Motivation

- Objects in rich categories exhibit significant variability
  - Viewpoint variation
  - Intra-class variability
    - bicycles of different types (e.g., mountain bikes, tandems...)
    - People wear different clothes and assume different poses

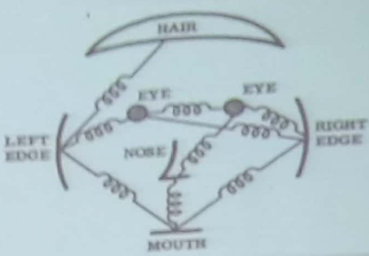
Solution Approaches

- Part Model
- Mixture Model
- Histogram of Gradient
- Feature Pyramid
- Support Vector Machine

- Part Model + Feature Pyramid
  - Pictorial Structures
- HOG + SVM
  - Human Detection: Dalal and Triggs
- All together
  - Deformable Part Model

Part-based Model

- Definition:
  - Root : Capture overall appearance of object
  - Part : Capture local appearance of parts
  - Spring : spatial connections between
- Displacement :
  - Using minimizing energy function to find the optimal displacement



[1] *Pictorial Structures for Object Recognition, Felzenszwalb, Huttenlocher, 2005*

Part-based representation

- K-fans model (D.Crandall, et.all, 2005)

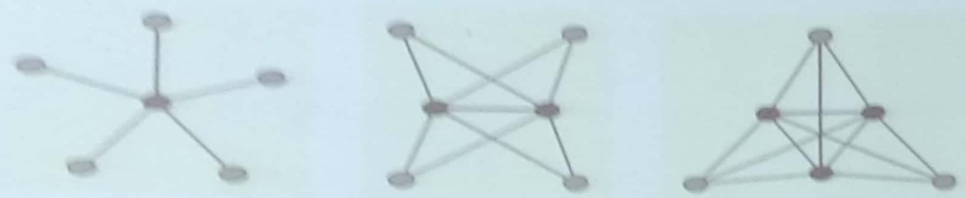


Figure 1. Some k-fans on 6 nodes. The reference nodes are shown in black while the regular nodes are shown in gray.

Pictorial Structure

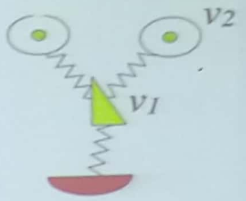
- Matching = Local part evidence + Global constraint

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

- $m_i(l_i)$: matching cost for part i
- $d_{ij}(l_i,l_j)$: deformable cost for connected pairs of parts
- (v_i,v_j): connection between part i and j

Matching on tree structure

$$E(L) = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i,l_j)$$



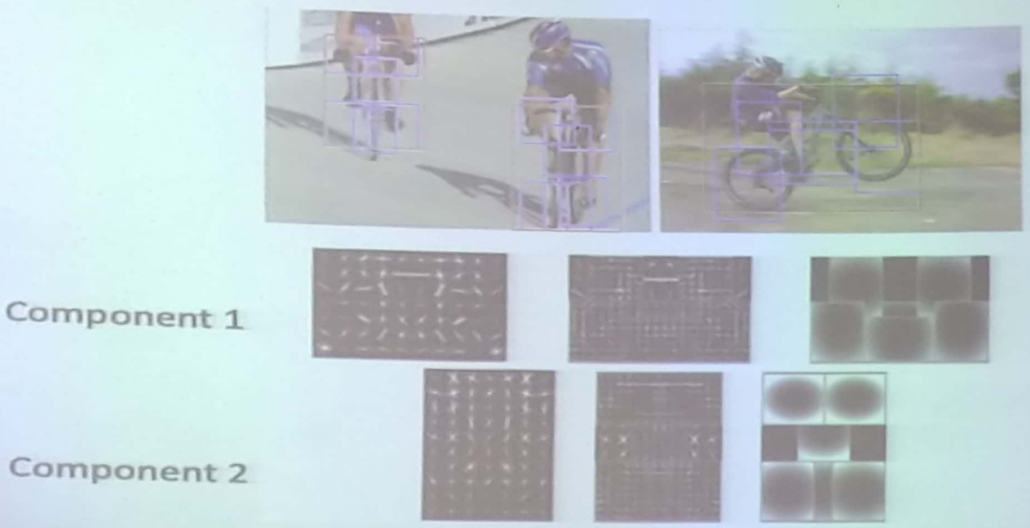
- For each l_1, find best l_2:

$$\text{Best}_2(l_1) = \min_{l_2} \left[m_2(l_2) + d_{12}(l_1,l_2) \right]$$

- Remove v_2, and repeat with smaller tree, until only a single part
- Complexity: $O(nk^2)$: n parts, k locations per part

