Boundaries and Sketches

Szeliski 4.2

Computer Vision

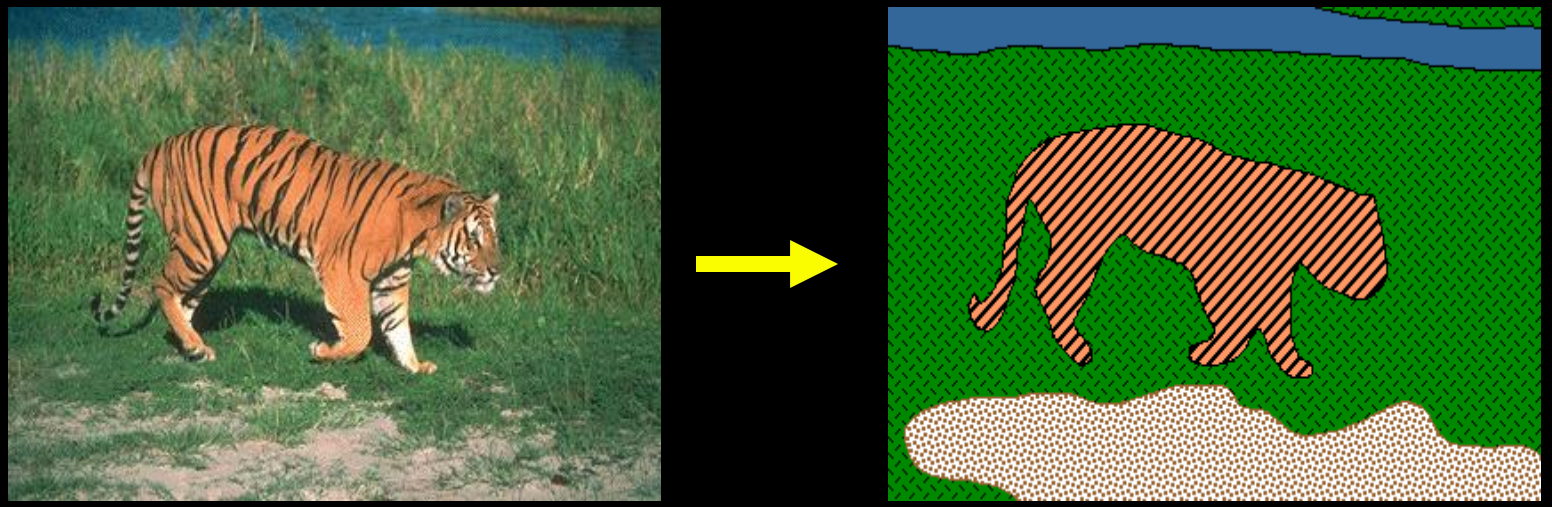
James Hays

Many slides from Michael Maire, Jitendra Malek

Today's lecture

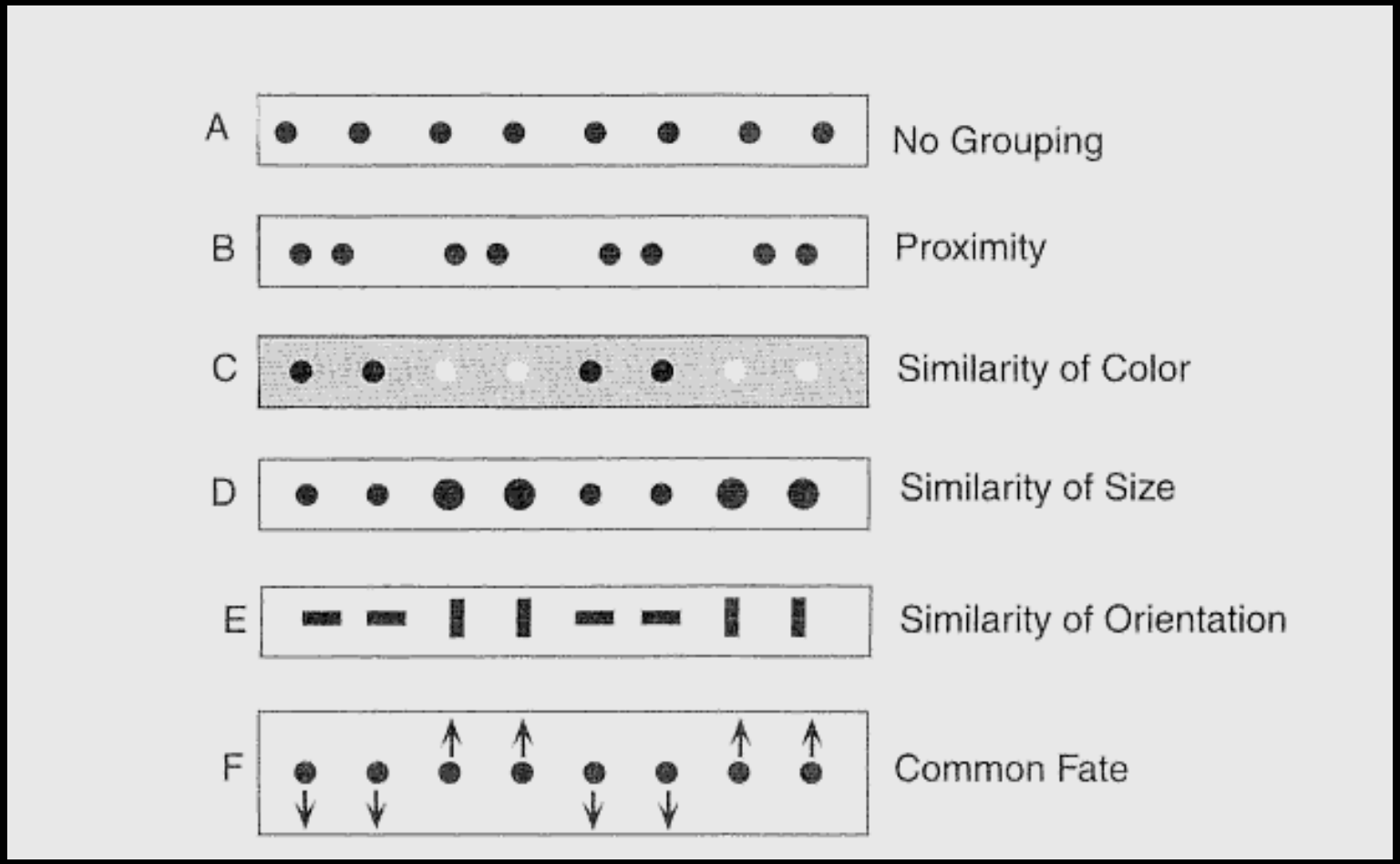
- Segmentation vs Boundary Detection
- Why boundaries / Grouping?
- Recap: Canny Edge Detection
- The Berkeley Segmentation Data Set
- pB boundary detector ~2001
- Sketch Tokens 2013
- How do Humans Sketch Objects?

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, 1923

Grouping factors



Recap: 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

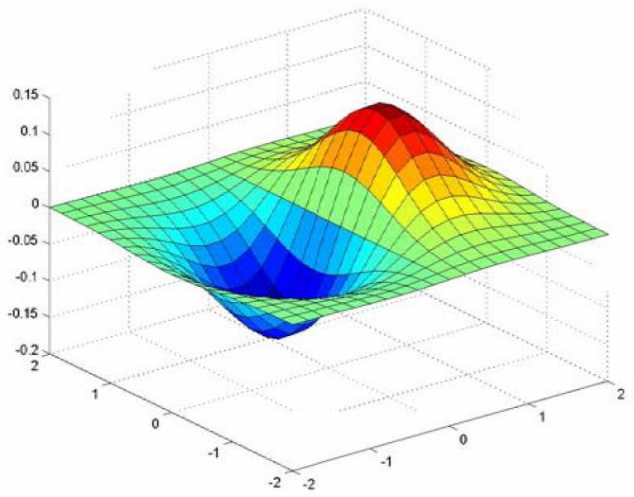
J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Source: L. Fei-Fei

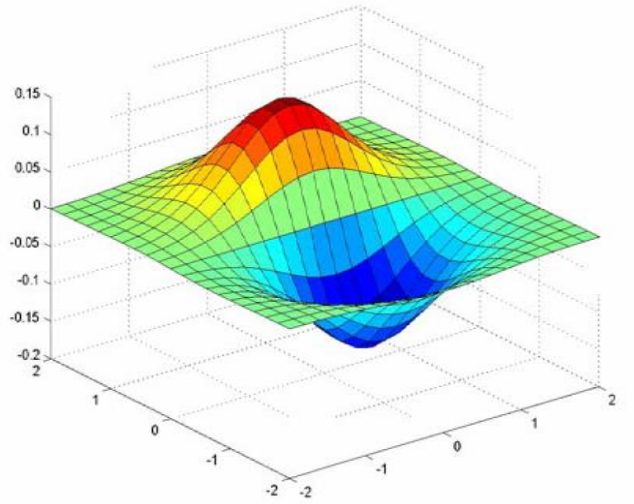
Example

original image (Lena)

