Modern Boundary Detection

Szeliski 4.2

Computer Vision

CS 143, Brown

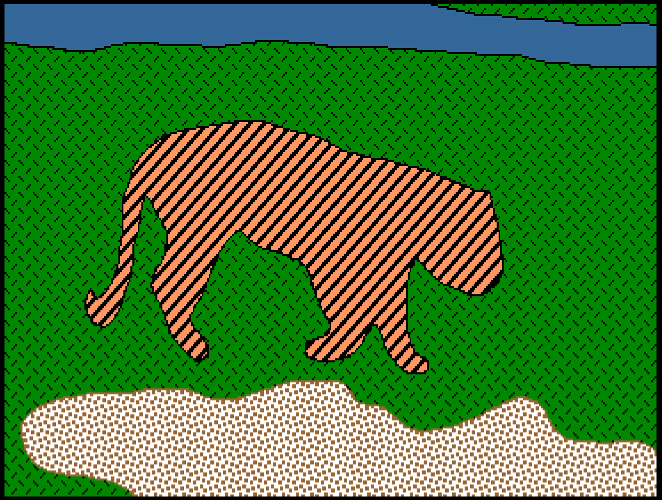
James Hays

Many slides Michael Maire, Jitendra Malek

Today’s lecture

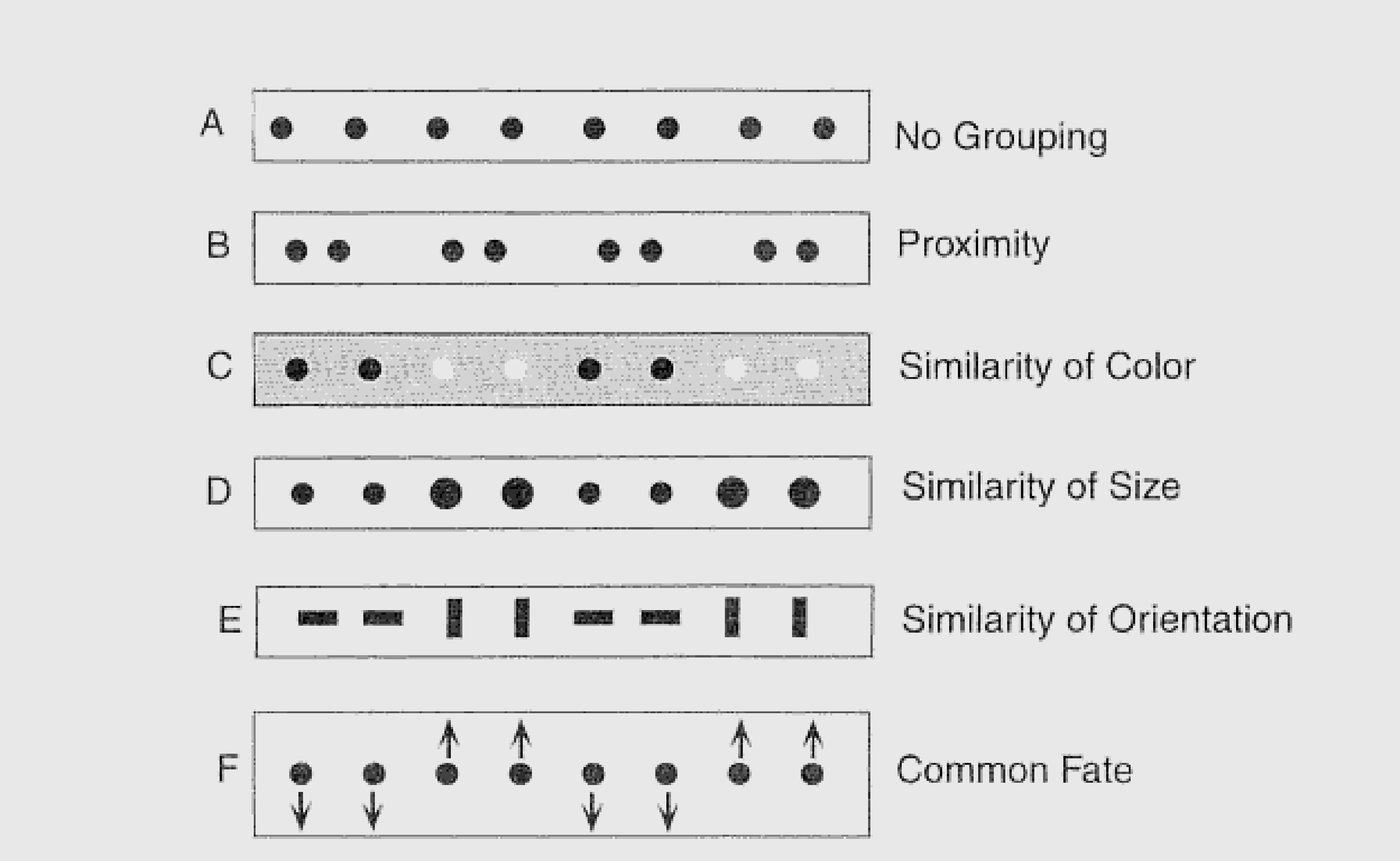
- Segmentation vs Boundary Detection
- Why boundaries / Grouping?
- Recap: Canny Edge Detection
- The Berkeley Segmentation Data Set
- pB boundary detector
  - “local” pB today and
  - “global” pB next lecture

From Images to Objects



"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees." --Max Wertheimer

Grouping factors



Canny edge detector

- This is probably the most widely used edge detector in computer vision
- Theoretical model: step-edges corrupted by additive Gaussian noise
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization

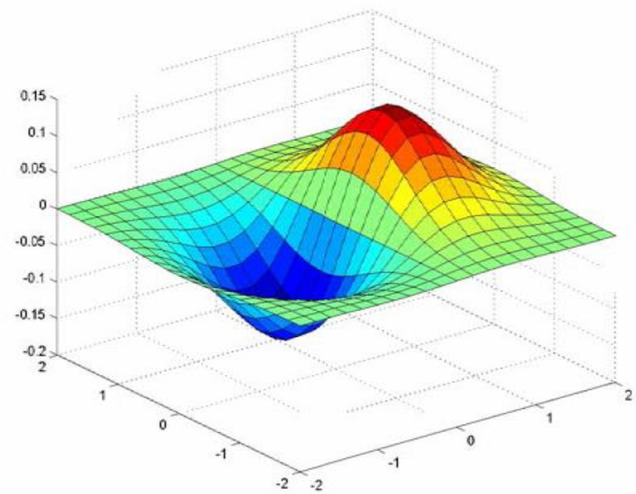
J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Source: L. Fei-Fei

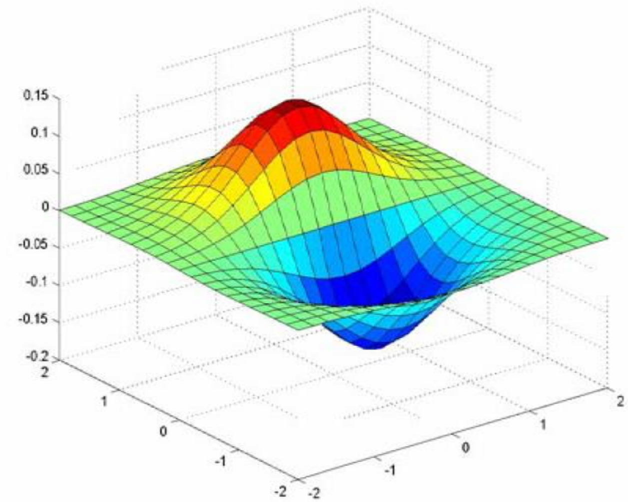
Example

original image (Lena)

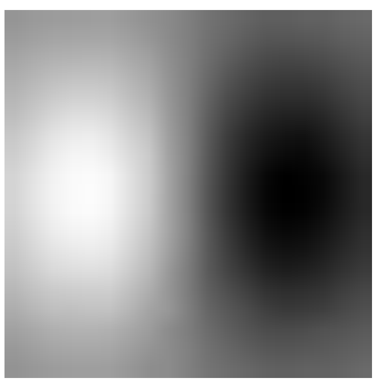
Derivative of Gaussian filter

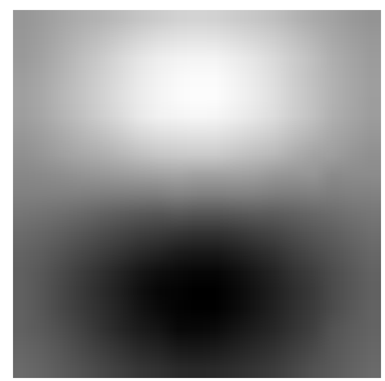


x-direction

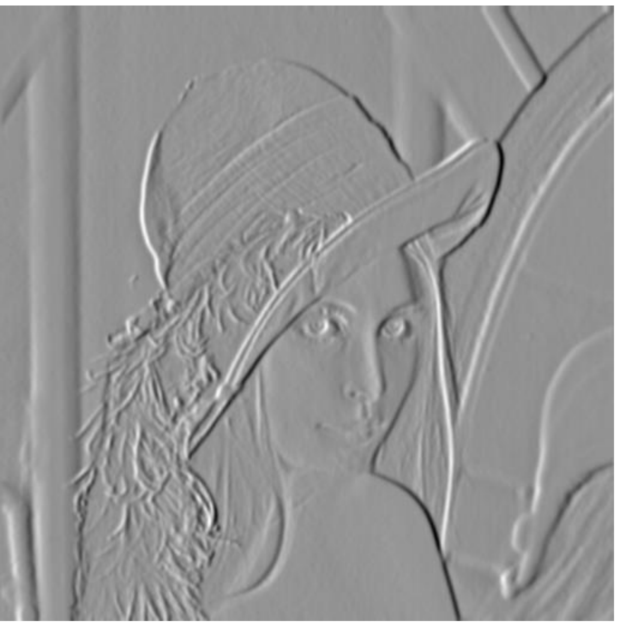


y-direction





Compute Gradients (DoG)