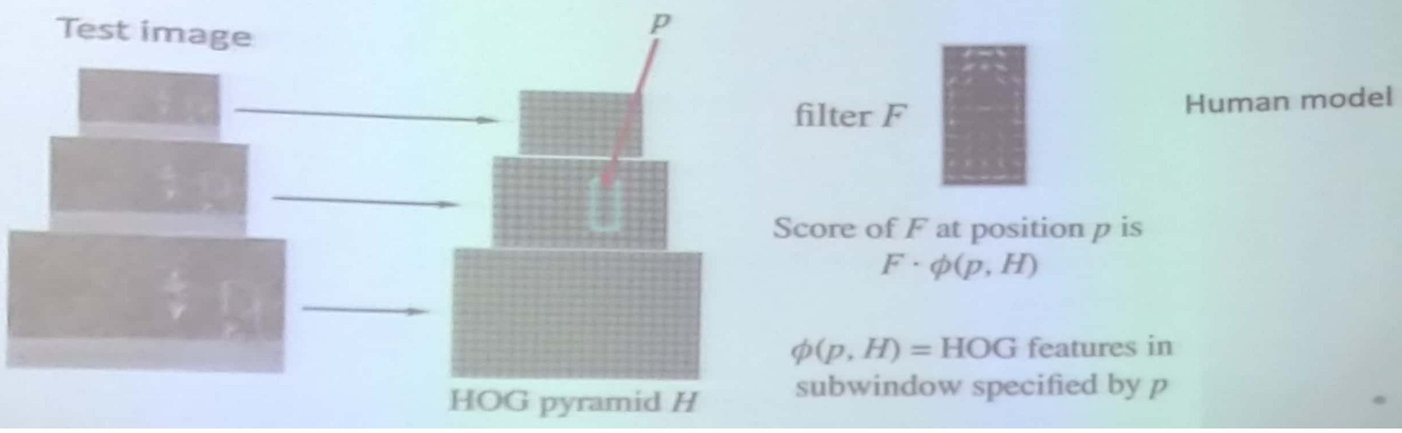
Mixture Model

Build a Mixture Model that includes different components of the same class



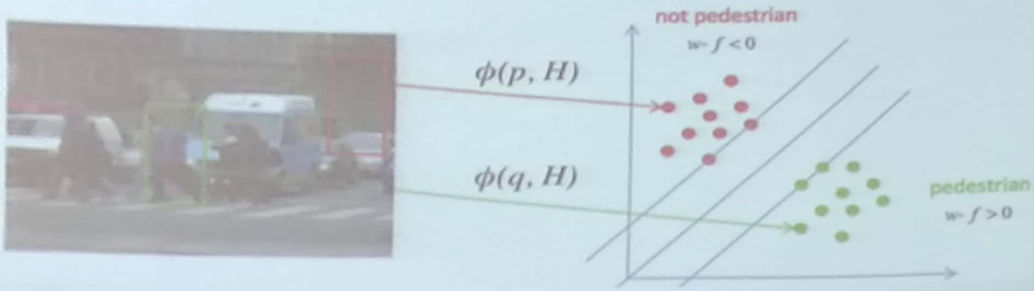
Feature Pyramid

- Develop a representation to decompose images into multiple scales by smoothing and subsampling to extract features of interest and avoid noise



Support Vector Machines

- Build a hyper plane separate positive examples from negative
- In this case, positive is when a human exist in the bounding box

