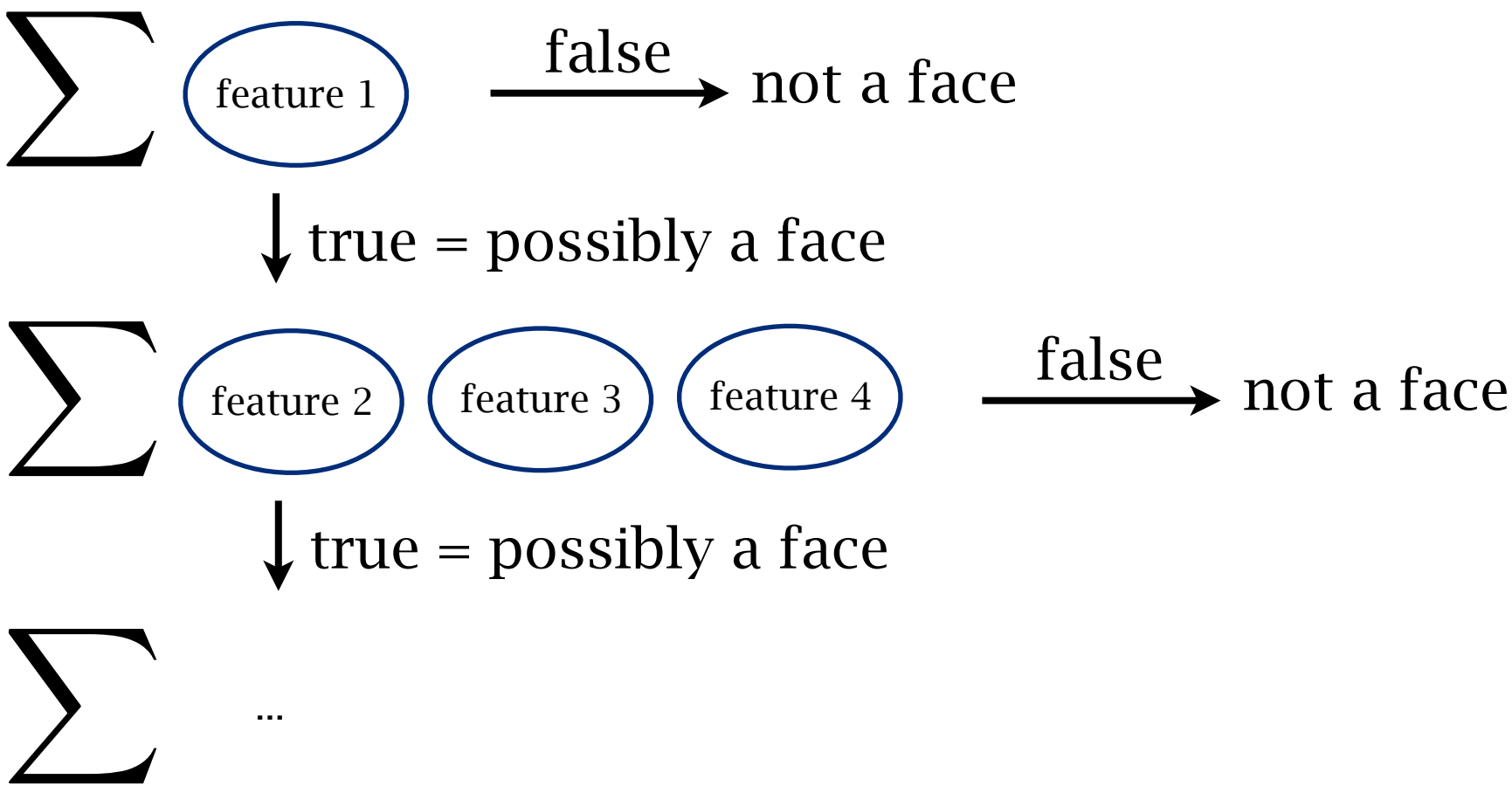
final classifier



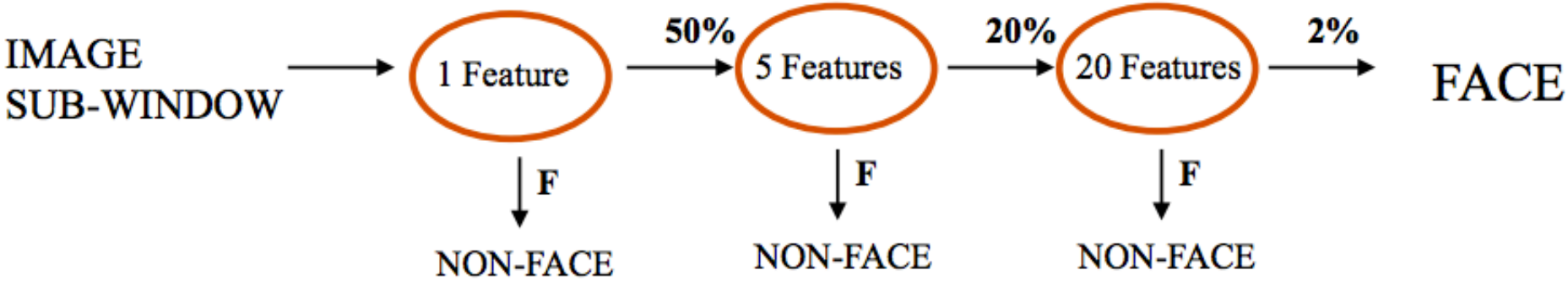
Viola&Jones face features [IJCV'01]