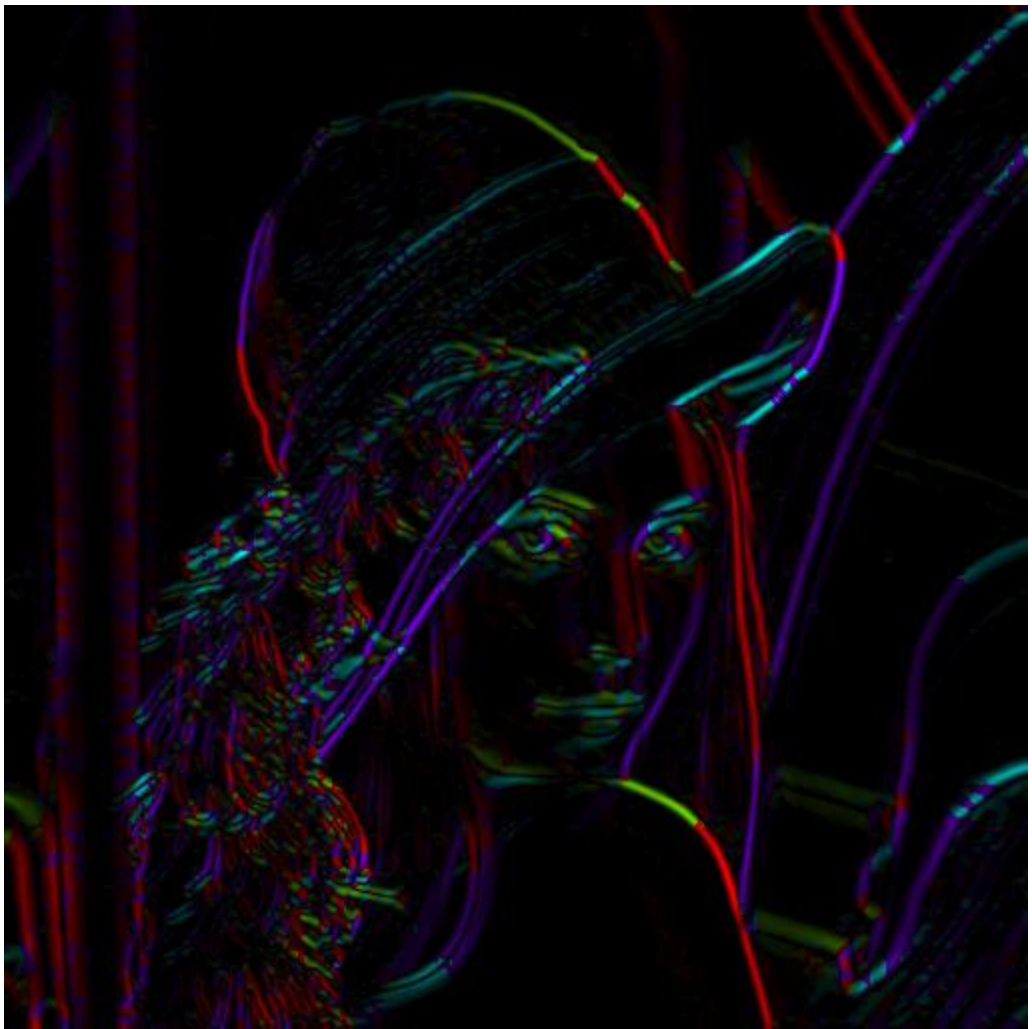
X-Derivative of Gaussian

Y-Derivative of Gaussian

Gradient Magnitude

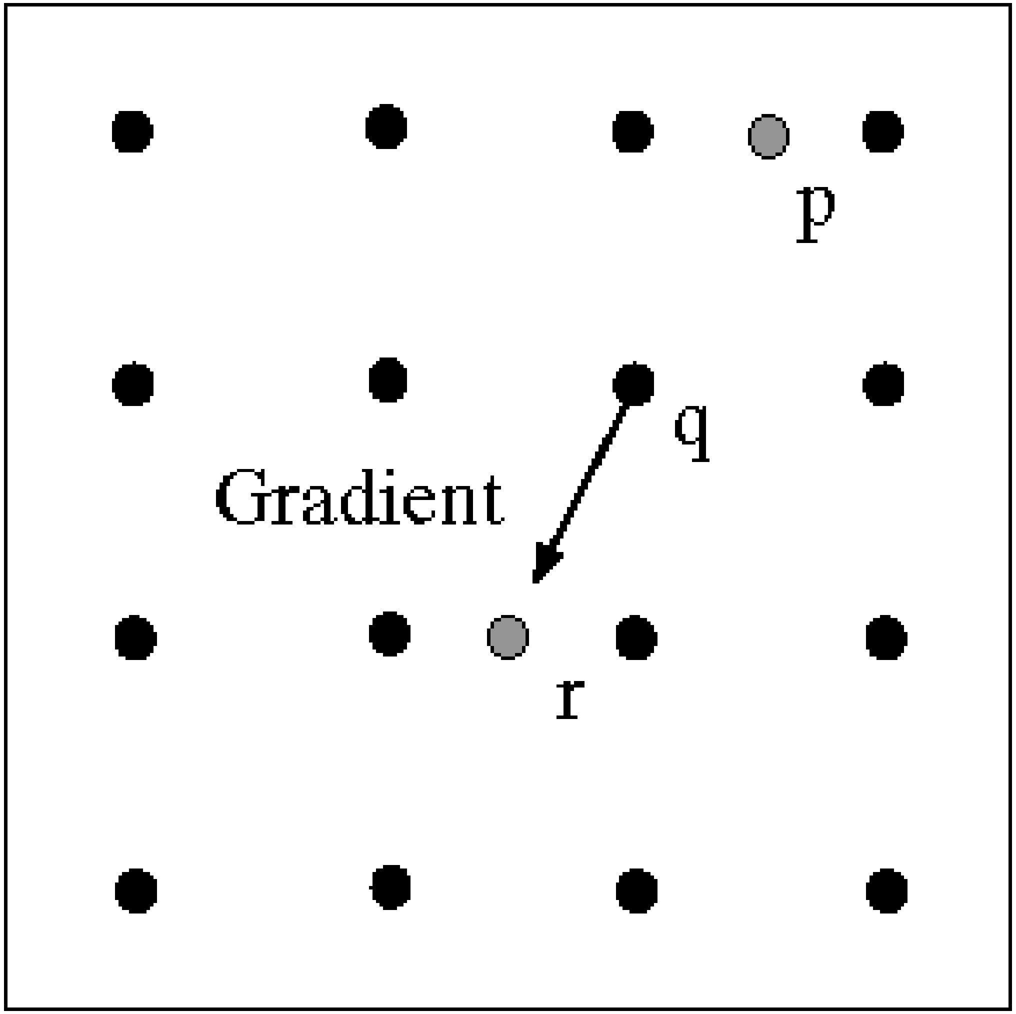
Get Orientation at Each Pixel

- Threshold at minimum level
- Get orientation

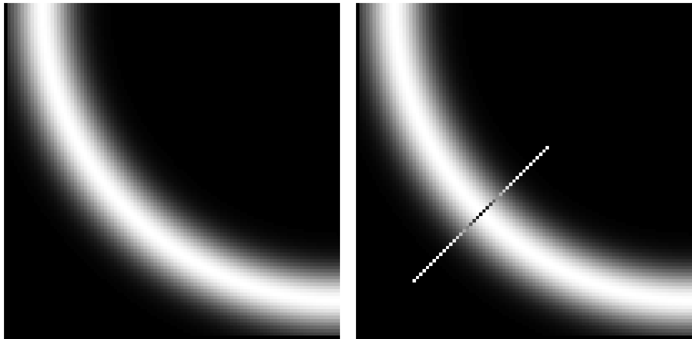


theta = atan2(gy, gx)

Non-maximum suppression for each orientation



At q, we have a maximum if the value is larger than those at both p and at r. Interpolate to get these values.



Source: D. Forsyth

Before Non-max Suppression

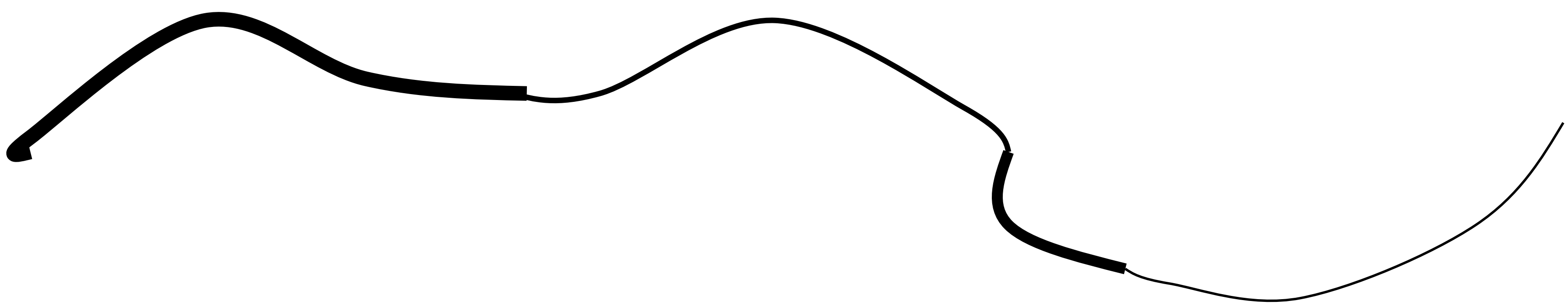
After non-max suppression

Hysteresis thresholding

- Threshold at low/high levels to get weak/strong edge pixels
- Do connected components, starting from strong edge pixels

Hysteresis thresholding

- Check that maximum value of gradient value is sufficiently large
  - drop-outs? use **hysteresis**
    - use a high threshold to start edge curves and a low threshold to continue them.



Source: S. Seitz

Final Canny Edges

Canny edge detector

1. Filter image with x, y derivatives of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   – Thin multi-pixel wide “ridges” down to single pixel width
4. Thresholding and linking (hysteresis):
   – Define two thresholds: low and high
   – Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: edge(image, ‘canny’)

Source: D. Lowe, L. Fei-Fei

I made a new boundary detector!

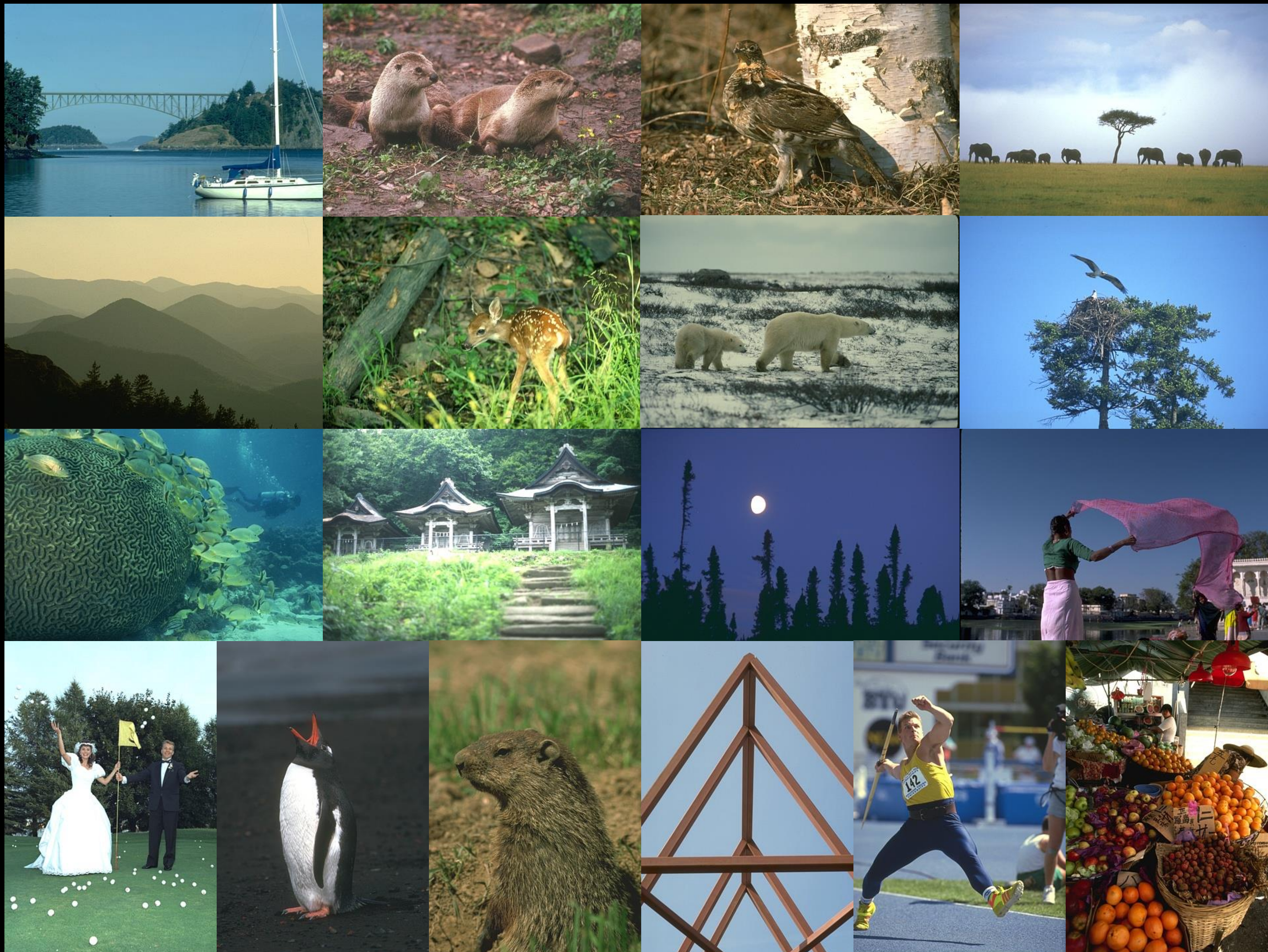
- How do I show that it is better than your boundary detector?