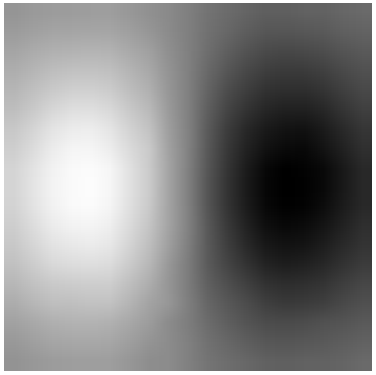
Derivative of Gaussian filter

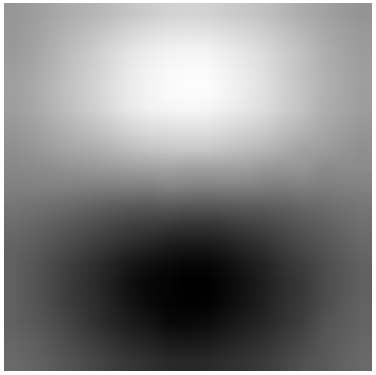


x-direction

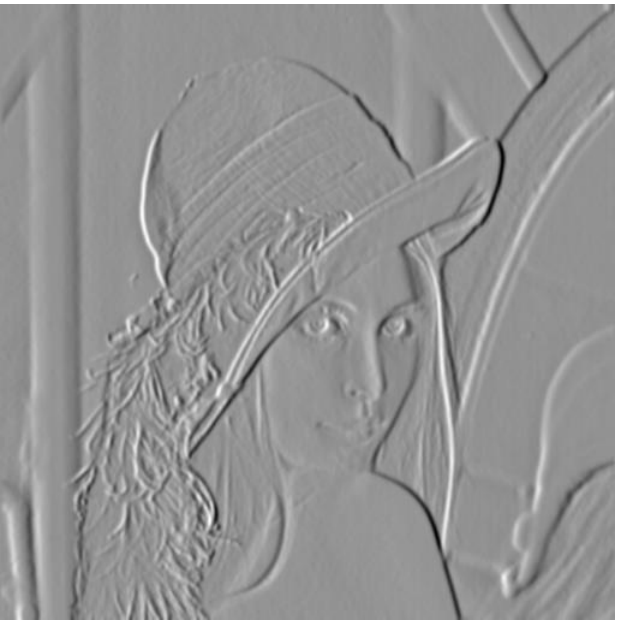


y-direction





Compute Gradients (DoG)



X-Derivative of Gaussian

Y-Derivative of Gaussian

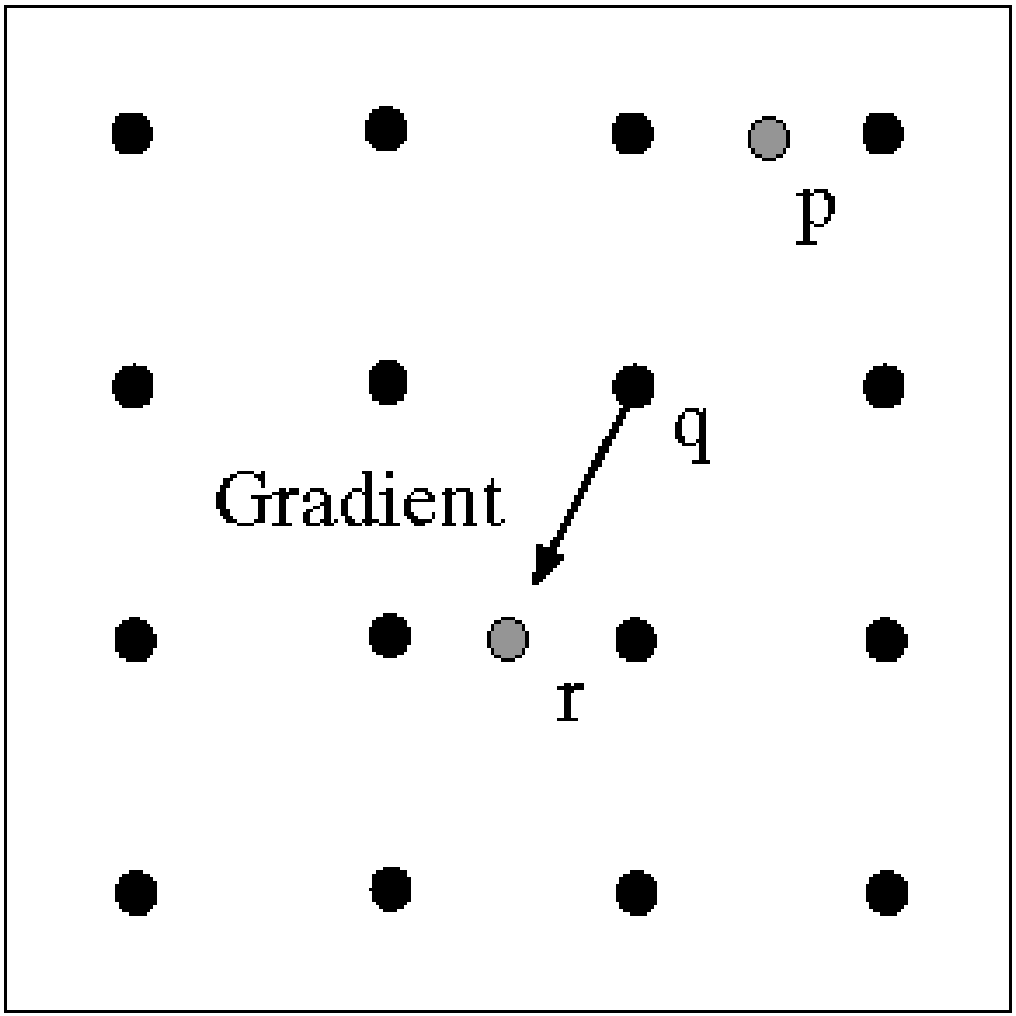
Gradient Magnitude

Get Orientation at Each Pixel

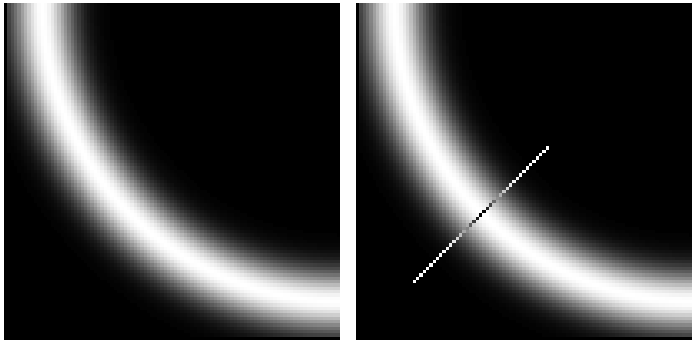
- Threshold at minimum level
- Get orientation

