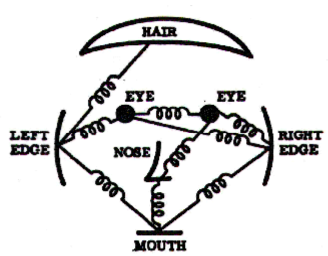
Part-Based Models

Andrew Harp

Part Based Models

- Detect object from physical arrangement of individual features



Implementation

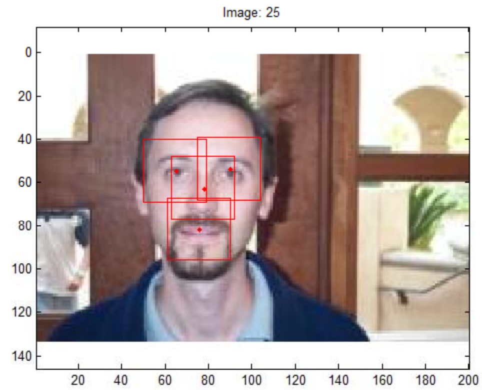
- Based on the Simple Parts and Structure Object Detector by R. Fergus
- Allows user training on N images
- Supports a variety of models
  - Simple part based model
  - efficient model by Felzenszwalb and Huttenlocher
  - Naïve bayes
  - Probabilistic Latent Semantic Analysis

http://people.csail.mit.edu/fergus/iccv2005/partsstructure.html

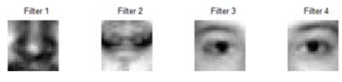
Steps

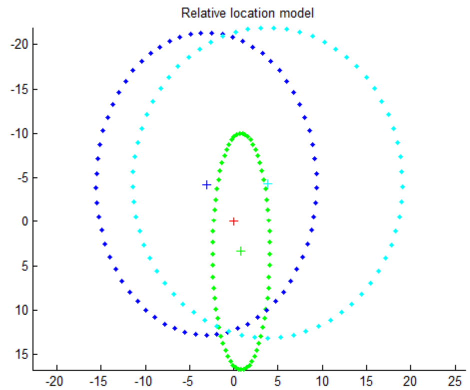
- Preprocessing
  - Resizes images and ground-truth information
- Training
  - User clicks on ordered features in random training examples
- Feature recognition
  - Heat maps of features are created in test image
  - Features are considered to be local maxima
- Object detection
  - Feature configurations are scored using model
  - Best score is selected as recognized object
- Evaluation
  - Bounding boxes are generated around guessed points and compared to ground truth data
  - RPC curves are computed

Model Training



Model Visualization

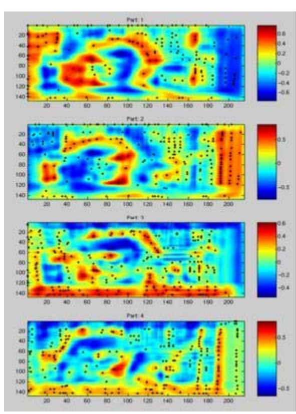


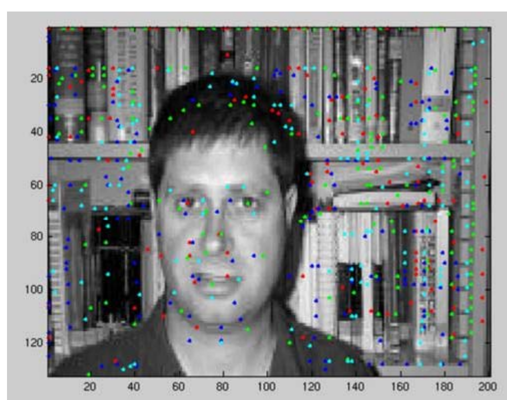


Feature Recognition

- Uses normalized correlation of filters passed over test image
- Sensitive to noise
- Is not size or translation invariant
- Outputs files containing the locations of local maxima

Feature Recognition



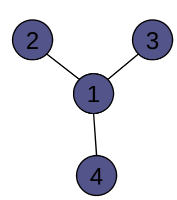


Object Detection

- Uses one of several particular approaches to assign scores to feature configurations
- Highest scoring match is returned with location information