Berkeley Segmentation Data Set

David Martin, Charless Fowlkes,
Doron Tal, Jitendra Malik

UC Berkeley

{dmartin,fowlkes,doron,malik}@eecs.berkeley.edu


142

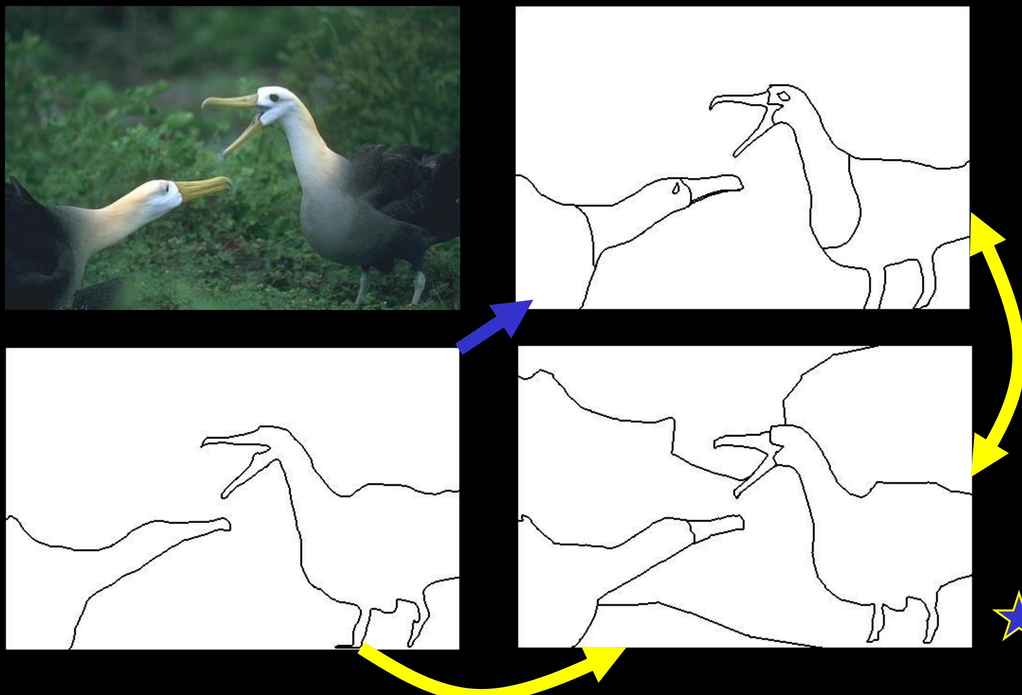

282
1257
1447

Protocol

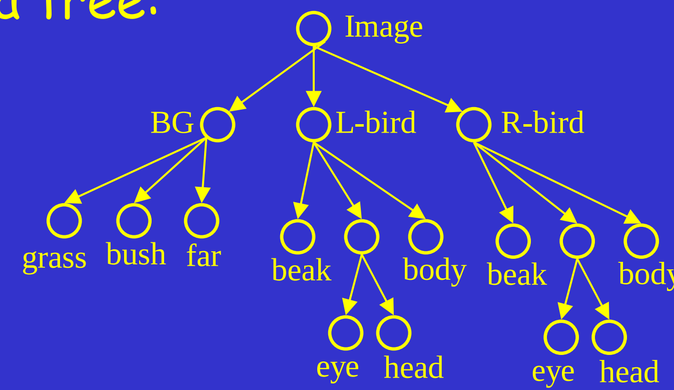
You will be presented a photographic image. Divide the image into some number of segments, where the segments represent "things" or "parts of things" in the scene. The number of segments is up to you, as it depends on the image. Something between 2 and 30 is likely to be appropriate. It is important that all of the segments have approximately equal importance.

- Custom segmentation tool
- Subjects obtained from work-study program
  (UC Berkeley undergraduates)

Segmentations are Consistent



Perceptual organization forms a tree:



- A,C are refinements of B
- A,C are mutual refinements
- A,B,C represent the same percept
  - Attention accounts for differences

☆ Two segmentations are consistent when they can be explained by the same segmentation tree (i.e. they could be derived from a single perceptual organization).

Dataset Summary

- 30 subjects, age 19-23
  - 17 men, 13 women
  - 9 with artistic training
- 8 months
- 1,458 person hours
- 1,020 Corel images
- 11,595 Segmentations
  - 5,555 color, 5,554 gray, 486 inverted/negated

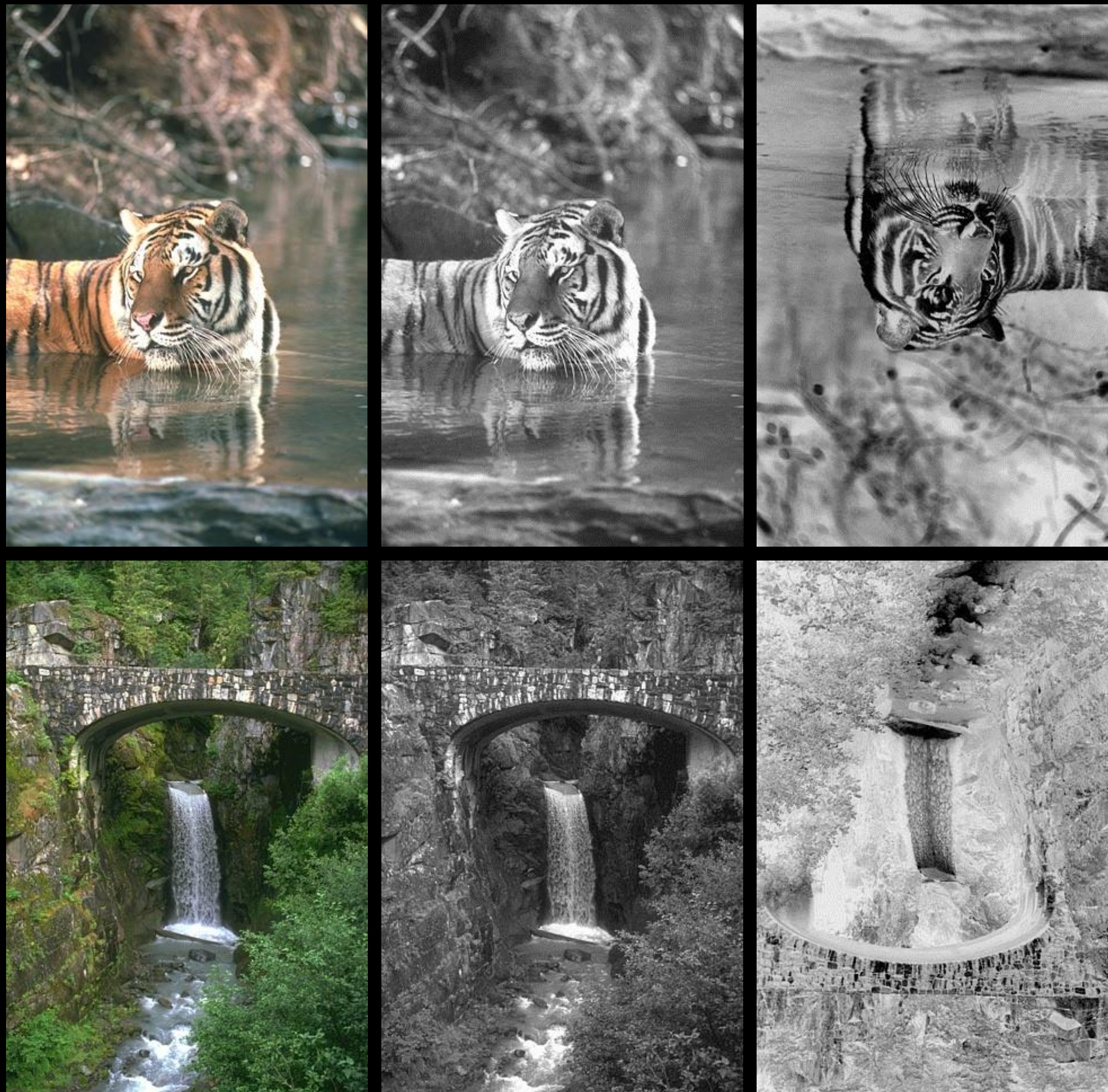
Gray, Color, InvNeg Datasets

- Explore how various high/low-level cues affect the task of image segmentation by subjects
  - Color = full color image
  - Gray = luminance image
  - InvNeg = inverted negative luminance image