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

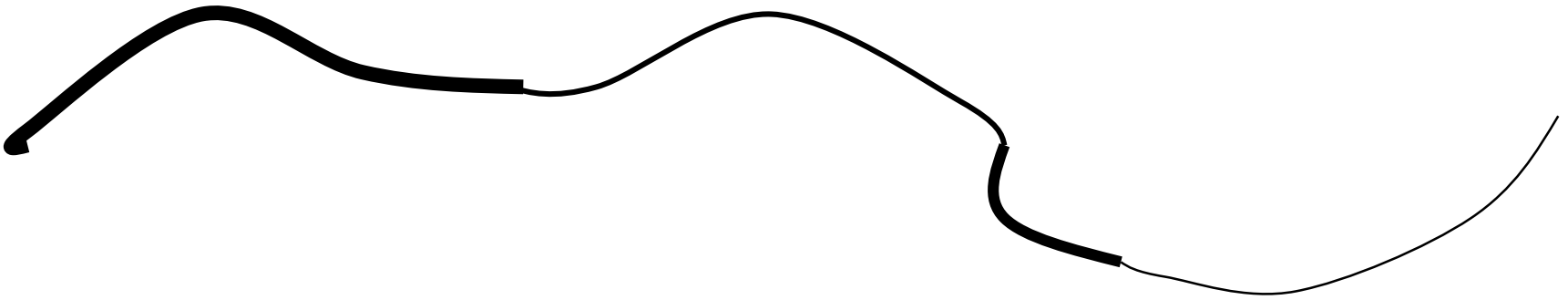
After non-max suppression

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

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

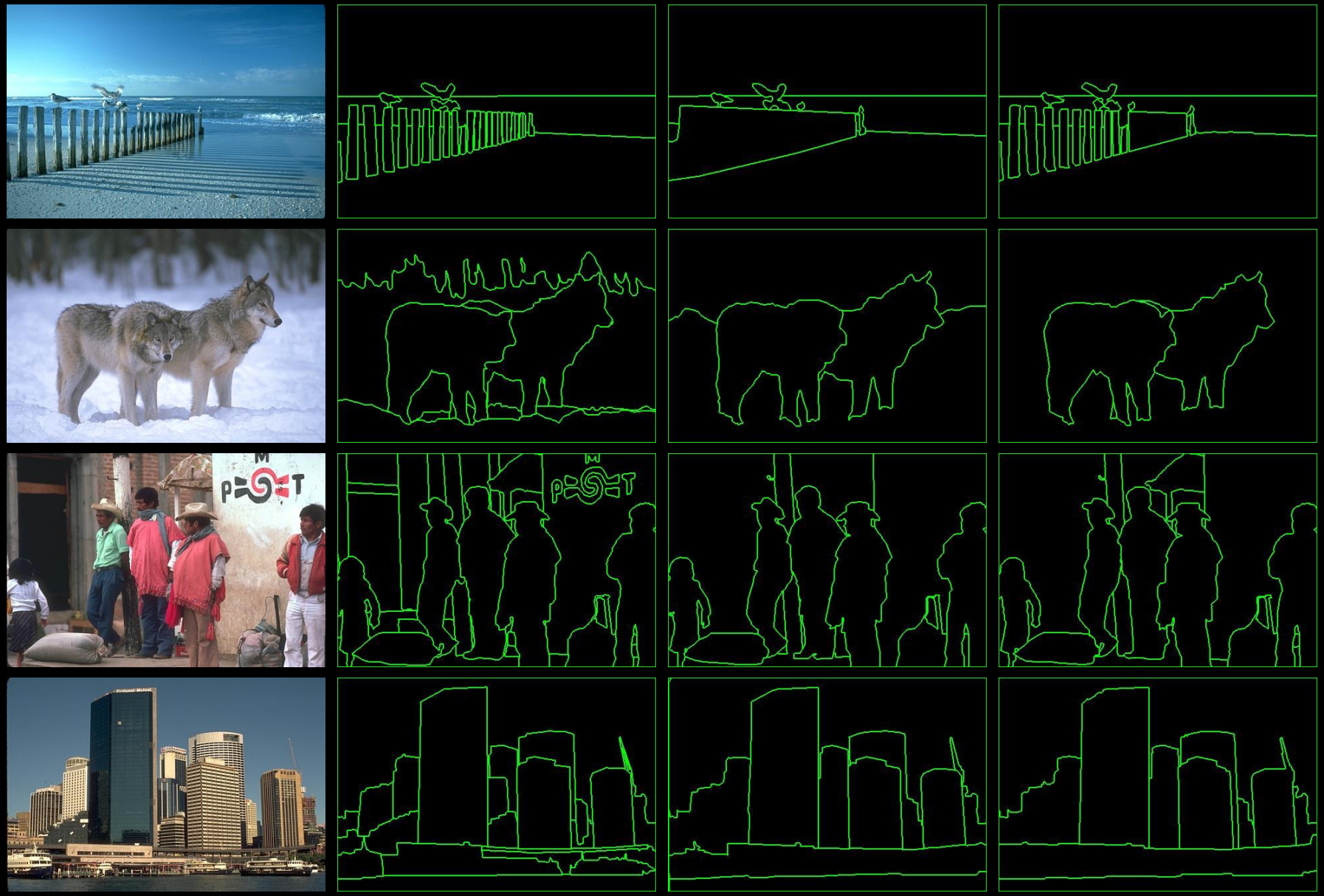
UC Berkeley

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

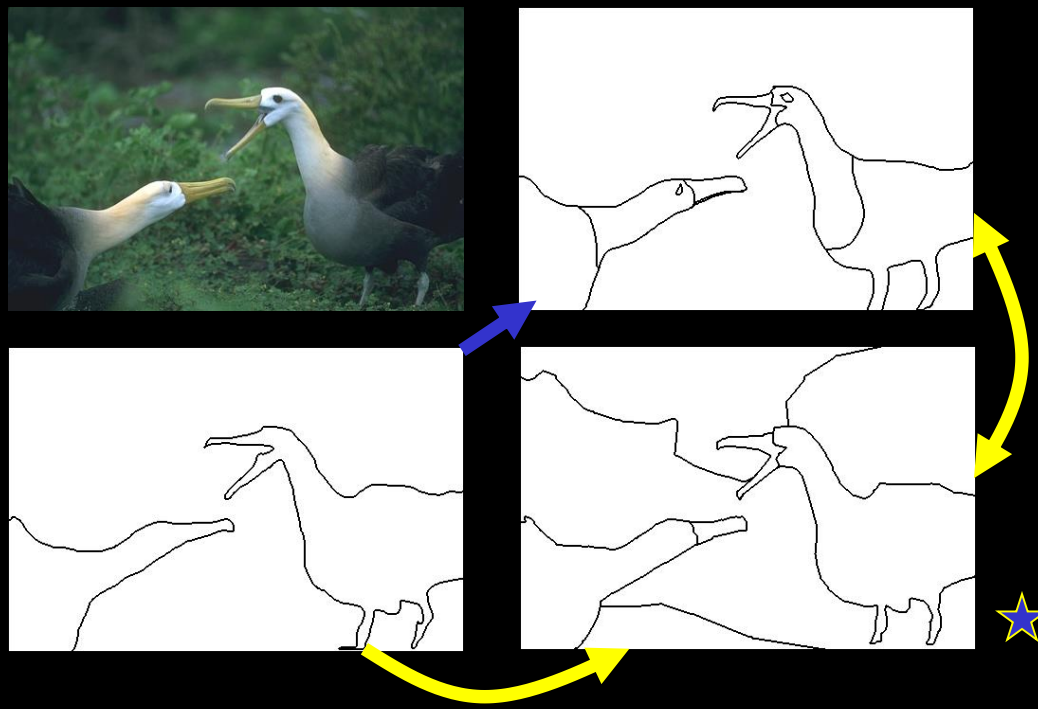
Protocol

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)

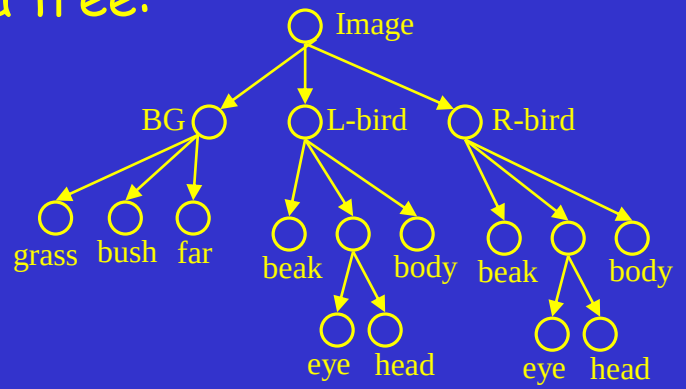


Segmentations are Consistent



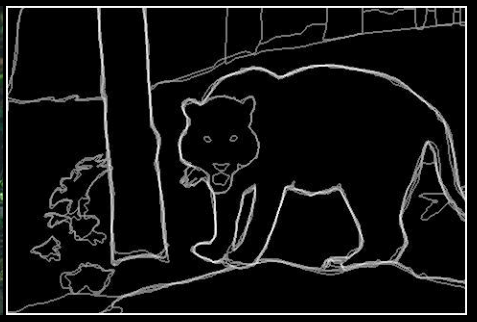
- A,C are refinements of B
- A,C are mutual refinements
- A,B,C represent the same percept
  - Attention accounts for differences

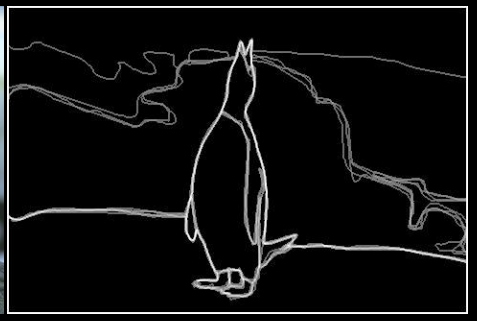
Perceptual organization forms a tree:



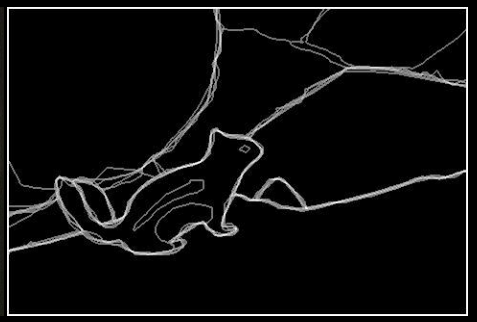
☆ Two segmentations are consistent when they can be explained by the same segmentation tree (i.e. they could be derived from a single perceptual organization).