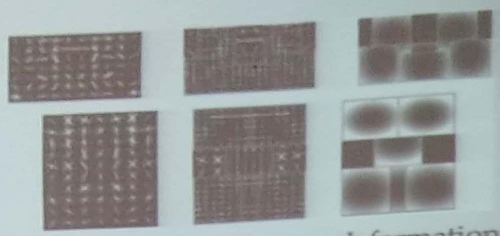


Combining D&T with PS

- Deformable Part Models
  - Build Models
  - Matching
  - Mixture Models
- Latent SVM
- Training Models

Deformable Part Model

- Build a model for an object with n parts :



$$(F_0, P_1, \ldots, P_n, b)$$

root filter

a model for the i-th part

real-valued bias term

(F_i, v_i, d_i)

a filter for the i-th part

for part i relative to the root position

coefficients of a quadratic function

Deformable Part Model

- Part filters are placed at twice the spatial resolution of the placement of the root
- z specifies the location of each filter in feature pyramid

p_i specifies the level and position of the ith filter

$$z = (p_0, \ldots, p_n) \quad p_i = (x_i, y_i, l_i)$$

Deformable Part Model

Score of hypothesis = filter scores - deformation costs

$$\operatorname{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} \underbrace{F_i'}_{\text{filters}} \cdot \underbrace{\phi(H, p_i)}_{\text{feature map}} - \sum_{i=1}^{n} \underbrace{d_i}_{\text{deformation parameters}} \cdot \phi_d\underbrace{(dx_i, dy_i)}_{\text{displacements}} + b,$$

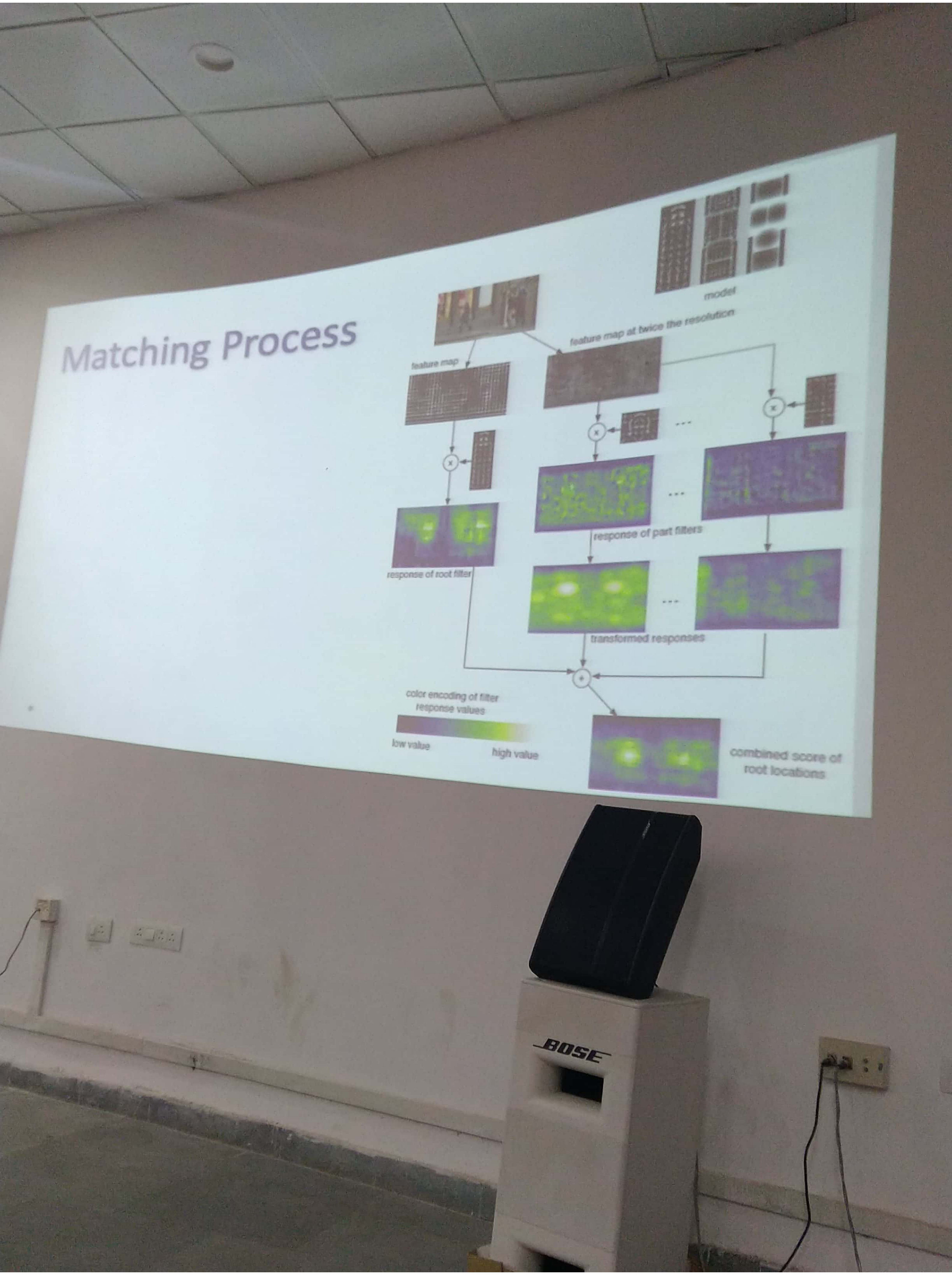
$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i)$$