Color **Gray** **InvNeg**

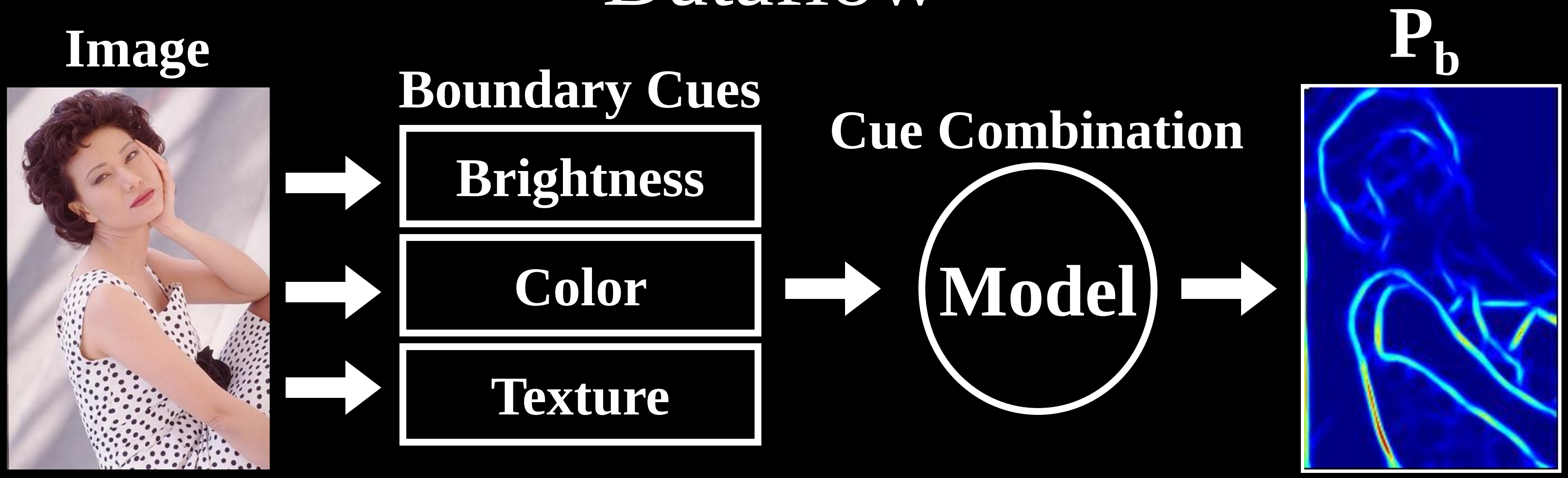
InvNeg

Color Gray InvNeg



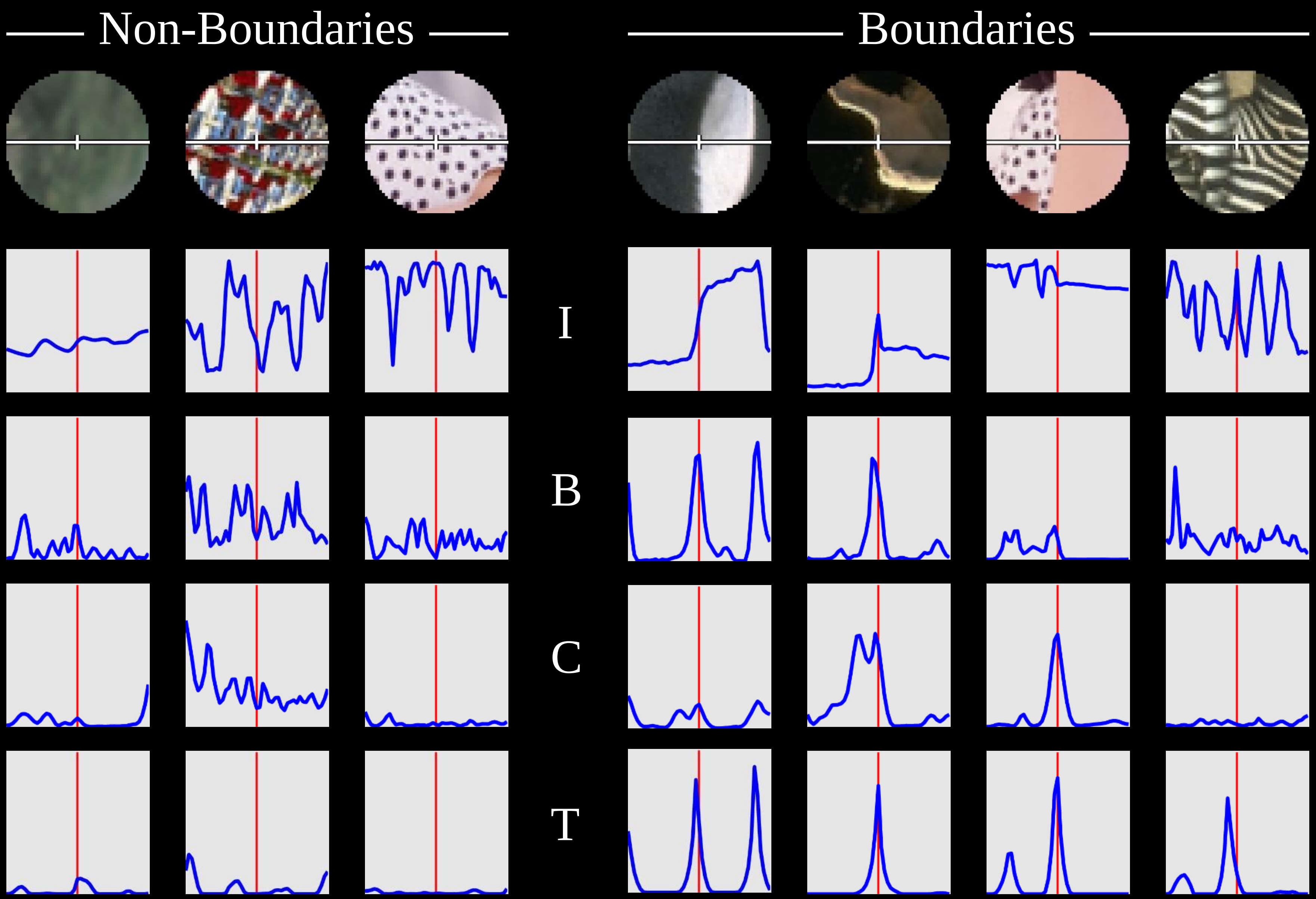
Pb Detector

Dataflow



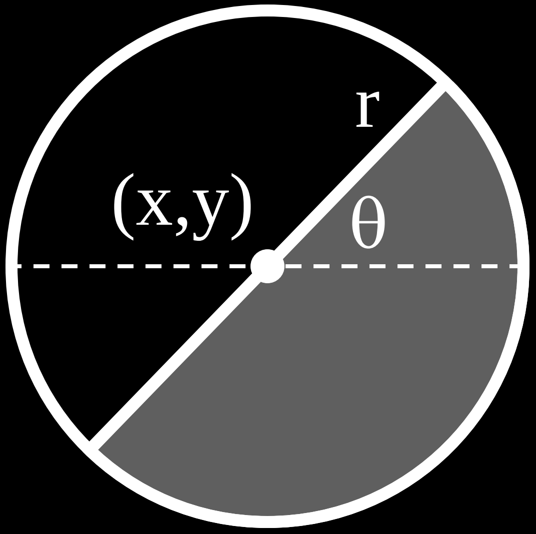
Challenges: texture cue, cue combination

Goal: learn the posterior probability of a boundary $P_b(x,y,\theta)$ from local information only

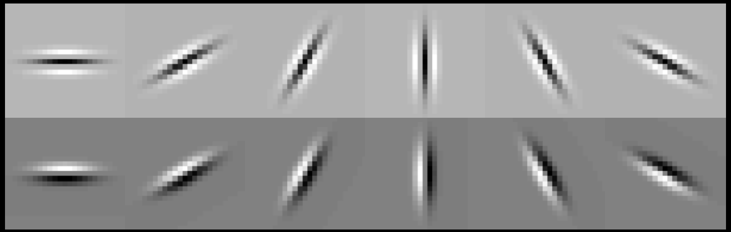

Non-Boundaries
Boundaries
I
B
C
T

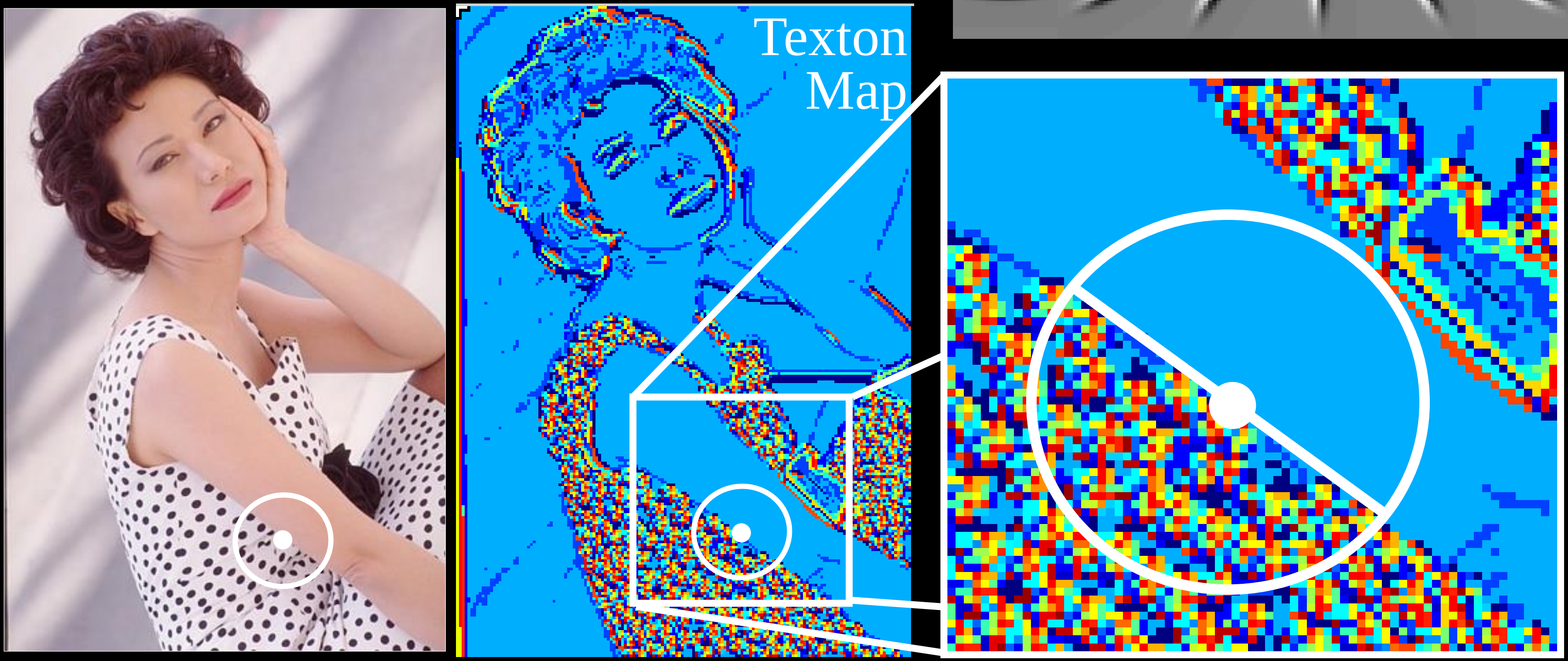
Brightness and Color Features

- 1976 CIE L*a*b* colorspace
- Brightness Gradient BG(x,y,r,θ)
  - χ^2 difference in L* distribution
- Color Gradient CG(x,y,r,θ)
  - χ^2 difference in a* and b* distributions



Texture Feature





- Texture Gradient TG(x,y,r,θ)
  - χ^2 difference of texton histograms
  - Textons are vector-quantized filter outputs

Cue Combination Models

- Classification Trees
  - Top-down splits to maximize entropy, error bounded
- Density Estimation
  - Adaptive bins using k-means
- Logistic Regression, 3 variants
  - Linear and quadratic terms
  - Confidence-rated generalization of AdaBoost (Schapire&Singer)
- Hierarchical Mixtures of Experts (Jordan&Jacobs)
  - Up to 8 experts, initialized top-down, fit with EM
- Support Vector Machines (`libsvm`, Chang&Lin)
  - Gaussian kernel, ν-parameterization