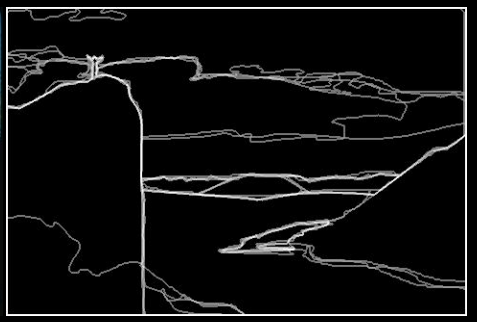










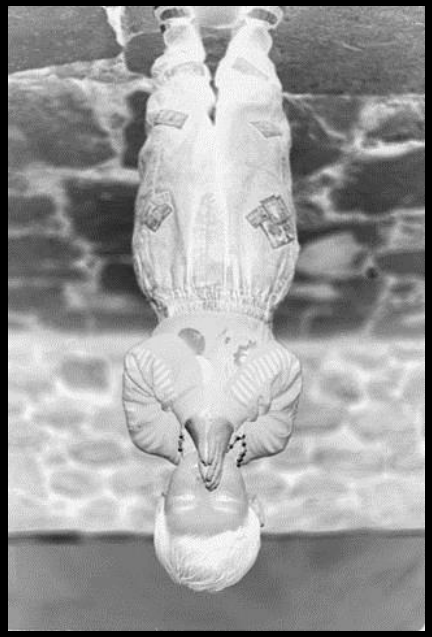
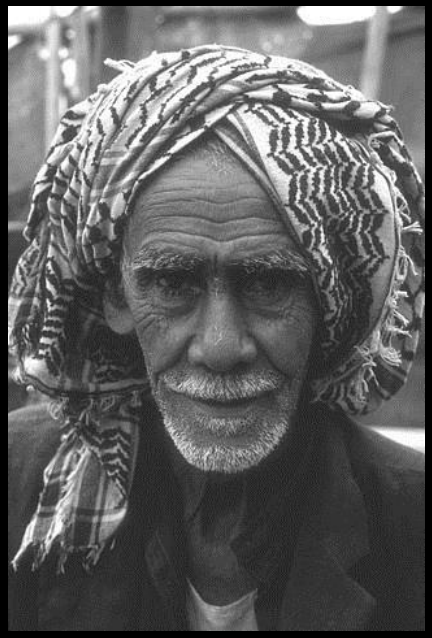
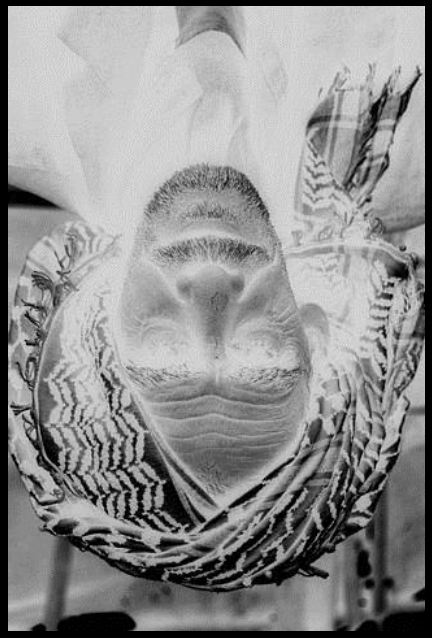


Dataset Summary

- 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

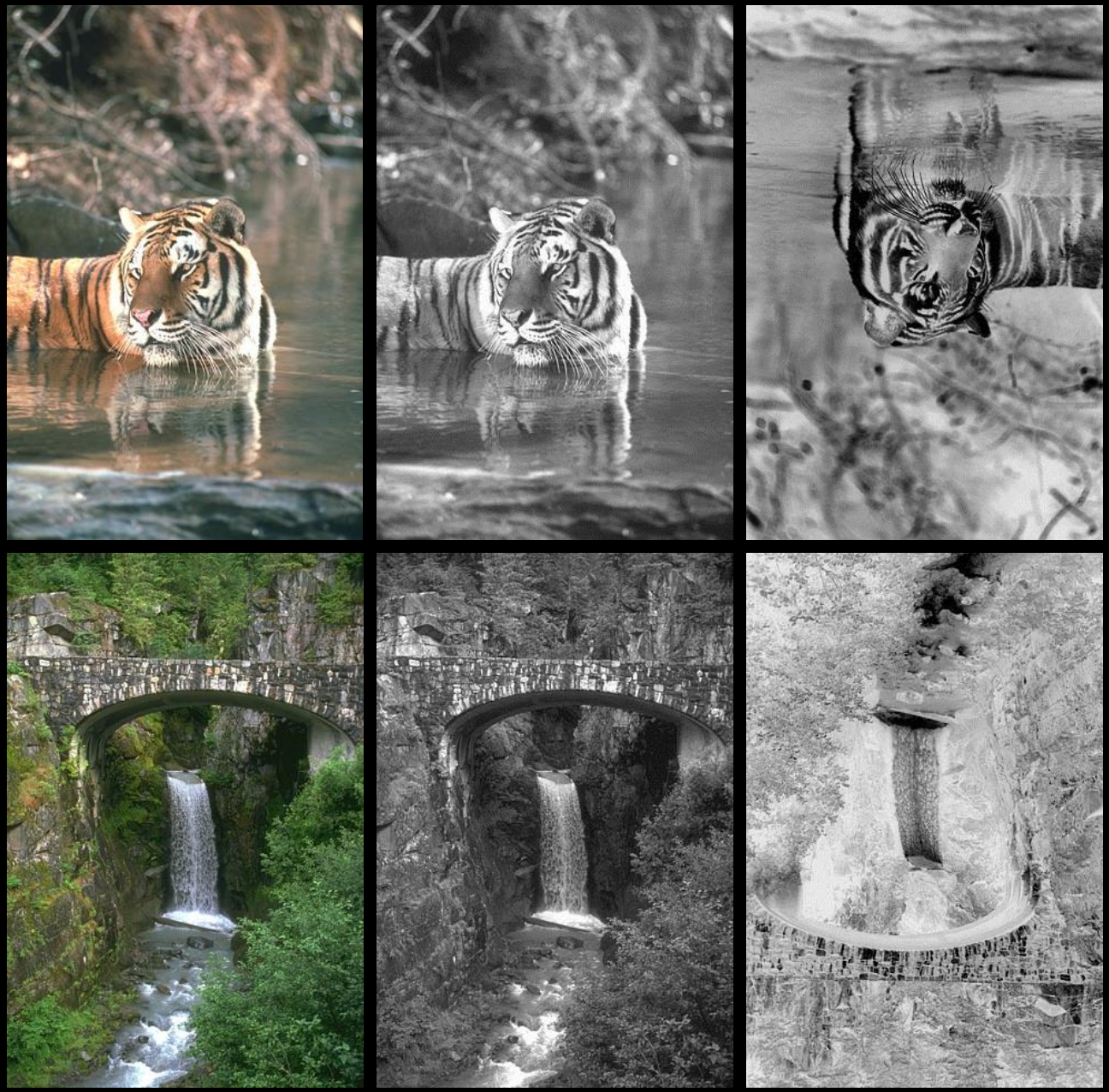
Gray, Color, InvNeg Datasets

- 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
		
		