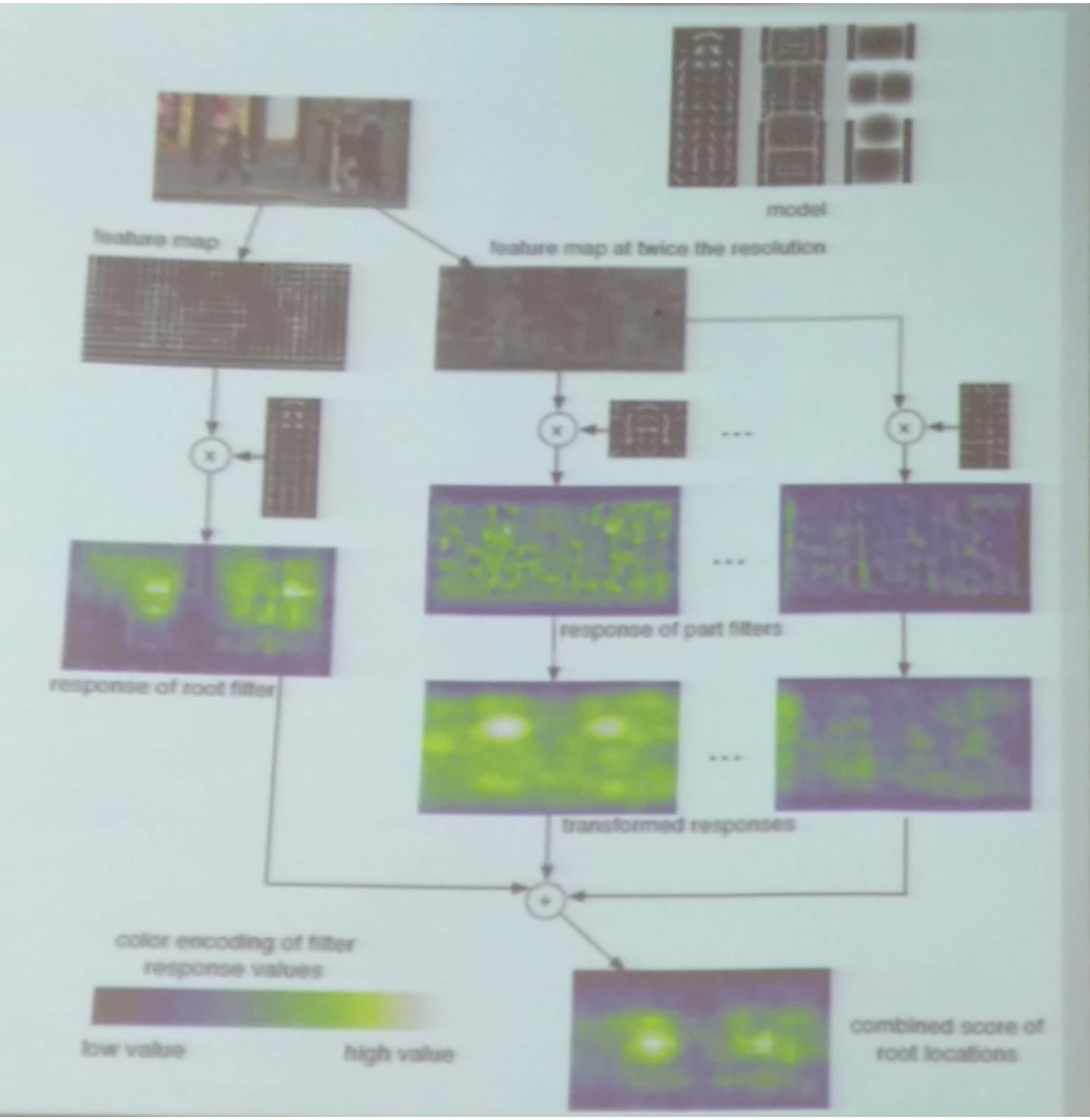
Deformable Part Model

- Given a root position find the best placement of parts:

$$\text{score}(p_0) = \max_{p_1,\ldots,p_n} \text{score}(p_0,\ldots,p_n).$$

- Using sliding window approach, high score of root score define detections


Matching Process
model
feature map at twice the resolution
feature map
response of root filter
response of part filters
transformed responses
color encoding of filter response values
low value
high value
combined score of root locations


model
feature map
feature map at twice the resolution
response of root filter
response of part filters
transformed responses
color encoding of filter response values
low value
high value
combined score of root locations

Matching

Overall root scores :

$$\text{score}(x_0, y_0, l_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b$$

Mixture Models

- A mixture model with m components $M = (M_1, \ldots, M_m)$
- 1<=c<=m

$$z = (c, p_0, \ldots, p_{n_c}) \qquad z' = (p_0, \ldots, p_{n_c})$$

$$\beta = (\beta_1, \ldots, \beta_m)$$

$$\psi(H, z) = (0, \ldots, 0, \psi(H, z'), 0, \ldots, 0)$$

$$\beta \cdot \psi(H, z) = \beta_c \cdot \psi(H, z')$$

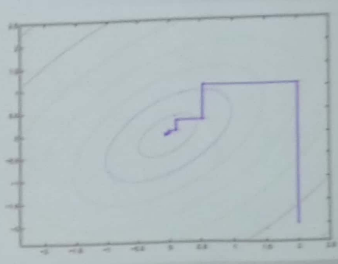
Mixture Models

- Detect objects using a mixture model, we use matching algorithm to find root positions independently for each component

Latent SVM (LSVM)

- Semi-convex:
  - is convex for negative examples
  - for positive examples, convex if latent values fixed
- Solution fixed latent values by coordinate decent:

1) *Relabel positive examples:* Optimize $L_D(\beta, Z_p)$ over Z_p by selecting the highest scoring latent value for each positive example,
$z_i = \text{argmax}_{z \in Z(x_i)} \beta \cdot \Phi(x_i, z)$.
2) *Optimize beta:* Optimize $L_D(\beta, Z_p)$ over β by solving the convex optimization problem defined by $L_{D(Z_p)}(\beta)$.



Training Models

- We initial k component with a specific class, sort the bounding boxes by aspect ratio and intraclass variation then split into k group
- Initial root filters and use coordinate decent to update
- Initial part filters by greedily place parts to cover high energy regions of the root filter
- Training by SVM

Experimental Results

- PASCAL VOC 2006,2007,2008 comp3 challenge datasets
- Some statistics:
    - It takes 2 seconds to evaluate a model in one image (4952 images in the test dataset)
    - It takes 4 hours to train a model
    - MUCH faster than most systems.
    - All of the experiments were done on a 2.8Ghz 8-core Intel Xeon Mac Pro computer running Mac OS X 10.5.

Experimental Results

Measurement: predicted bounding box is correct if it overlaps more than 50 percent with ground truth bounding box; otherwise, considered false positive

Conclusions

- Deformable Part Model
  - Fast matching algorithm
  - handle Viewpoint variation, and Intra-class variability problems
- Still have some problem need to solve:
  - Fixed box size
  - Fixed number of components
- Future Work
  - Build grammar based models that represent objects with variable hierarchical structures
  - Sharing part models between components