➢ Range over bias, complexity, parametric/non-parametric

Computing Precision/Recall

Recall = Pr(signal|truth) = fraction of ground truth found by the signal

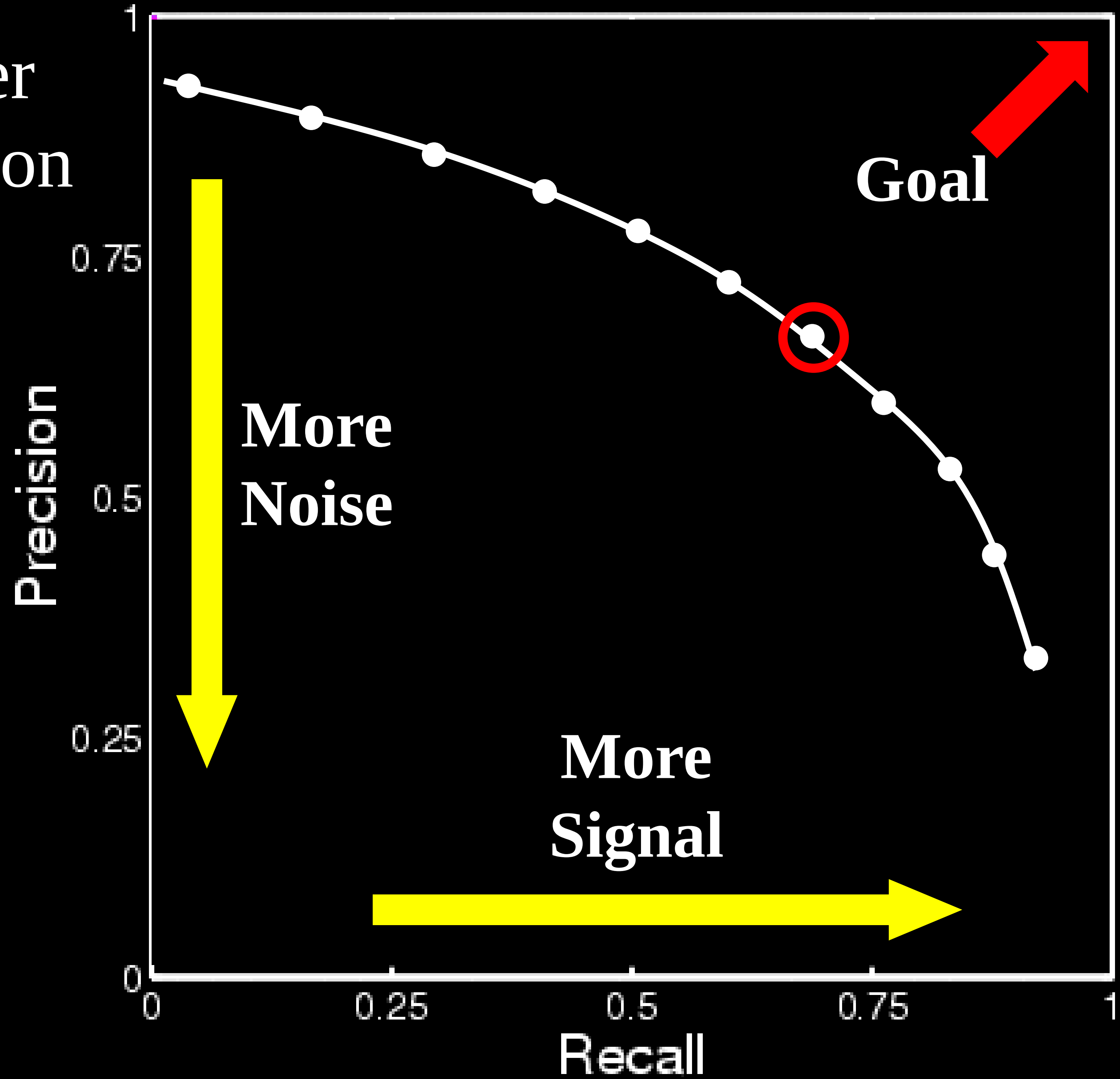
Precision = Pr(truth|signal) = fraction of signal that is correct

- Always a trade-off between the two
- Standard measures in information retrieval (van Rijsbergen XX)
- ROC from standard signal detection the wrong approach

Strategy

- Detector output (Pb) is a soft boundary map
- Compute precision/recall curve:
  - Threshold Pb at many points t in [0,1]
  - Recall = Pr(Pb>t|seg=1)
  - Precision = Pr(seg=1|Pb>t)

Classifier Comparison

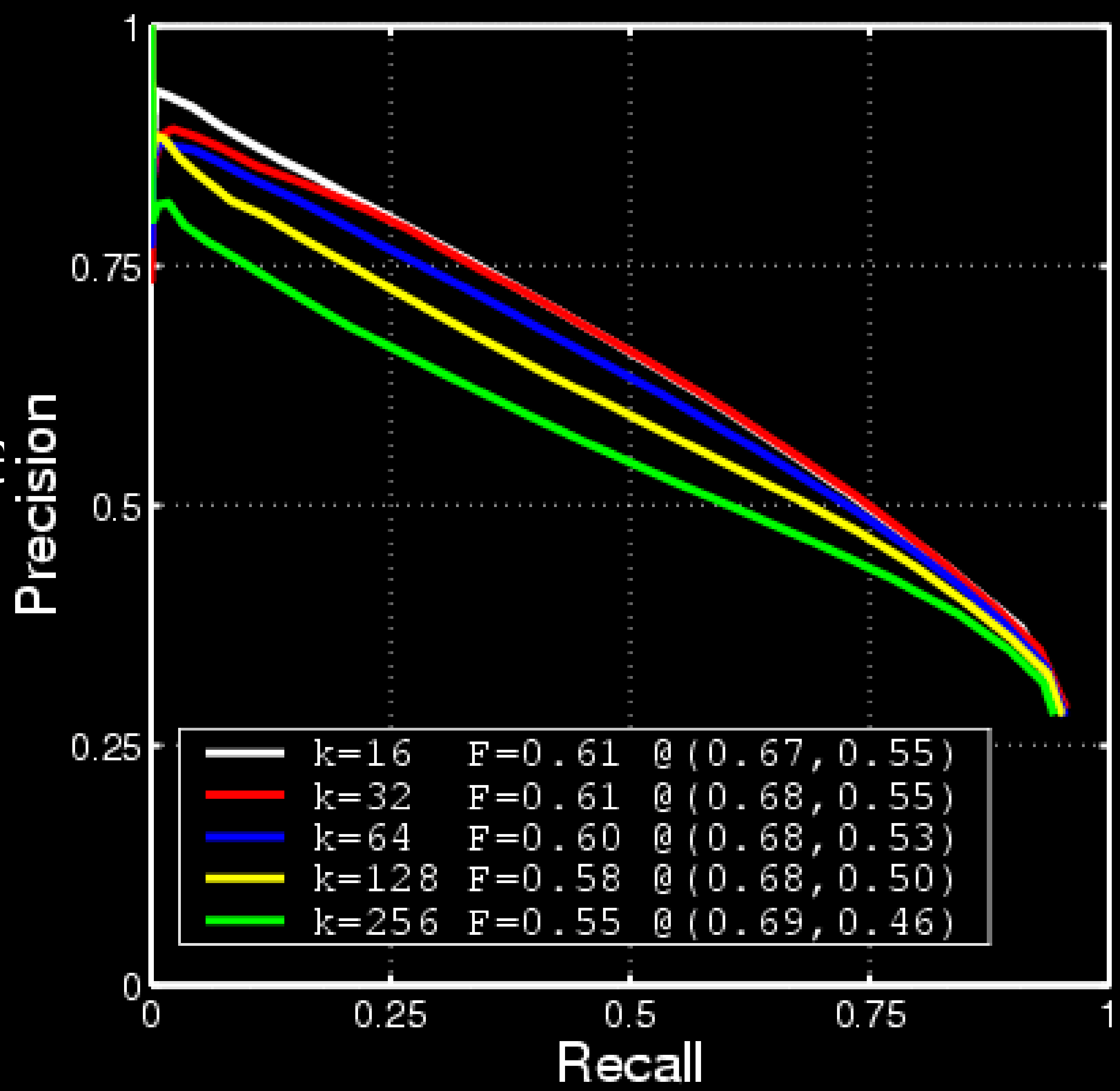


Cue Calibration

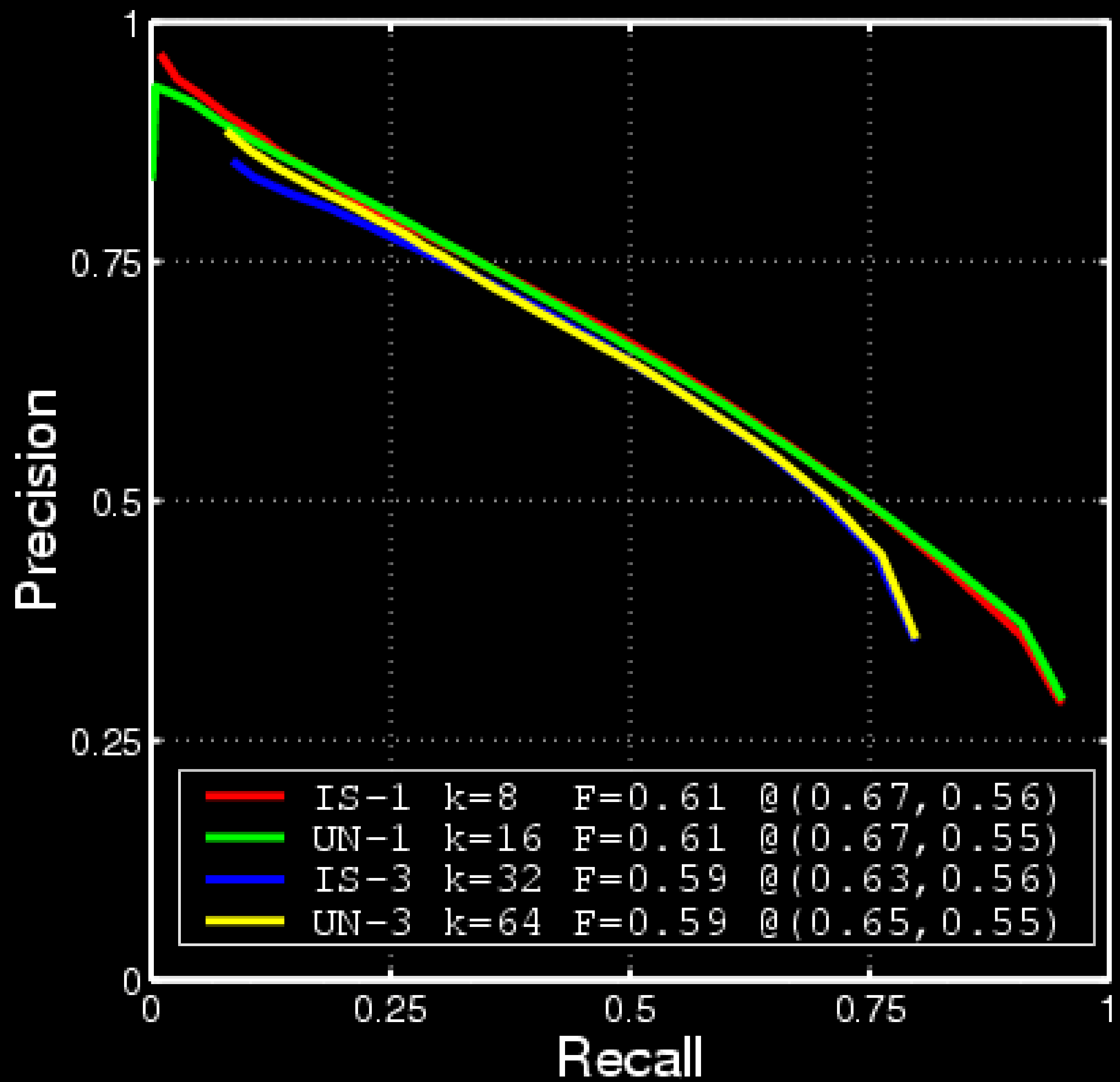
- ➢ All free parameters optimized on training data
- ➢ All algorithmic alternatives evaluated by experiment