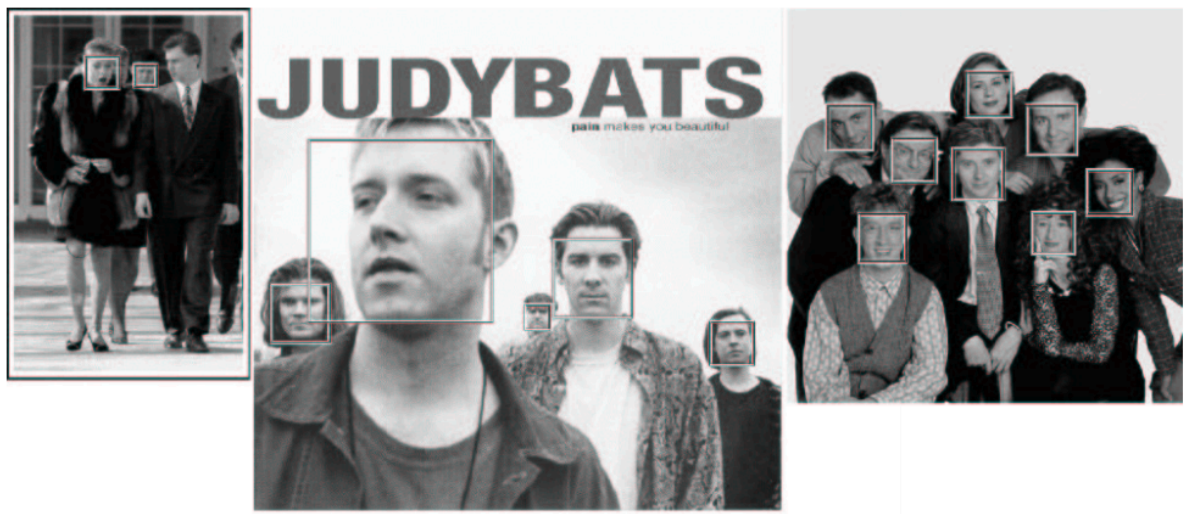


face area is small



Viola&Jones face features [IJCV'01]





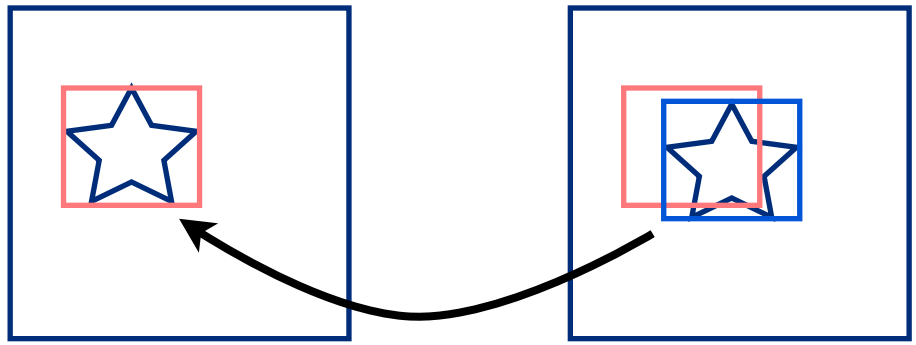
"15 times faster" than a state-of-the art while keeping the accuracy"

The data-driven approaches

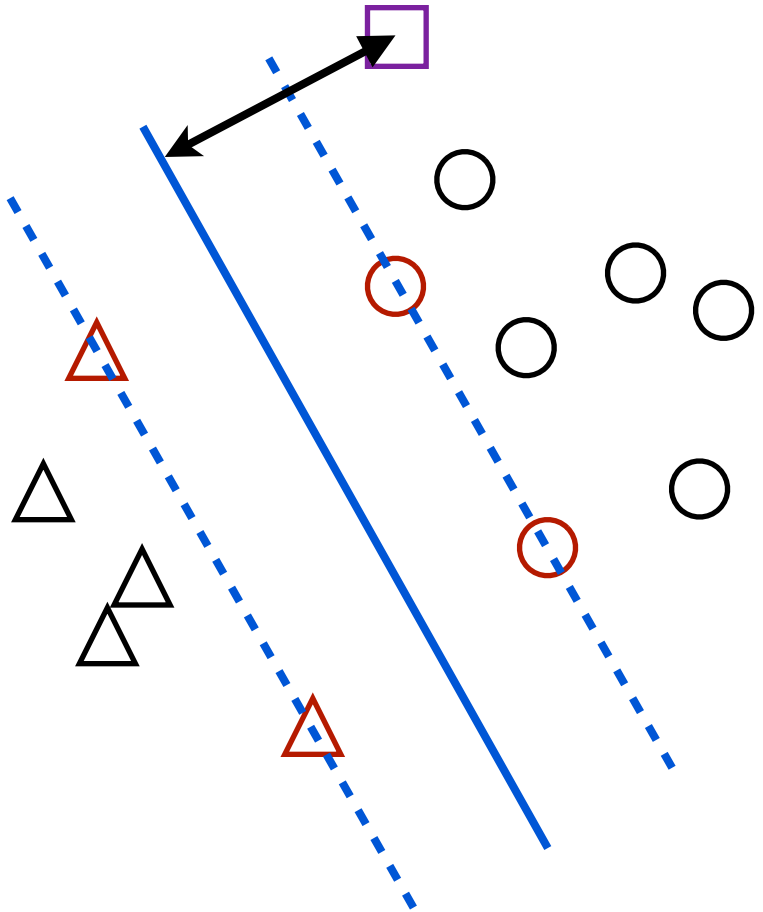
Viola&Jonse's work does not only result an efficient face detector, but also activate the data-driven approaches in CV.

Object tracking

calculation of similarity?