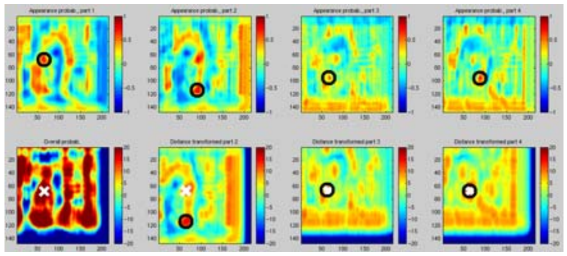
Default Object Detector

- Simple part-based method:
  - Uses a star model
  - Computes all log probabilities of every non-landmark point being associated with a landmark part
  - Sums log probabilities for every landmark



Efficient Object Detector

- Looks at energy used by configuration
- Uses distance transform to efficiently compute energy to deform part given response image and part variance.



My additions

- Expectation Maximization
  - Run trials iteratively
- HOG filtering
  - Compare HOG descriptor correlations to last known filters
- Heatmap shifting
  - Try to shift feature heatmaps over

Expectation Maximization (EM)

- Seeks to feed model based on model's output
- Very sensitive to starting parameters
- Steps:
  - E-step: Compute expected part locations using model
  - M-step: Update model parameters

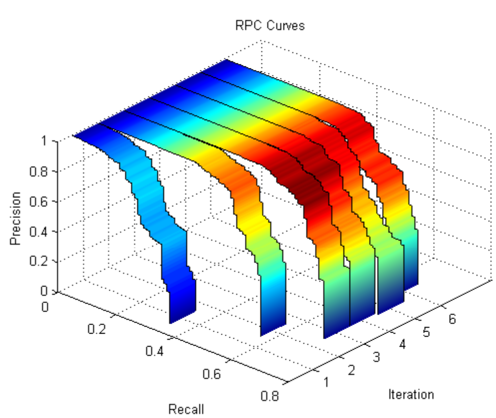
EM Implementation

- Seeded with N initial training examples
- K iterations
- Uses X best matches from the previous iteration to extract new filters
- Model composition variance initially determined by user input
- beyond initial training, unsupervised
  - ground truth data is hidden from it
  - test data selection will influence model

EM Training

- Uses best scoring images from previous iteration as training examples
- Repeats guessed part positions as if it was ground truth information

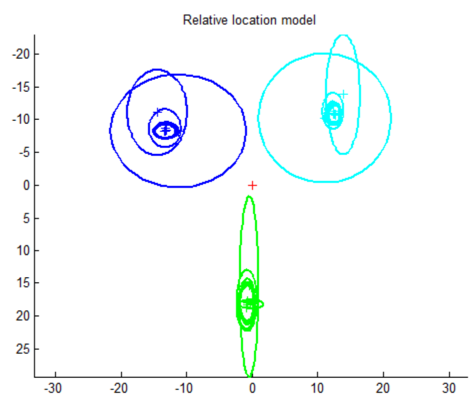
EM Results



Filters Over iterations

Generally, filters get less noisy over time (good for correlation)

Position Variance Over Iterations



HOG filtering

- At training time, computes HOG descriptors of:
  - model parts from previous iteration
  - extracted images from training images
- Determines correlation between HOG descriptors
- If arbitrary threshold is not met, entire match is thrown out
- Prevents incorporation of noisy data that is too radically different from current model

HOG filtering (cont)

Without HOG filtering

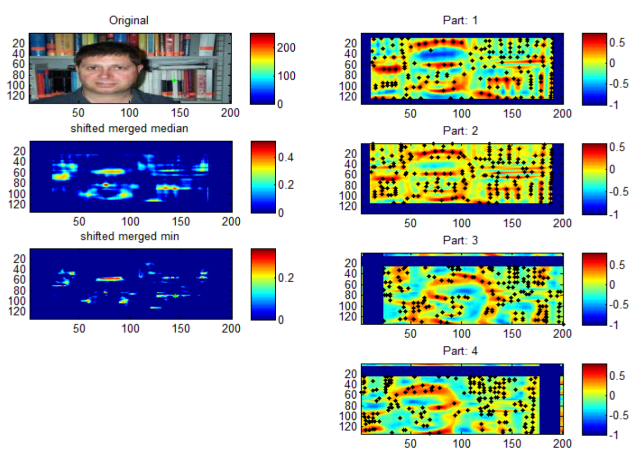
With HOG filtering

Between iterations 1 and 2, HOG Filtering threw out 5 degenerate training examples in the top 20 scorers.