- Brightness Gradient
  - Scale, bin/kernel sizes for KDE
- Color Gradient
  - Scale, bin/kernel sizes for KDE, joint vs. marginals
- Texture Gradient
  - Filter bank: scale, multiscale?
  - Histogram comparison: L^1, L^2, L^∞, χ^2, EMD
  - Number of textons, Image-specific vs. universal textons
- Localization parameters for each cue

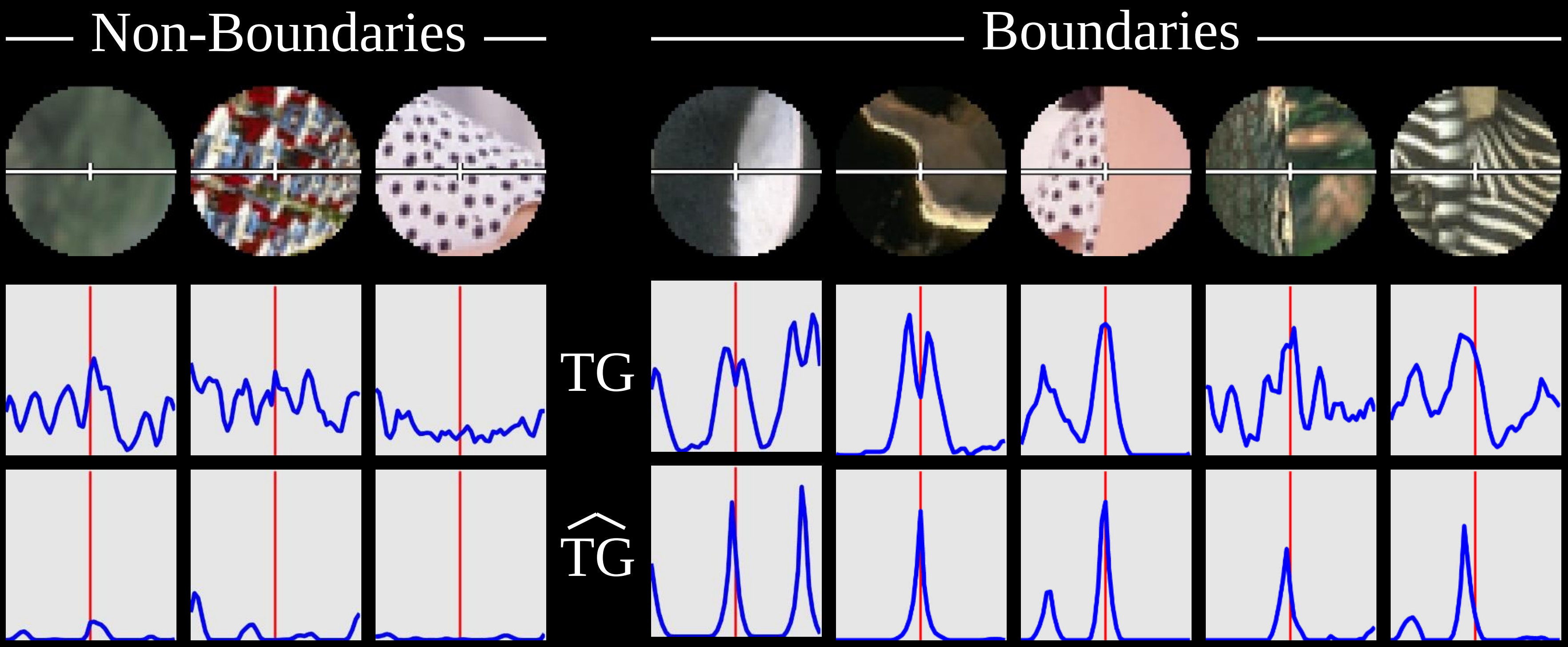
Calibration Example: Number of Textons for the Texture Gradient



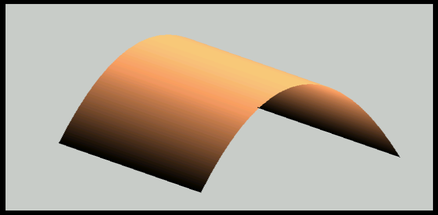
Calibration Example #2: Image-Specific vs. Universal Textons



Boundary Localization



(1) Fit cylindrical parabolas to raw oriented signal to get local shape: (Savitsky-Golay)



(2) Localize peaks:

Dataflow

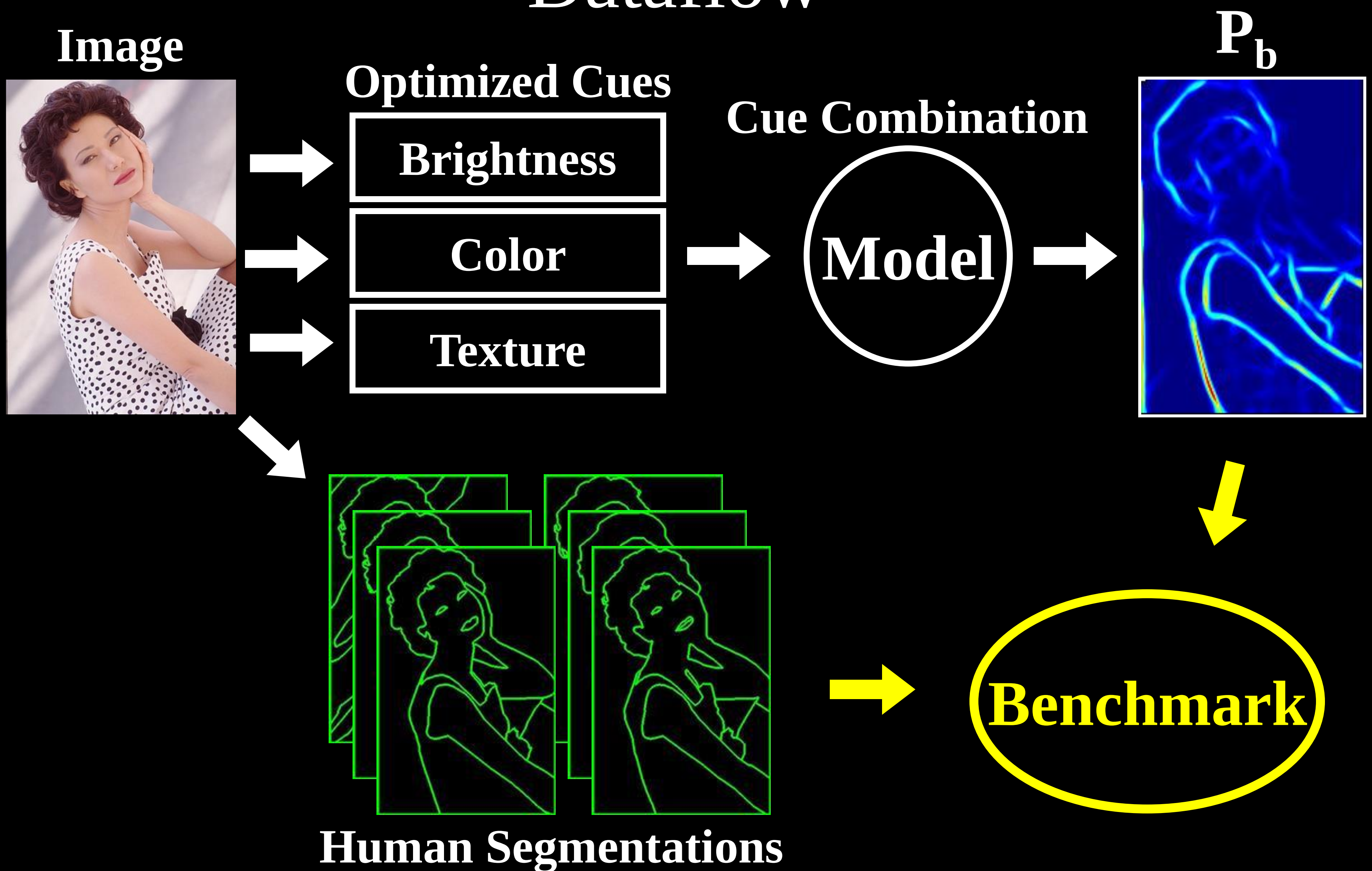
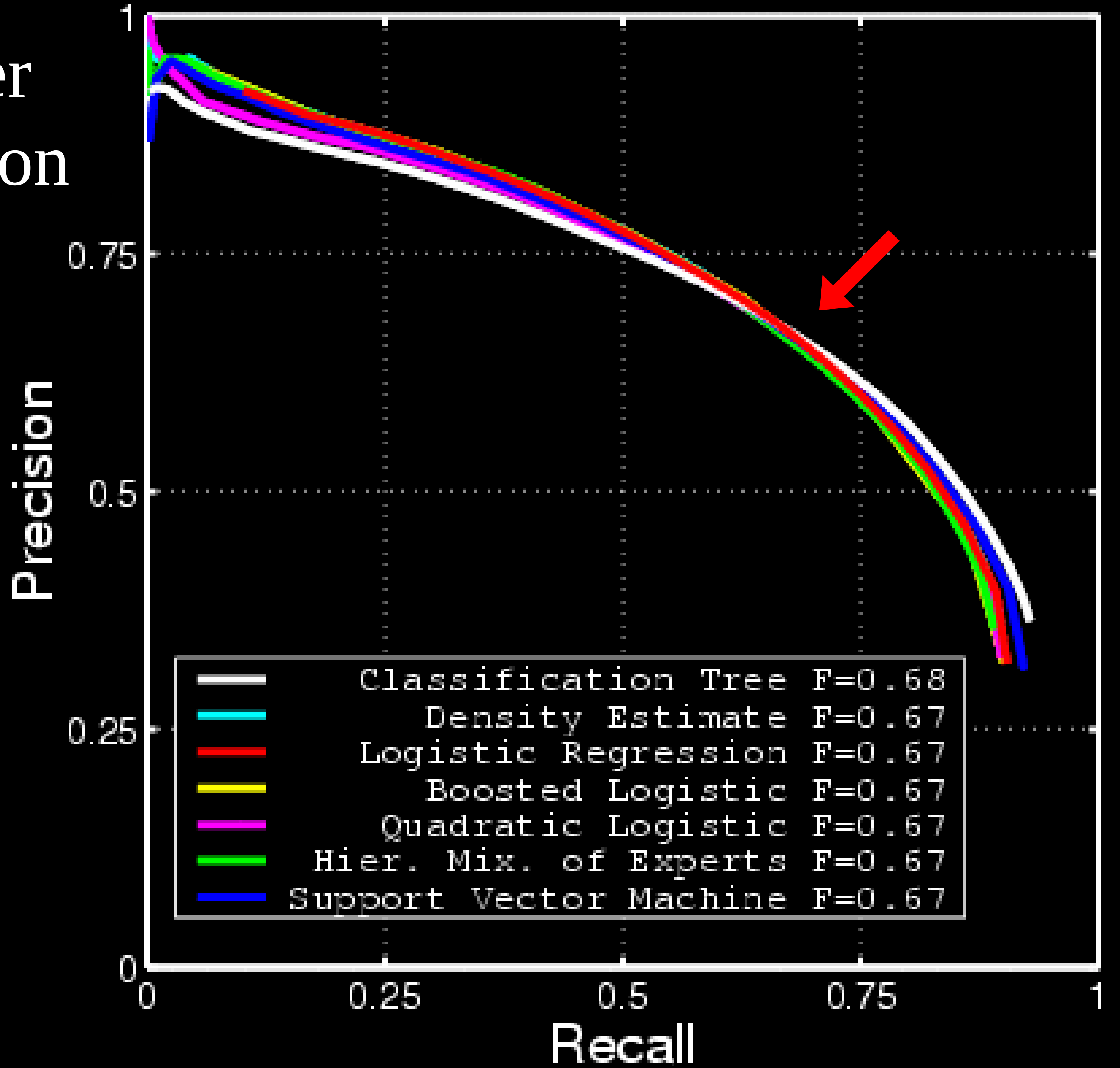
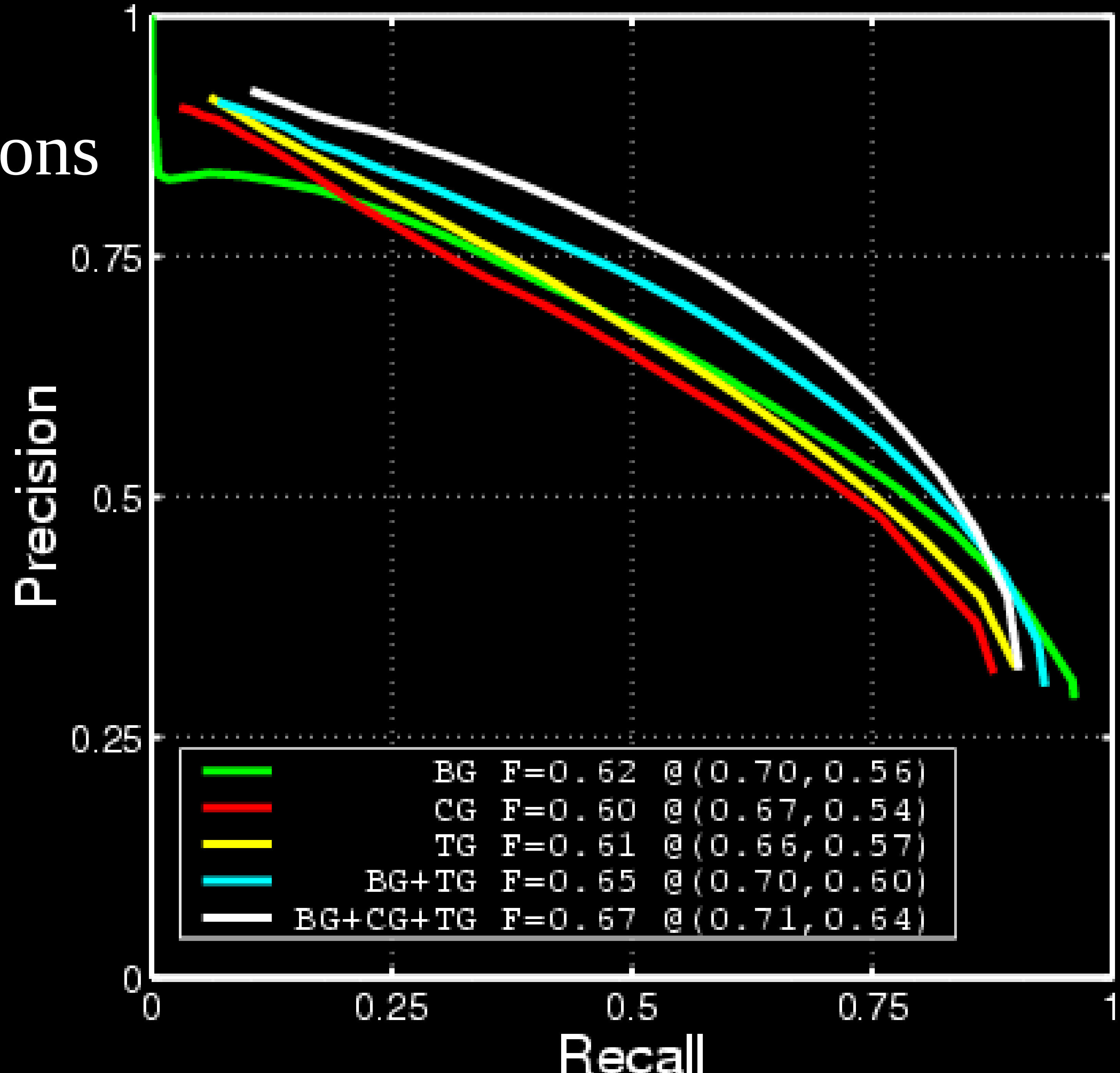


Image
Optimized Cues
Brightness
Color
Texture
Cue Combination
Model
P_b
Benchmark
Human Segmentations