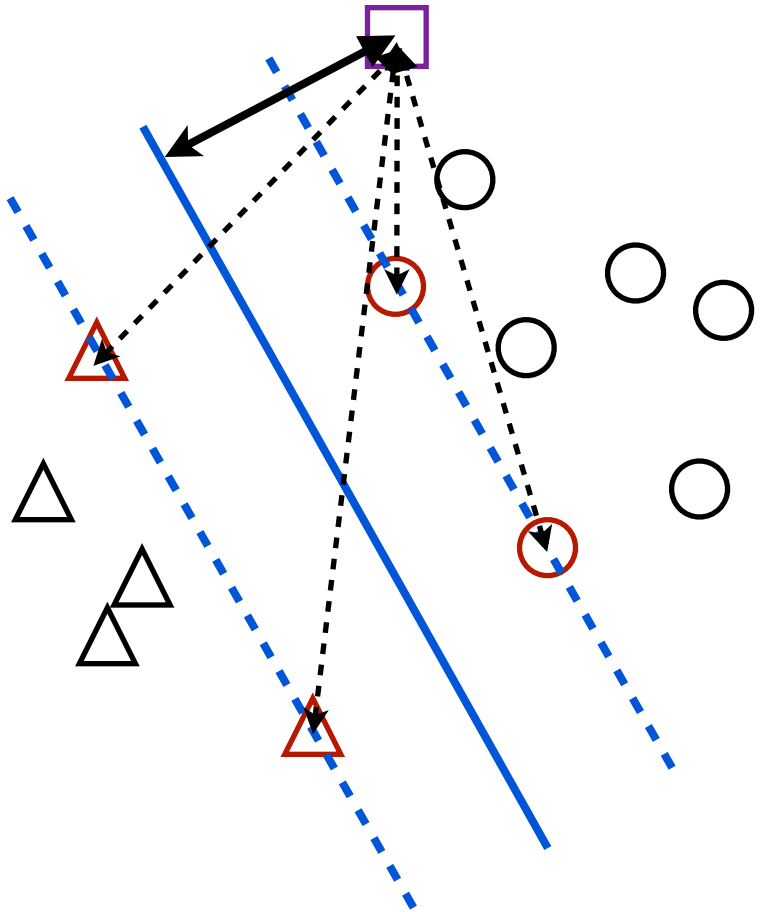
Support vector tracking



the optimal function is in the form of

$$f^*(\cdot) = \sum_i \alpha_i K(\boldsymbol{x}_i, \cdot)$$

Support vector tracking

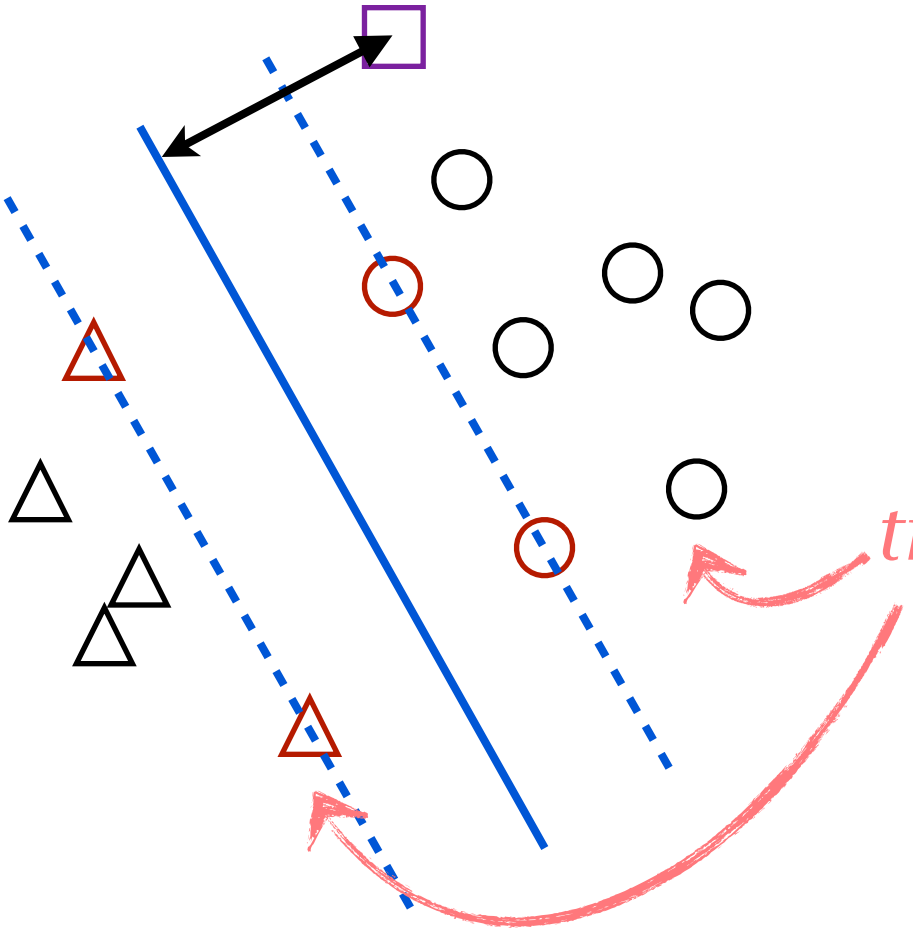


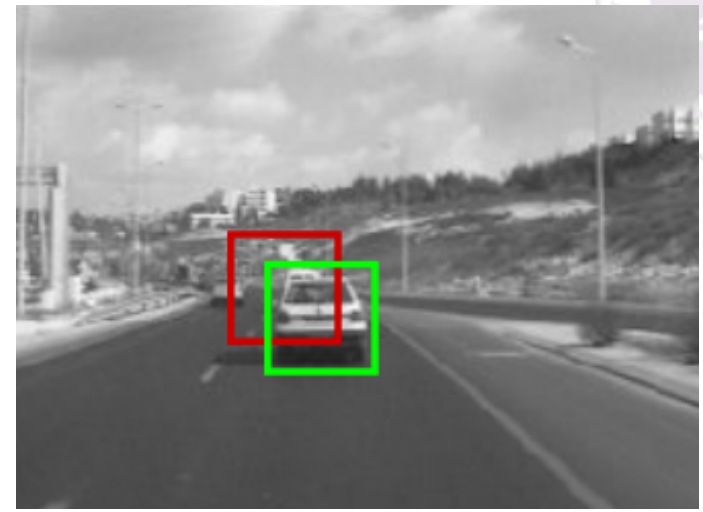
support vectors

the optimal function is in the form of