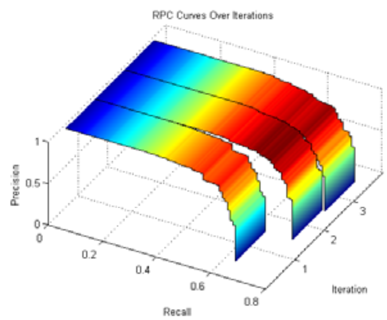
Response Map Shifting

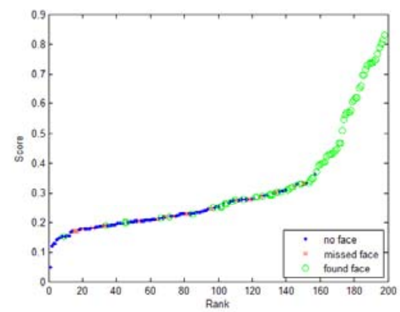
- Similar to Hough Transform
  - Shifts all response maps by mean displacement and blurs by variance
- Benefits:
  - Very fast once filter passes are done
- Drawbacks:
  - Detects entire match based, and not individual parts

Response Map Shifting



Response Map Shifting





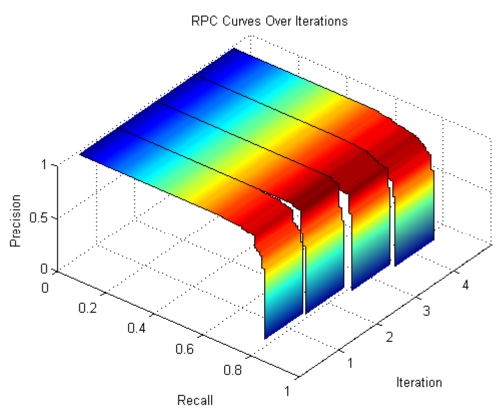
Observations

- Model Variance spikes initially as problem space is explored with messy filters, but fitness of matches with low variance forces it down

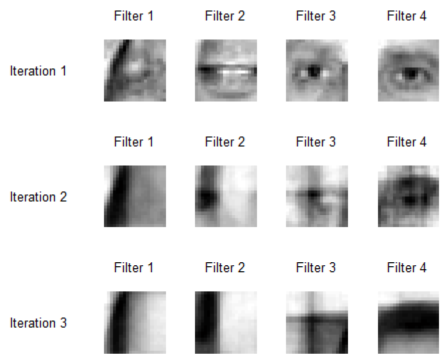
Efficient Method

- Felzenszwalb, P. and Huttenlocher
- Provides better results, converges quickly
- Prefers matches with low deviation from archetype, leading to low variance in next iteration.
- Had to jury-rig model variance to a constant because it converges to 0

Efficient Method



Degenerate Cases



Motorcycle Dataset

- 826 annotated motorcycles, facing left to right
- Testing was inconclusive:
  - Recall was very good, even in noisy images
  - However, number of potential configurations was very low, due to relative model size



Observations

- EM can reinforce degenerate cases.
- Requires some knowledge of the training data to find
- Can find largest cluster, while excluding outliers
- Helps most if initial input was bad, but correctable

Potential Improvements

- Maintain multiple archetypes for each part
  - Would take advantage of iterative nature of EM to expand feature library
- Use better method than correlation for determining feature maps
  - HOG?

References

- Felzenszwalb, P. and Huttenlocher, D. "Pictorial Structures for Object Recognition." Intl. Journal of Computer Vision, 61(1), pp. 55-79, January 2005.
- Fischler, M. and Elschlager, R. "The representation and matching of pictorial structures." IEEE Transactions on Computers, 22(1):67--92, 1973.

Thanks