Classifier Comparison



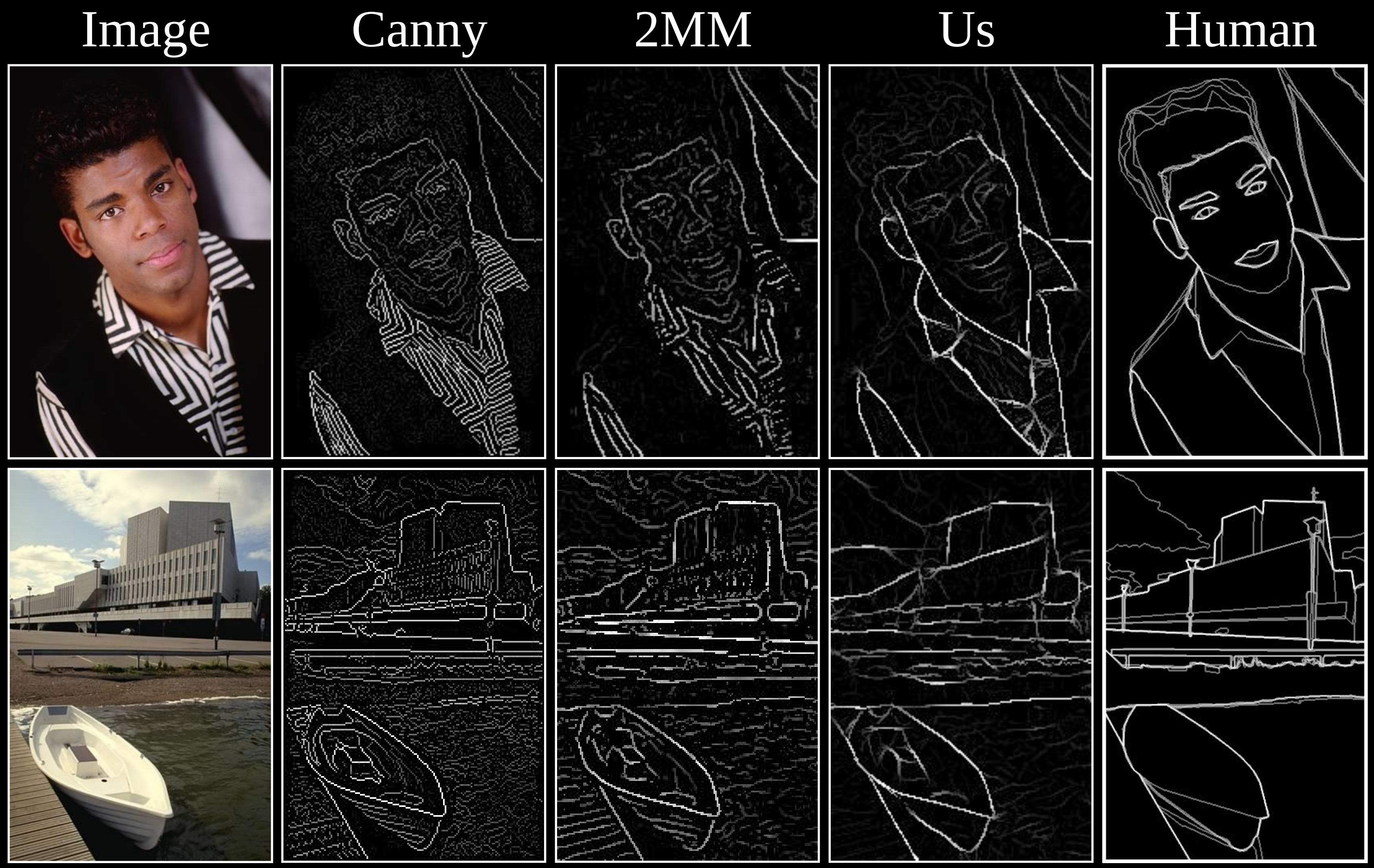
Cue Combinations



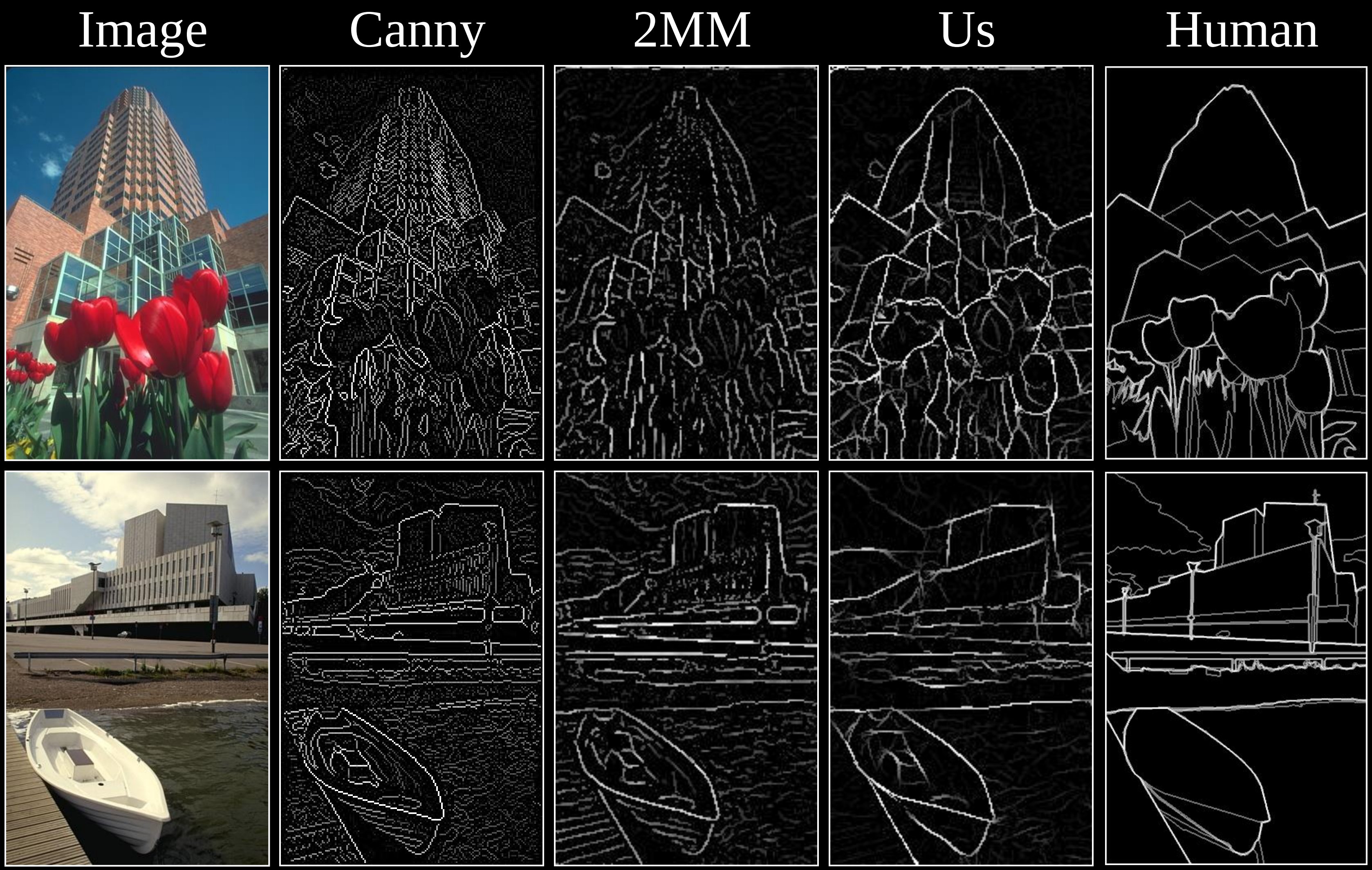
Alternate Approaches

- Canny Detector
  - Canny 1986
  - MATLAB implementation
  - With and without hysteresis
- Second Moment Matrix
  - Nitzberg/Mumford/Shiota 1993
  - cf. Förstner and Harris corner detectors
  - Used by Konishi et al. 1999 in learning framework
  - Logistic model trained on full eigenspectrum

P_b Images

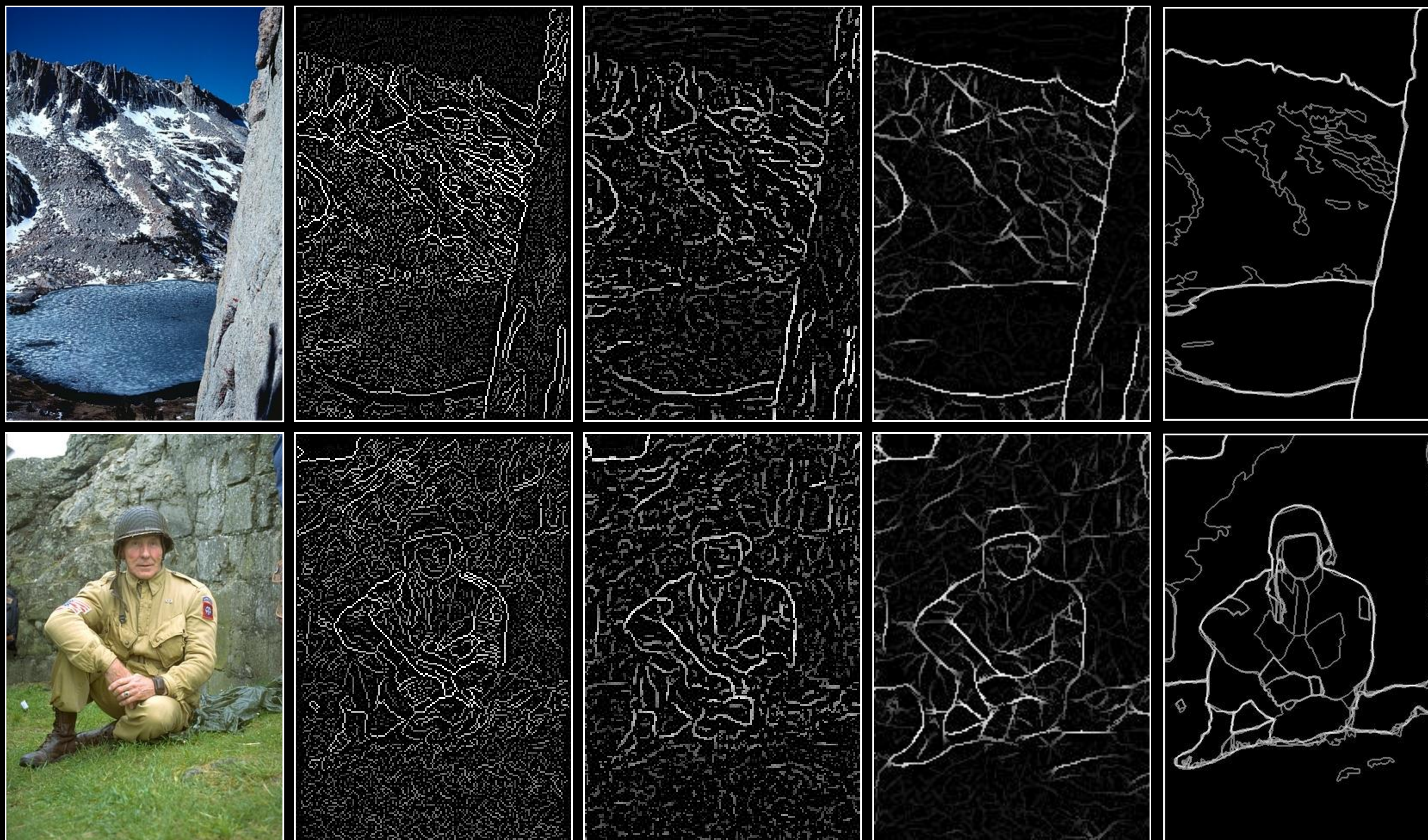


P_b Images II

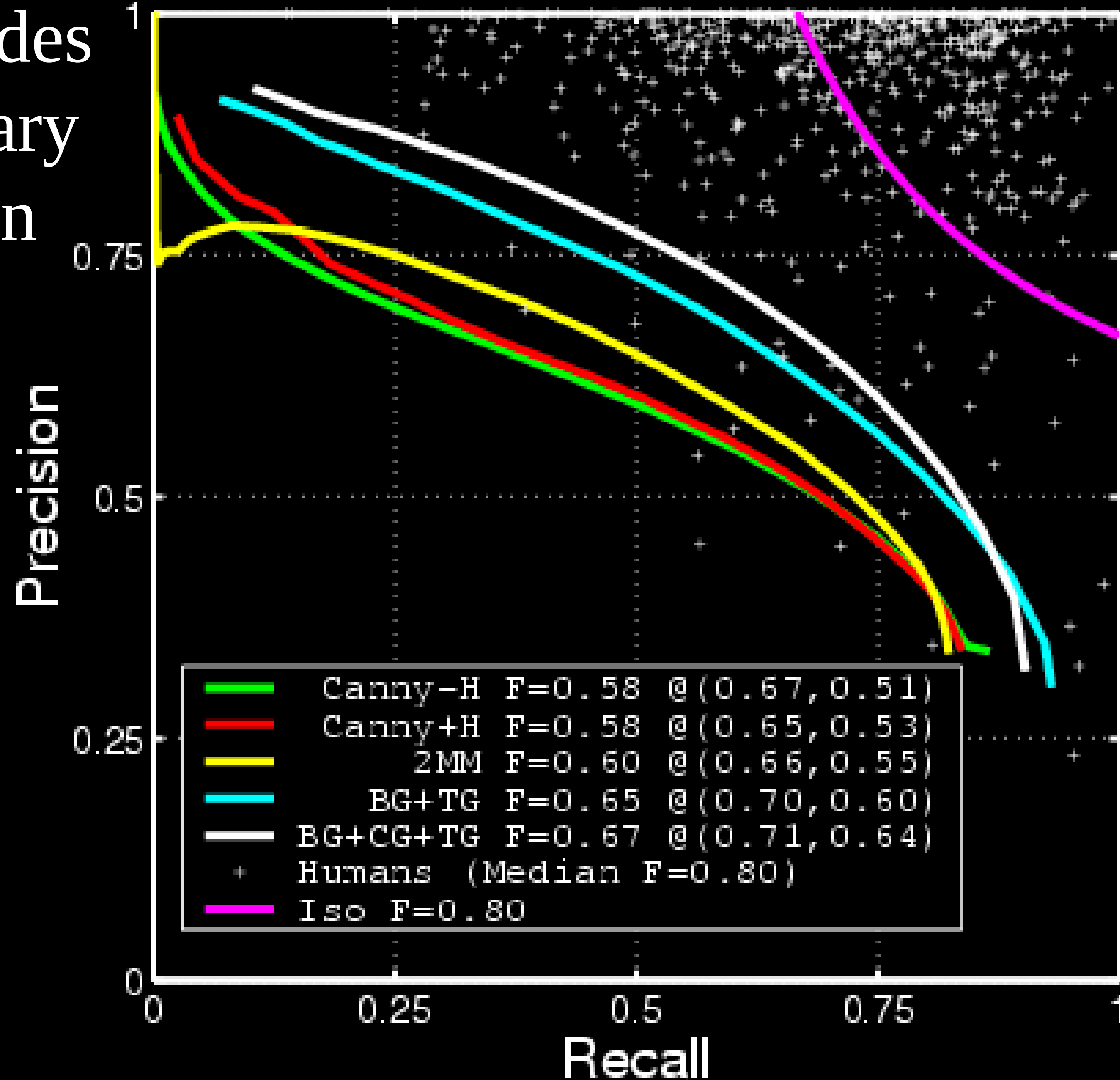


P_b Images III

Image | Canny | 2MM | Us | Human



Two Decades of Boundary Detection



Findings

1. A simple linear model is sufficient for cue combination
   – All cues weighted approximately equally in logistic
2. Proper texture edge model is not optional for complex natural images
   – Texture suppression is not sufficient!
3. Significant improvement over state-of-the-art in boundary detection
   – $P_b(x,y,\theta)$ useful for higher-level processing
4. Empirical approach critical for both cue calibration and cue combination