$$f^*(\cdot) = \sum_i \alpha_i K(\boldsymbol{x}_i, \cdot)$$

Support vector tracking





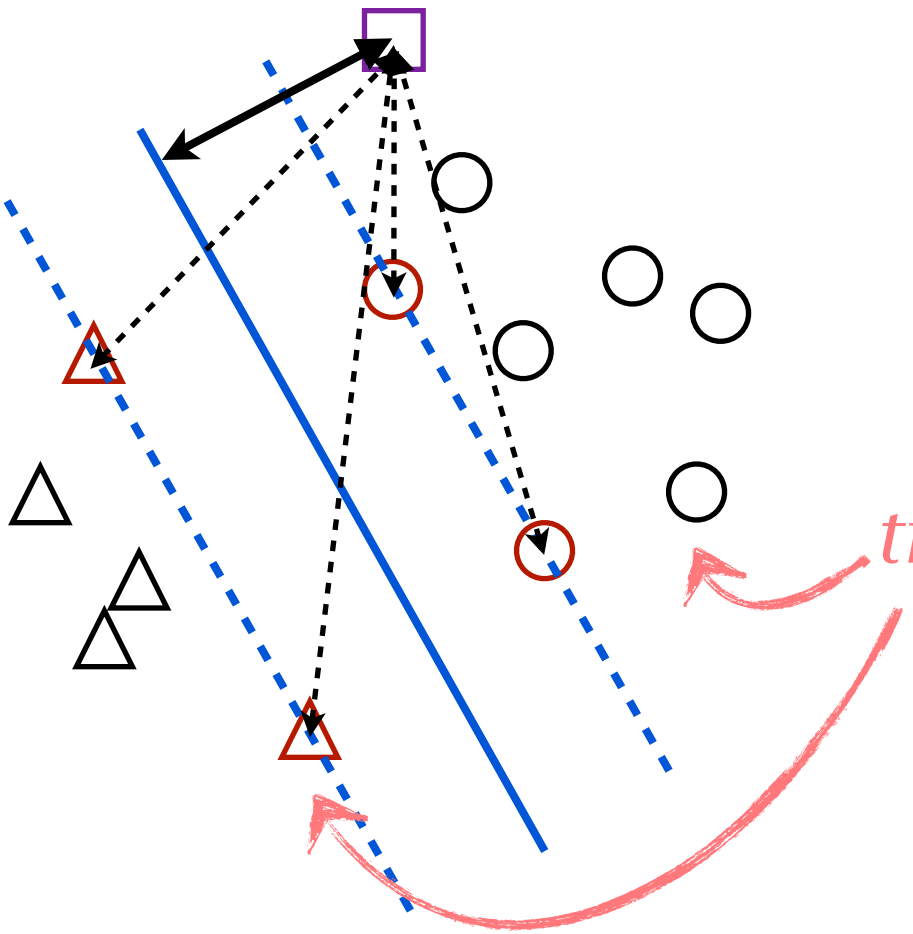
training images cars/noncars

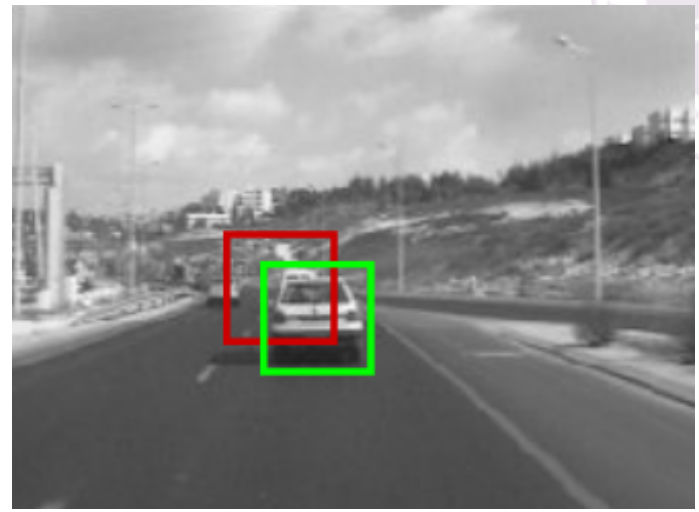
find the largest score about the initial guess

$$score(I) = \sum_i \alpha_i K(\boldsymbol{x}_i, I)$$

$$score(I) = \sum_i \alpha_i (K(\boldsymbol{x}_i, I_{init}) - K(\boldsymbol{x}_i, I))^2$$