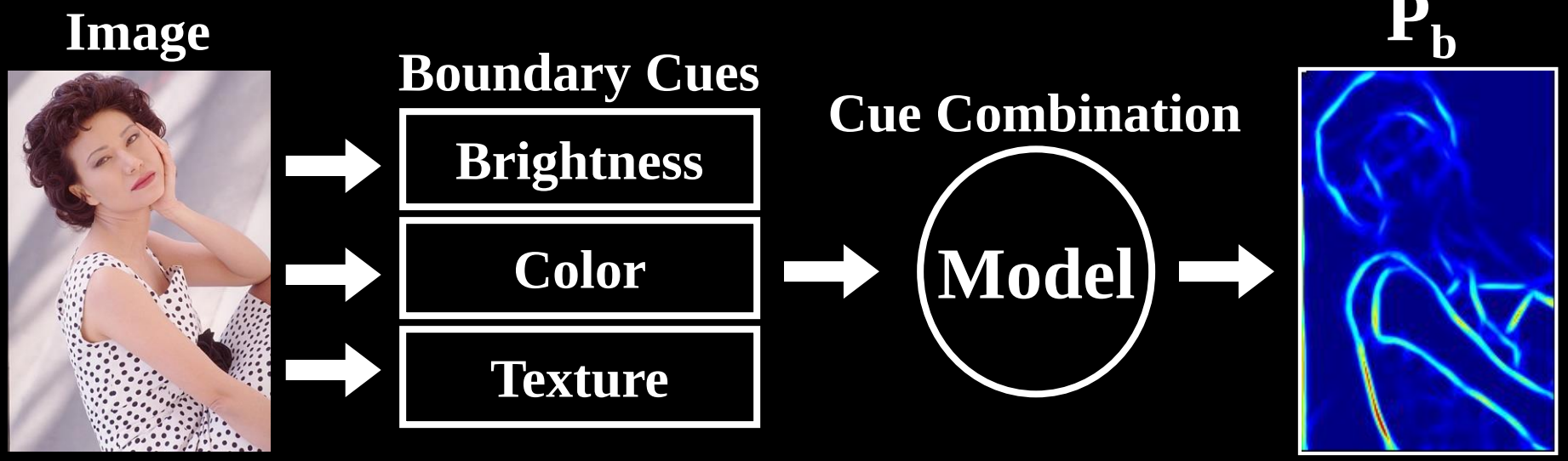
InvNeg

Color Gray InvNeg



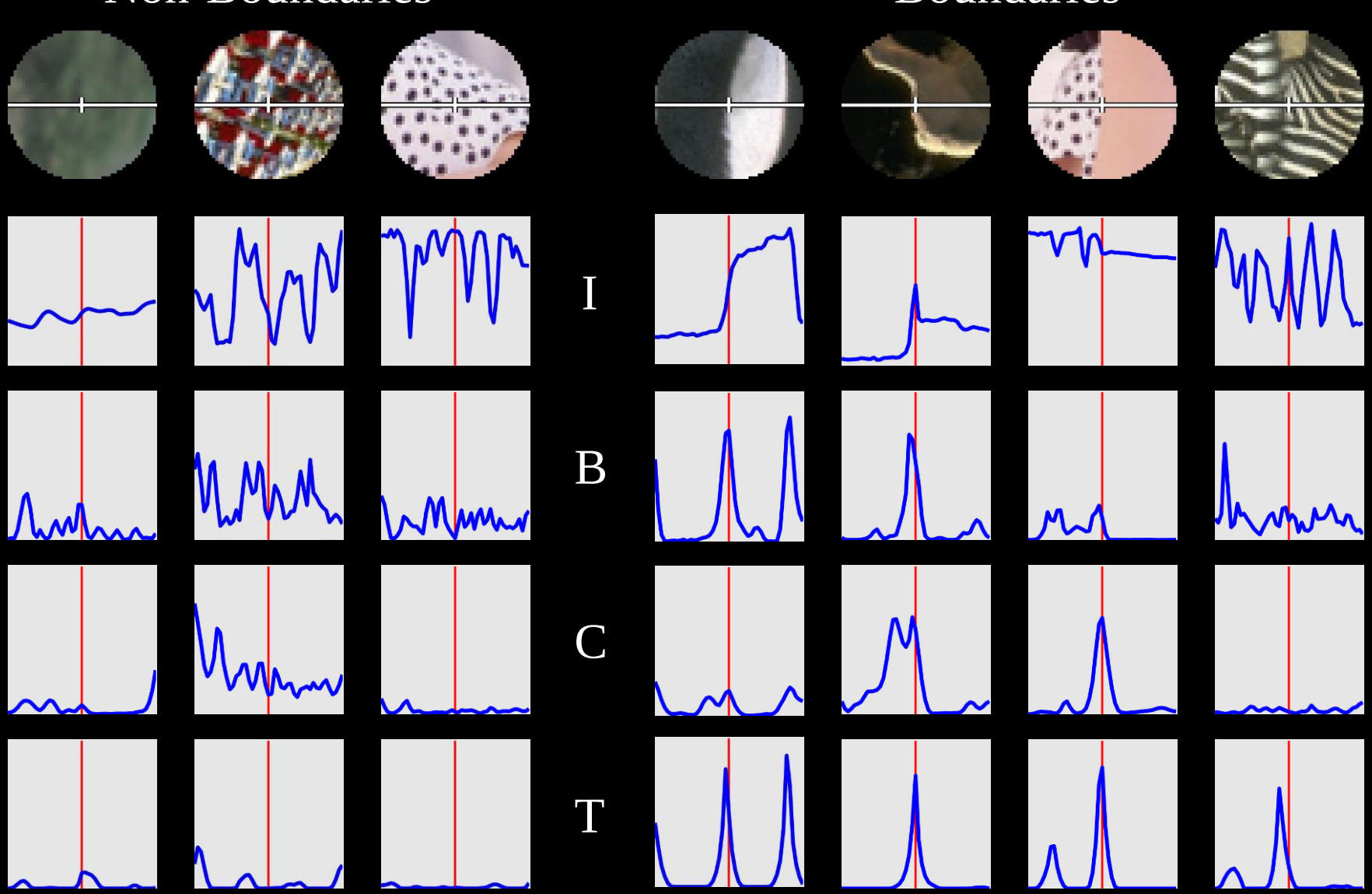
Pb Detector

Dataflow



Challenges: texture cue, cue combination

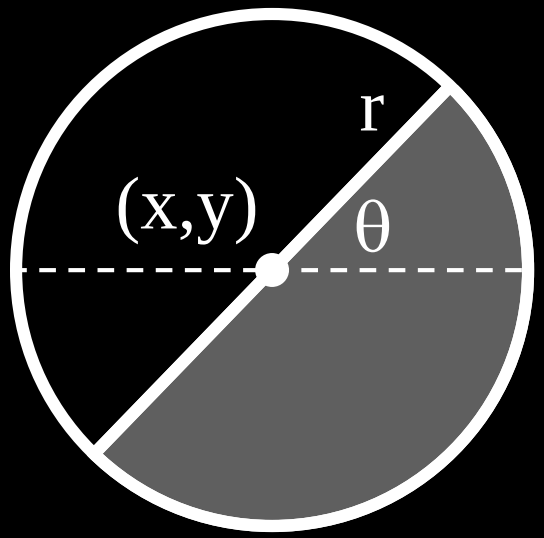
Goal: learn the posterior probability of a boundary $P_b(x,y,\theta)$ from local information only


Non-Boundaries
Boundaries
I
B
C
T

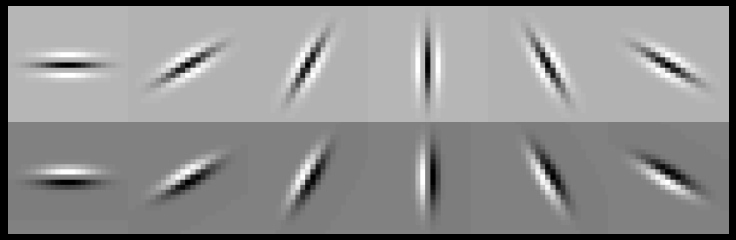
Brightness and Color Features

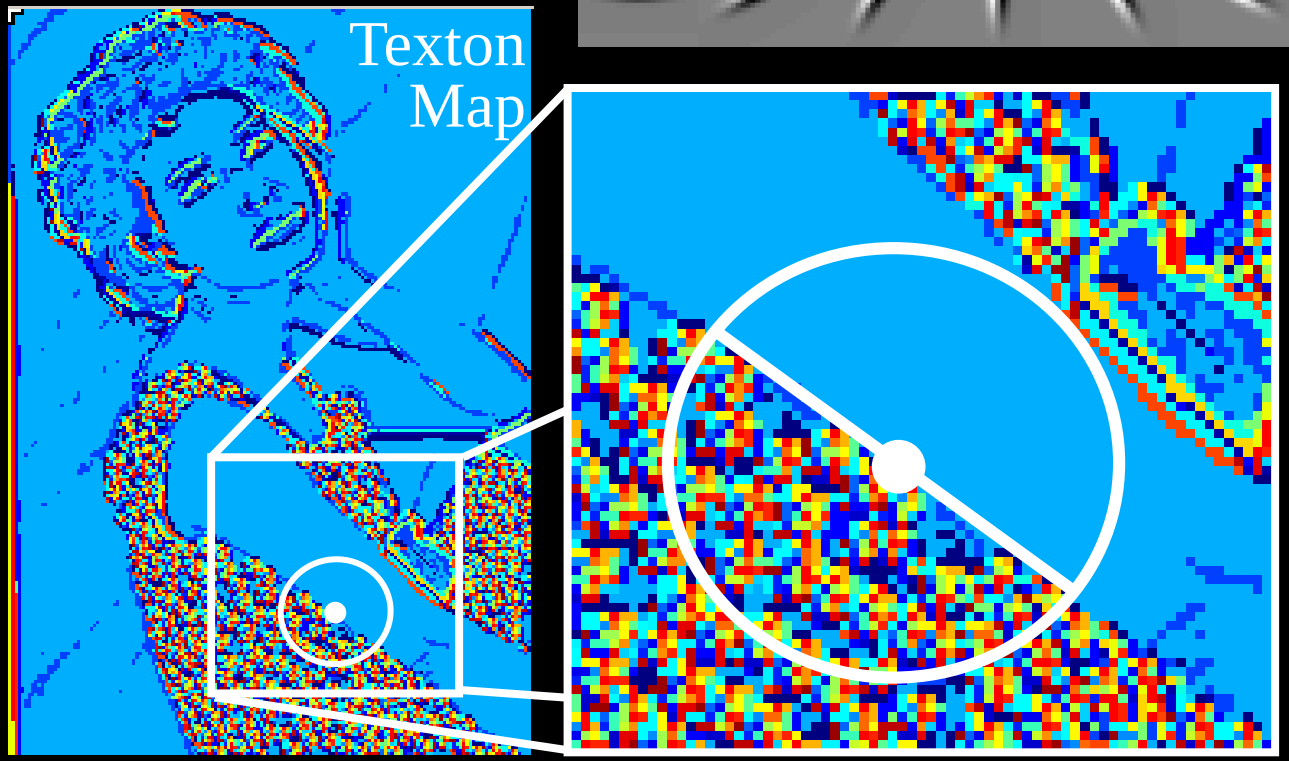
- 1976 CIE L*a*b* colorspace
- Brightness Gradient BG(x,y,r,θ)
  - χ^2 difference in L* distribution
- Color Gradient CG(x,y,r,θ)
  - χ^2 difference in a* and b* distributions

$$\chi^2(g,h) = \frac{1}{2}\sum_i \frac{(g_i - h_i)^2}{g_i + h_i}$$



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

Dataflow

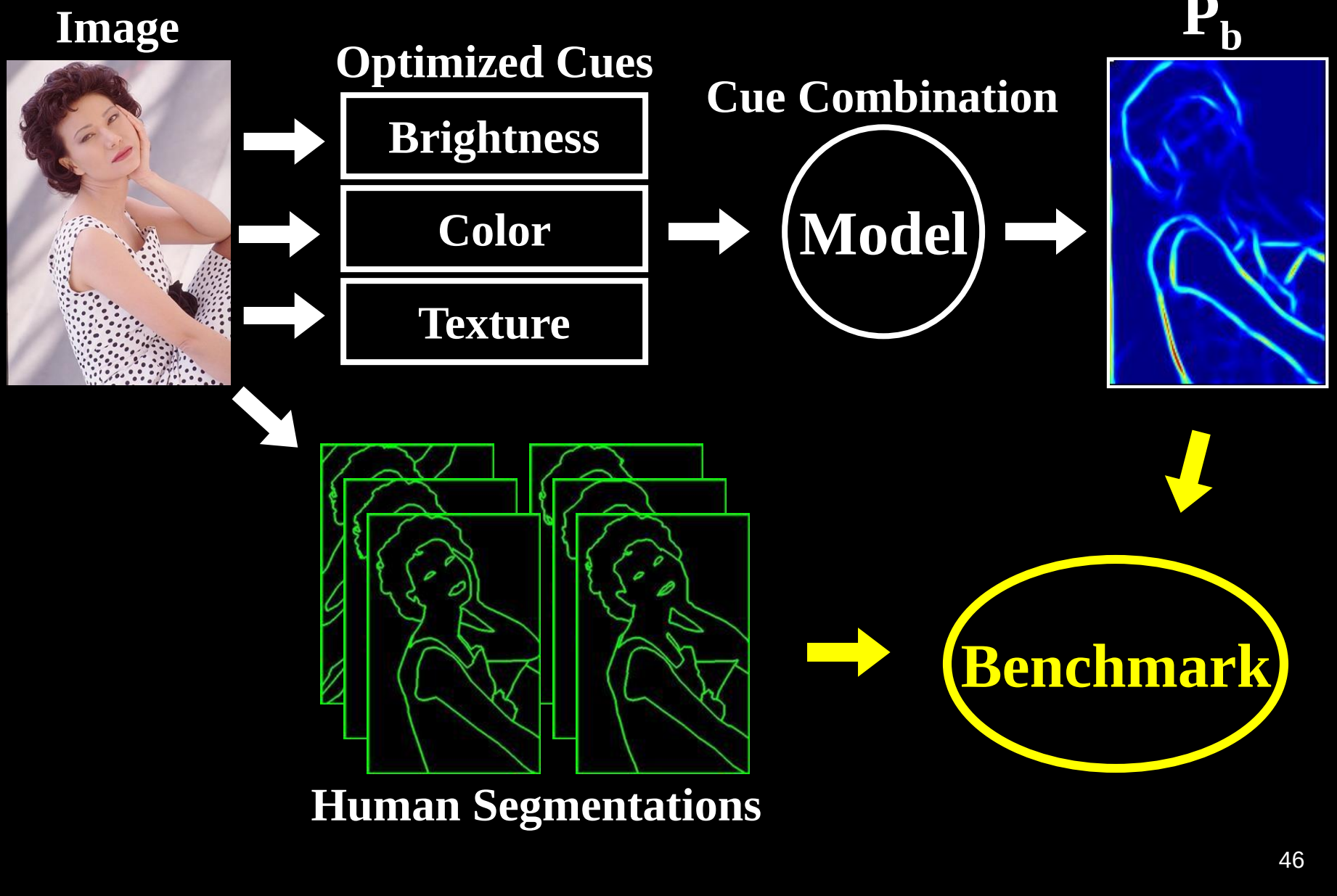
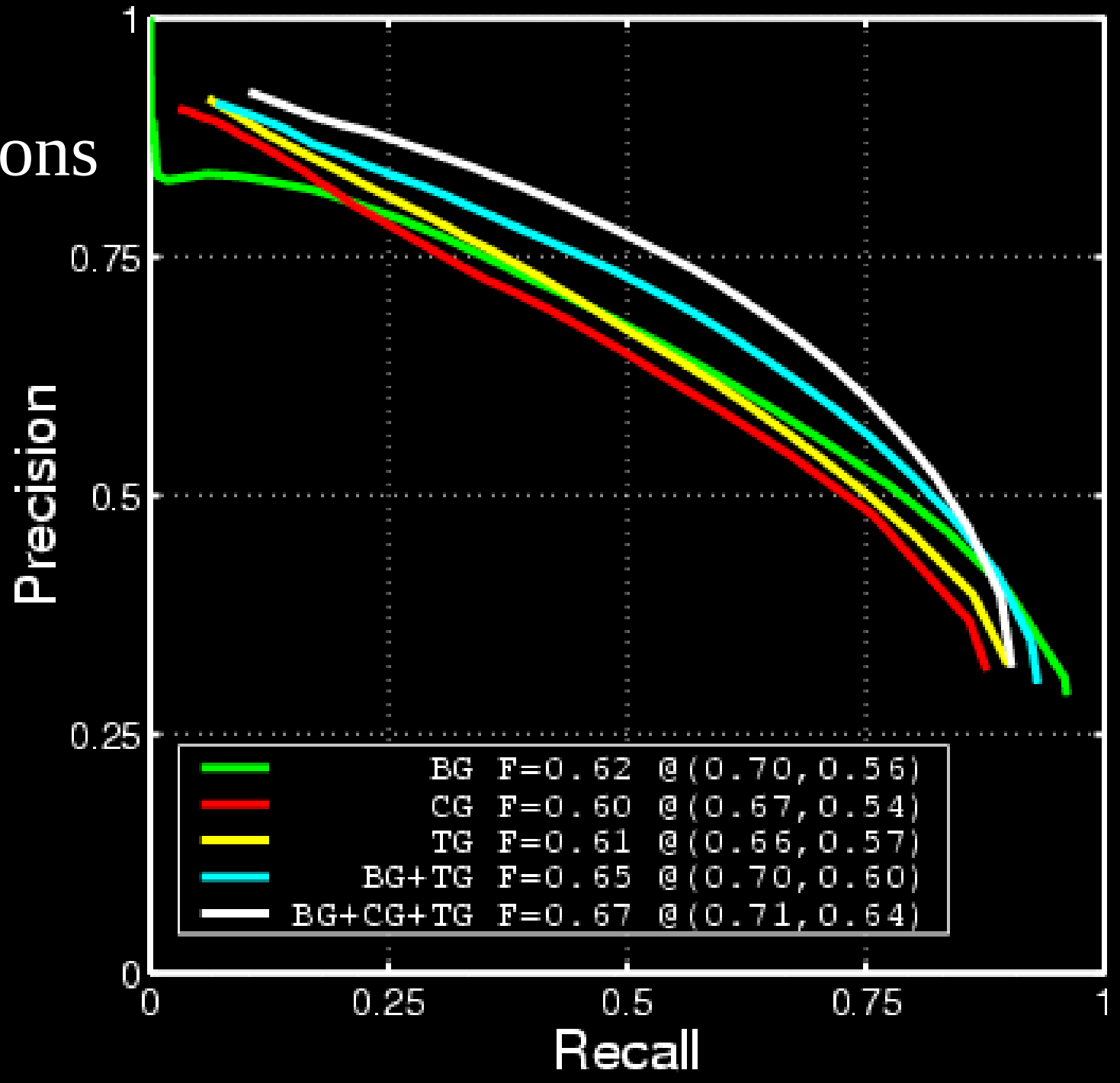


Image
Optimized Cues
Brightness
Color
Texture
Cue Combination
Model
P_b
Human Segmentations
Benchmark