Support vector tracking





training images cars/noncars

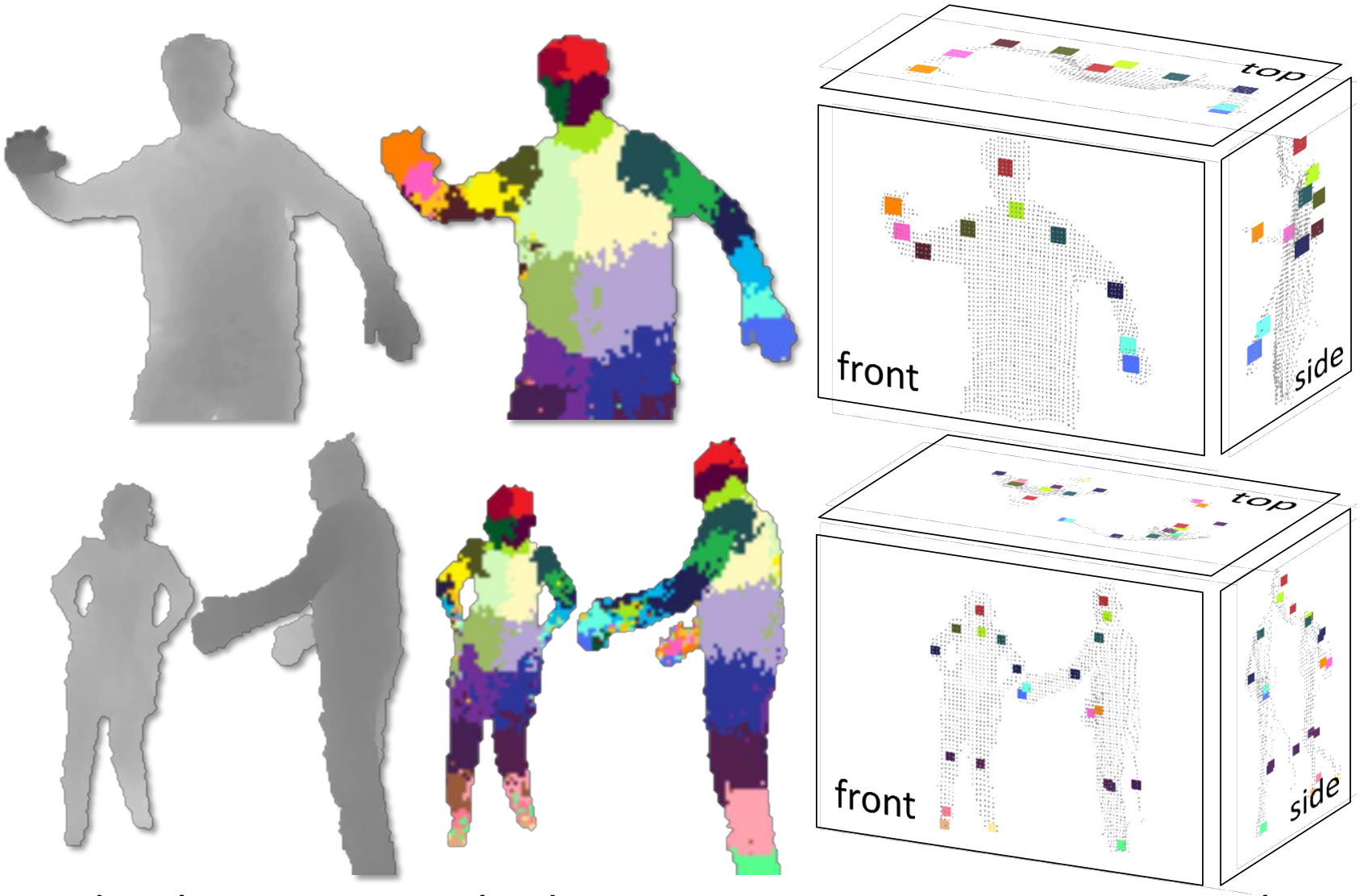
find the largest score about the initial guess

$$score(I) = \sum_i \alpha_i K(\boldsymbol{x}_i, I)$$

$$score(I) = \sum_i \alpha_i (K(\boldsymbol{x}_i, I_{init}) - K(\boldsymbol{x}_i, I))^2$$

Pose estimation from depth data

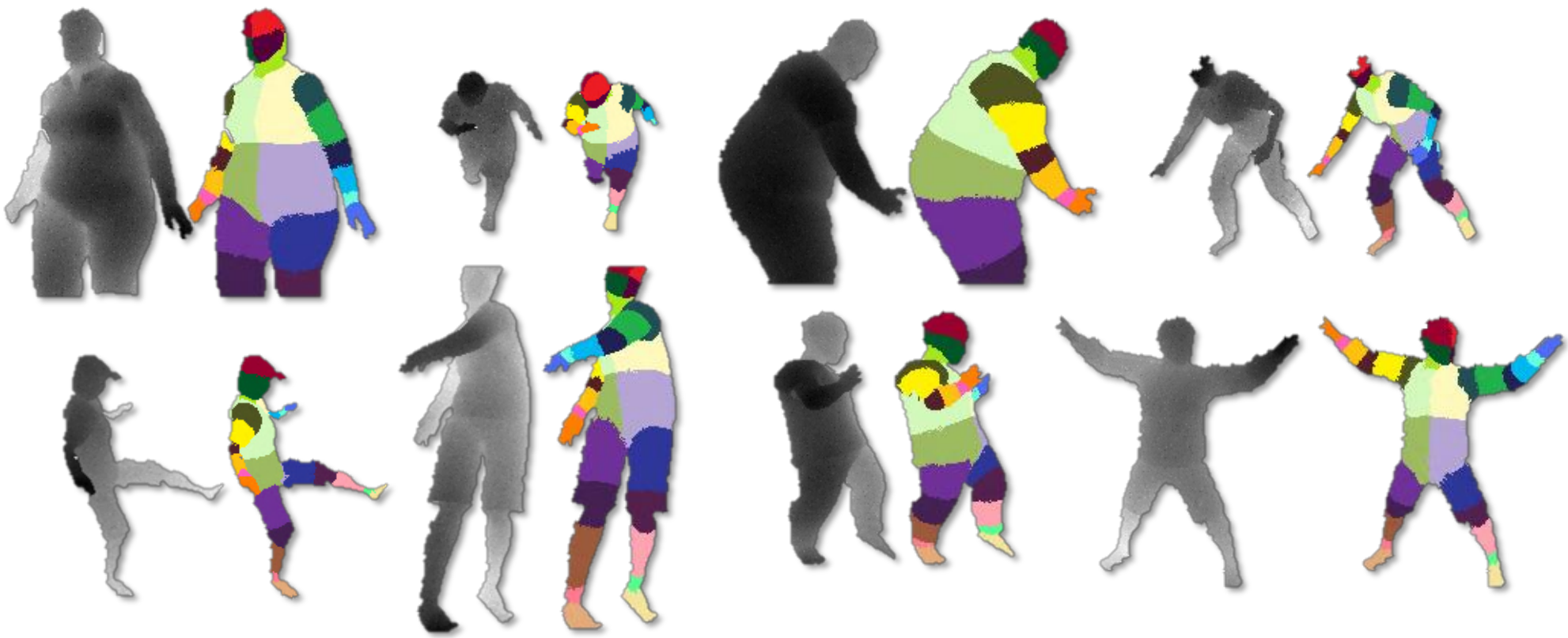
Pose Recognition in Parts [CVPR'11]



depth image ➡ body parts ➡ 3D joint proposals

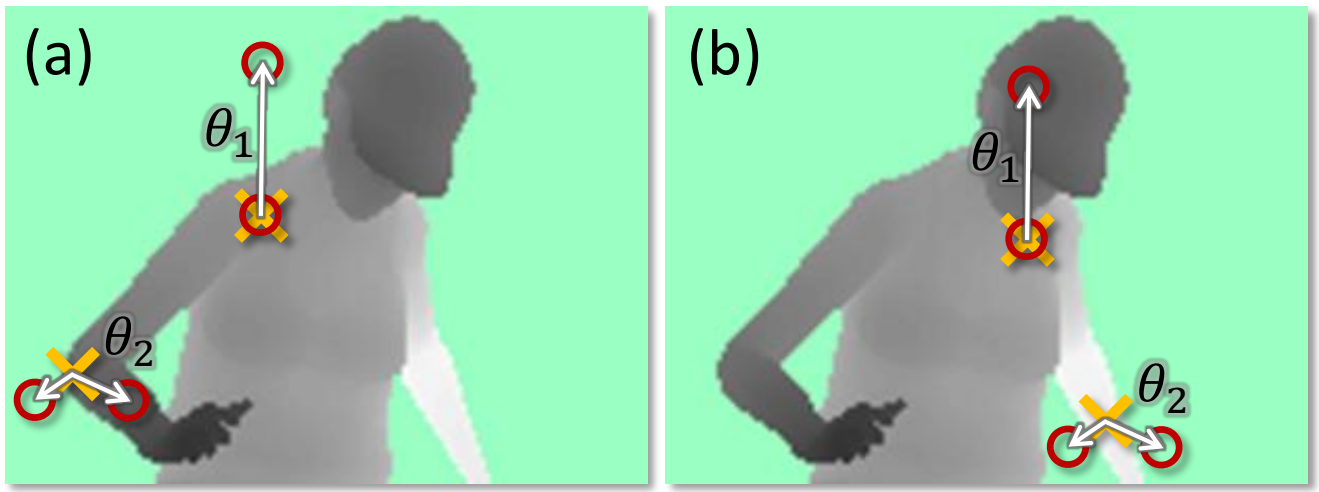
Pose Recognition in Parts [CVPR'11]

Training data from 3D models

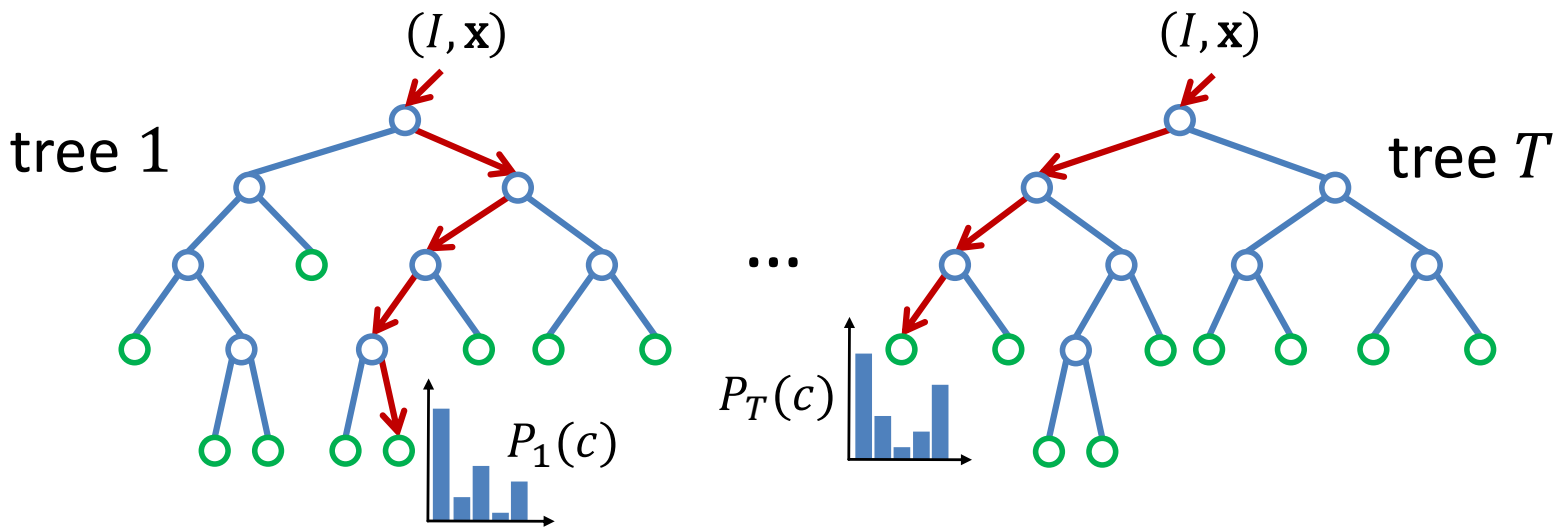


Pose Recognition in Parts [CVPR'11]