Cue Combinations

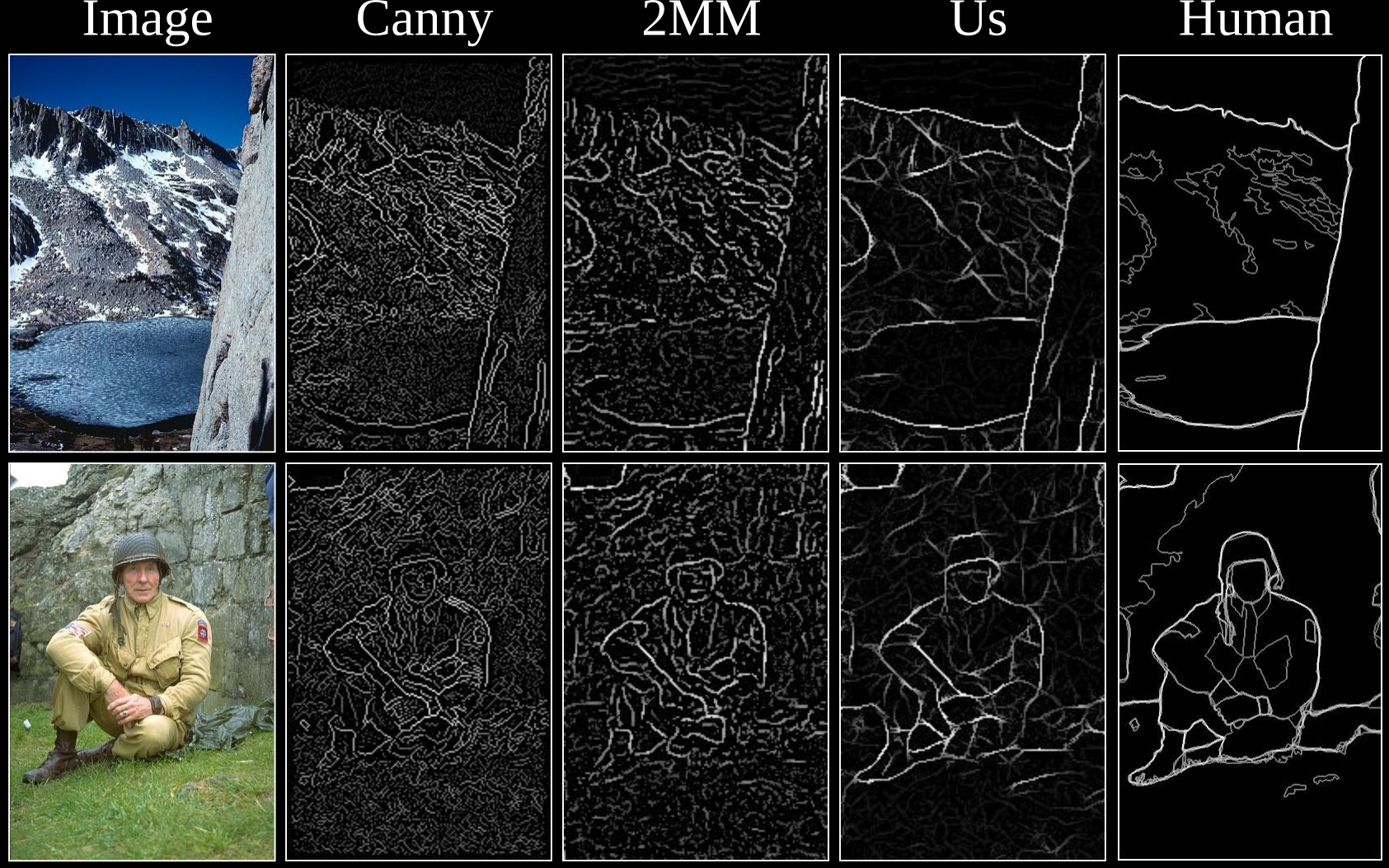


Alternate Approaches

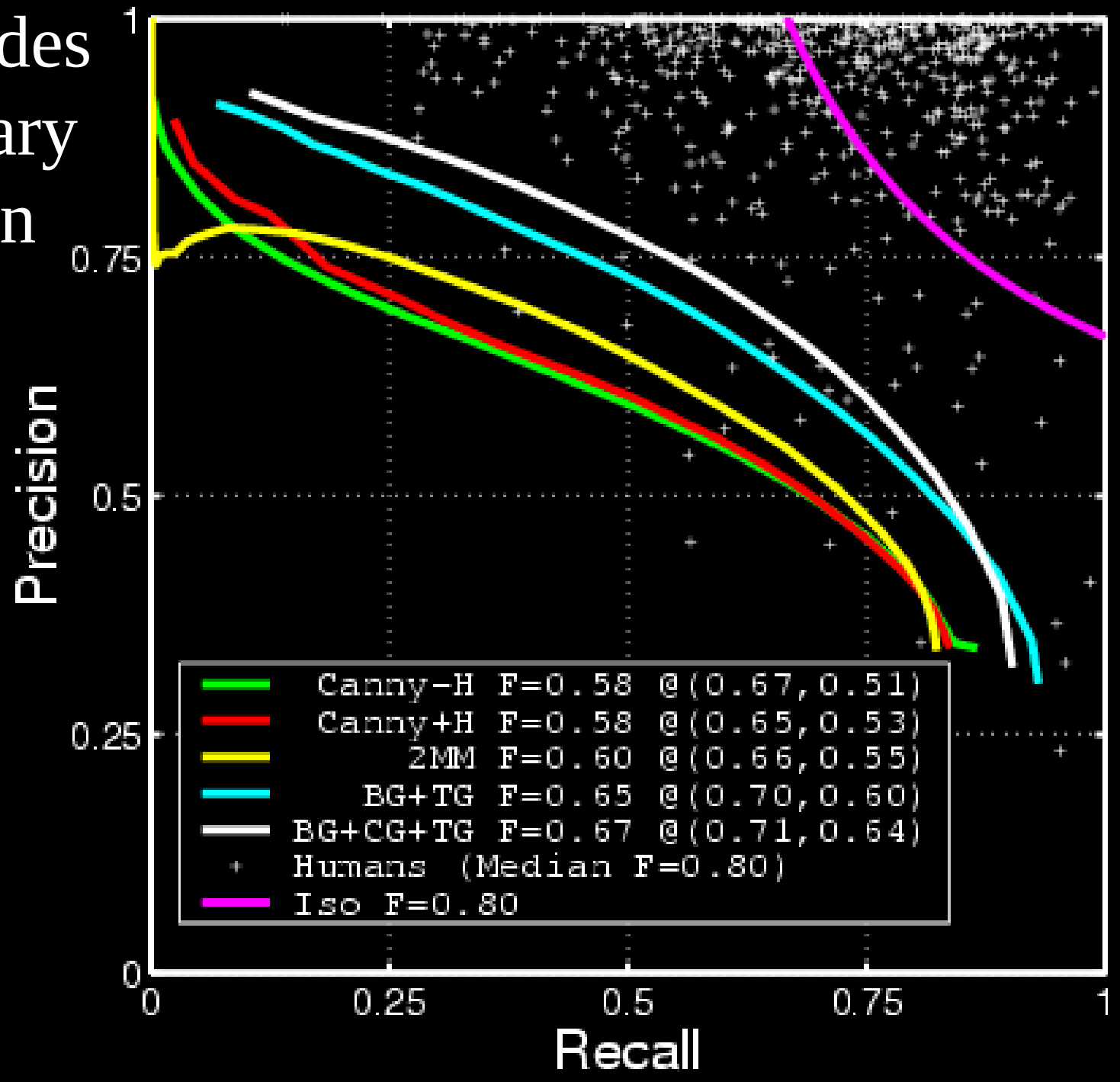
- Canny Detector
  - Canny 1986
  - MATLAB implementation
  - With and without hysteresis
- Second Moment Matrix
  - Nitzberg/Mumford/Shiota 1993
  - cf. Förstner and Harris corner detectors
  - Used by Konishi et al. 1999 in learning framework
  - Logistic model trained on full eigenspectrum