Features: random subtractions

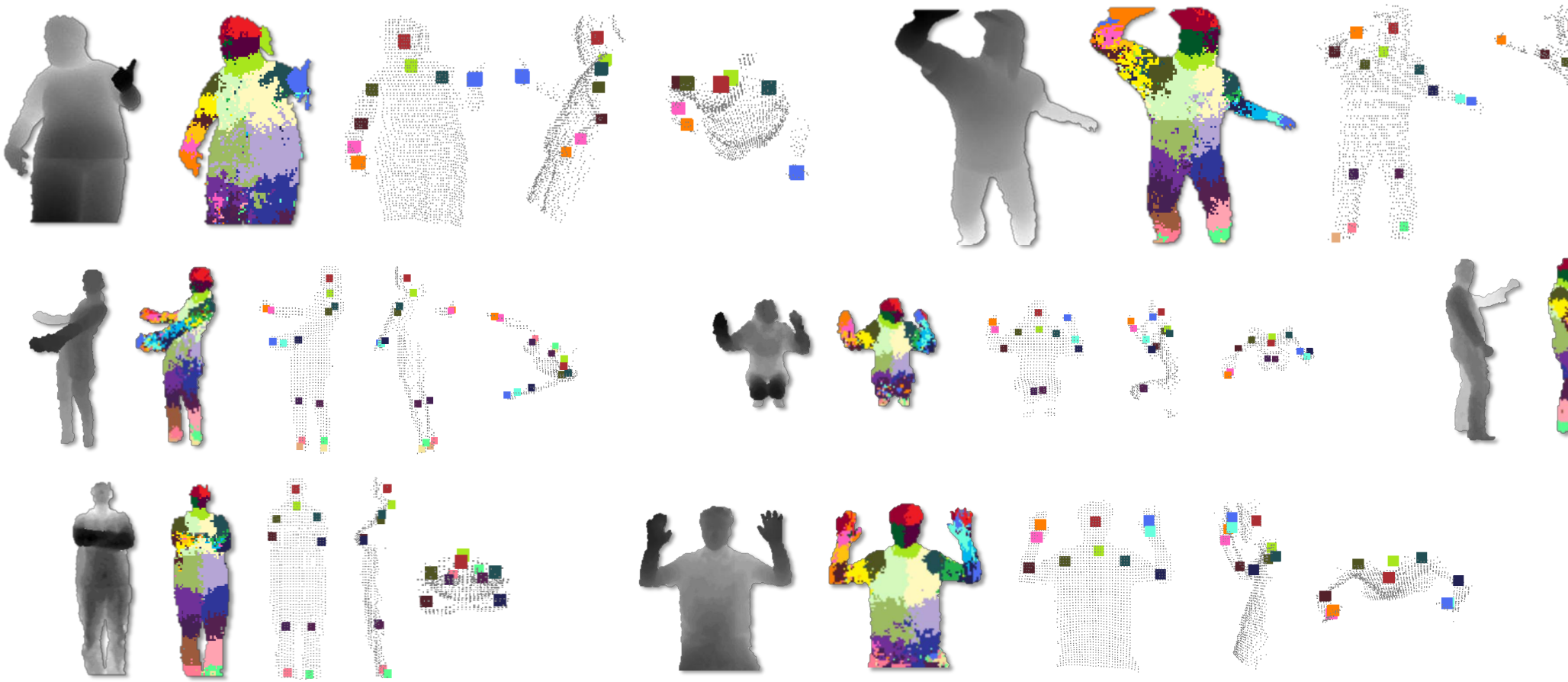


Classifier: random forests of 3 trees

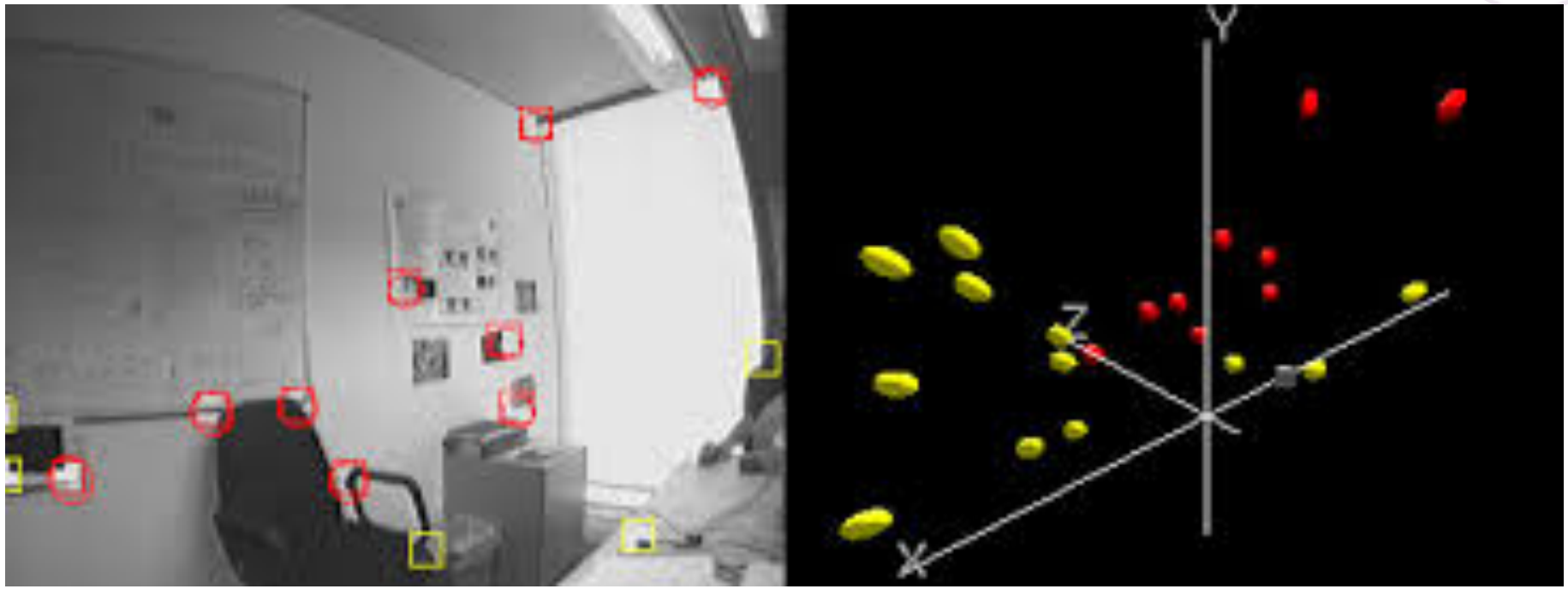


Pose Recognition in Parts [CVPR'11]

results:

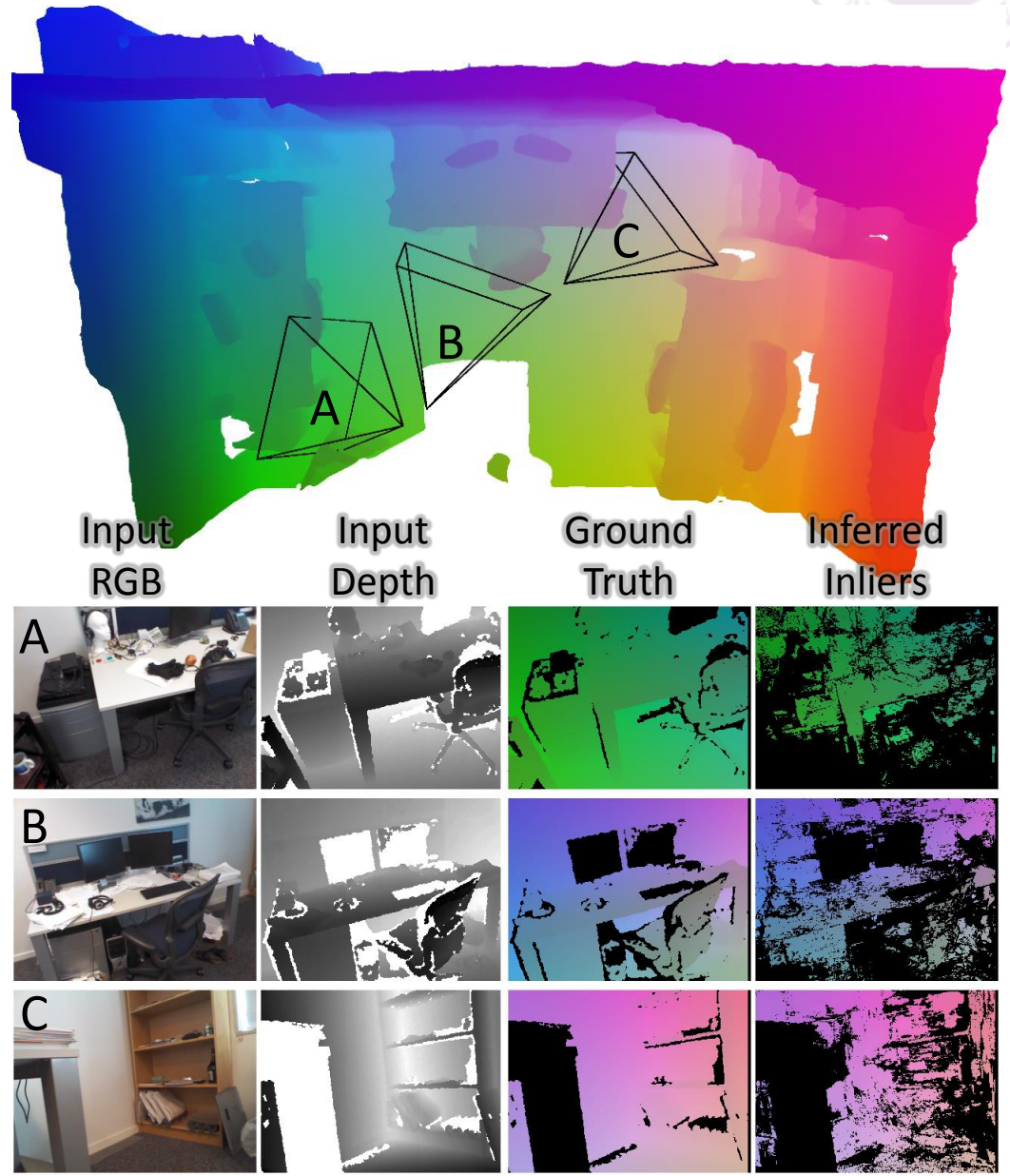


Camera Relocalization



Camera Relocalization in RGB-D Images [CVPR13]

Prediction the location of every pixel



Camera Relocalization in RGB-D Images [CVPR13]

Features: random subtractions

$$f_{\phi}^{\text{depth}}(\mathbf{p}) = D\left(\mathbf{p} + \frac{\boldsymbol{\delta}_1}{D(\mathbf{p})}\right) - D\left(\mathbf{p} + \frac{\boldsymbol{\delta}_2}{D(\mathbf{p})}\right) \quad (2)$$

$$f_{\phi}^{\text{da-rgb}}(\mathbf{p}) = I\left(\mathbf{p} + \frac{\boldsymbol{\delta}_1}{D(\mathbf{p})}, c_1\right) - I\left(\mathbf{p} + \frac{\boldsymbol{\delta}_2}{D(\mathbf{p})}, c_2\right) \quad (3)$$

Classifier: random forests