P_b Images

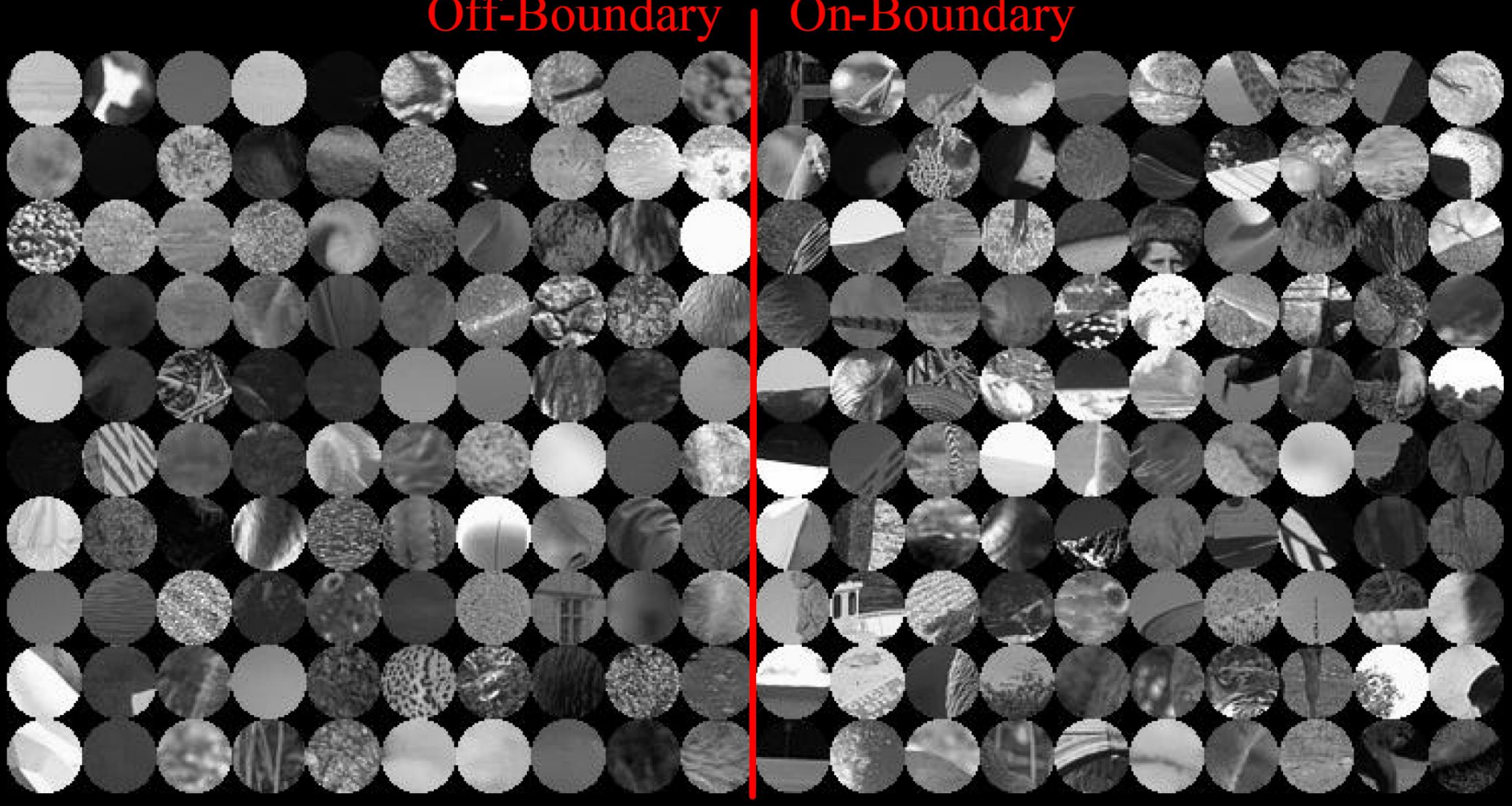
P_b Images II



Two Decades of Boundary Detection

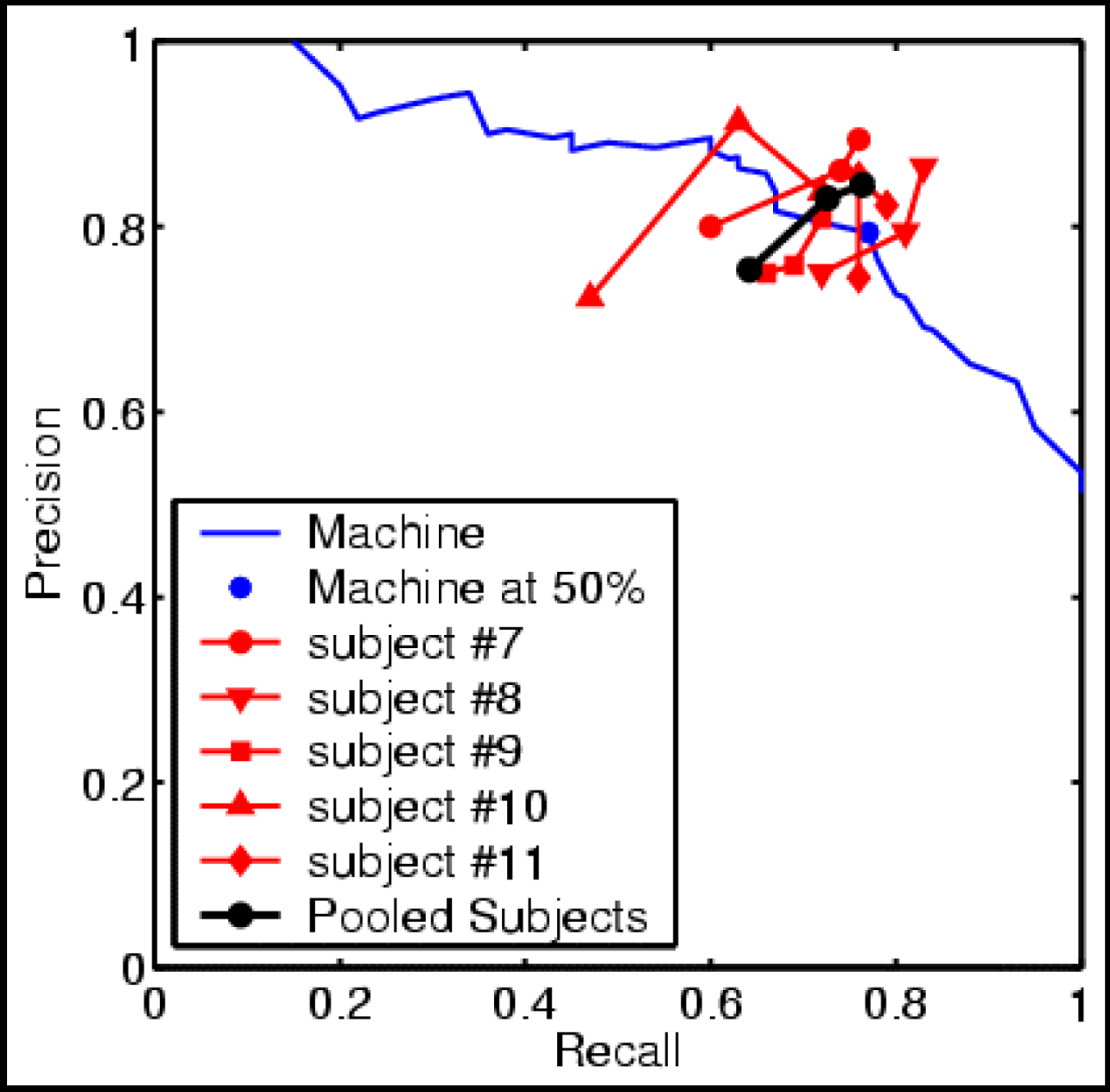


How good are humans locally?



- Algorithm: r = 9, Humans: r = {5,9,18}
- Fixation(2s) -> Patch(200ms) -> Mask(1s)

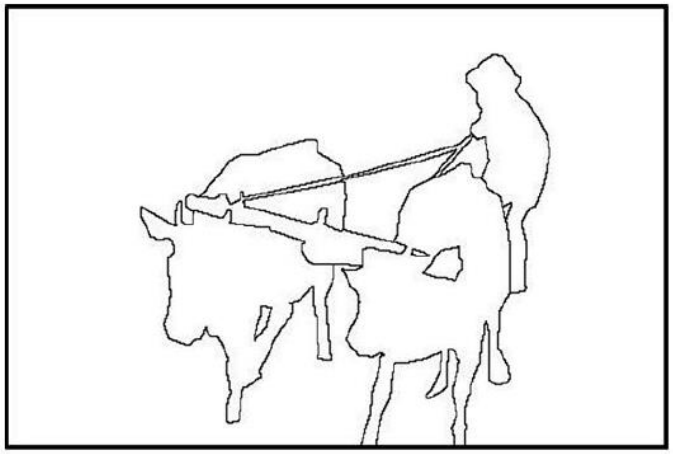
Man versus Machine:

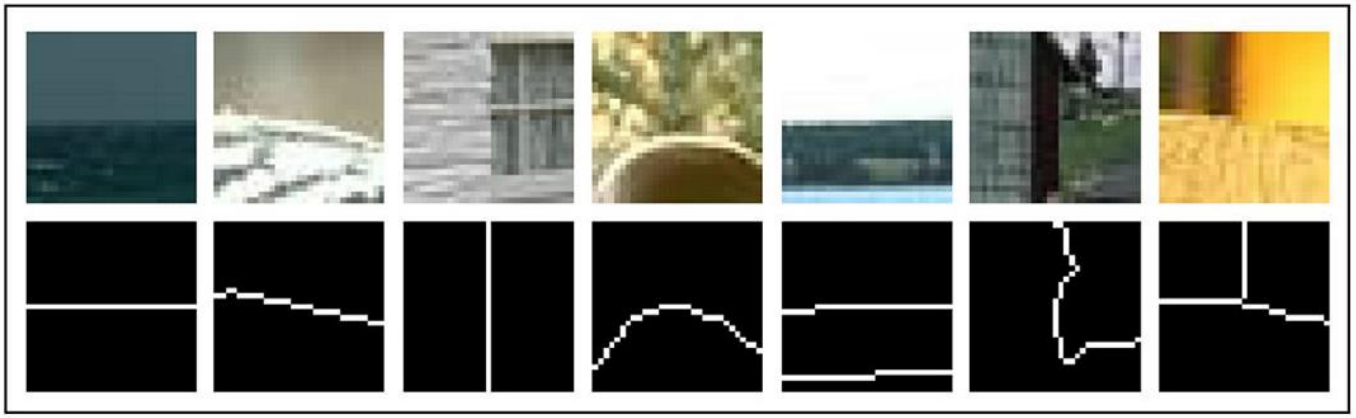


Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection

Joseph Lim, C Lawrence Zitnick, Piotr Dollar

CVPR 2013





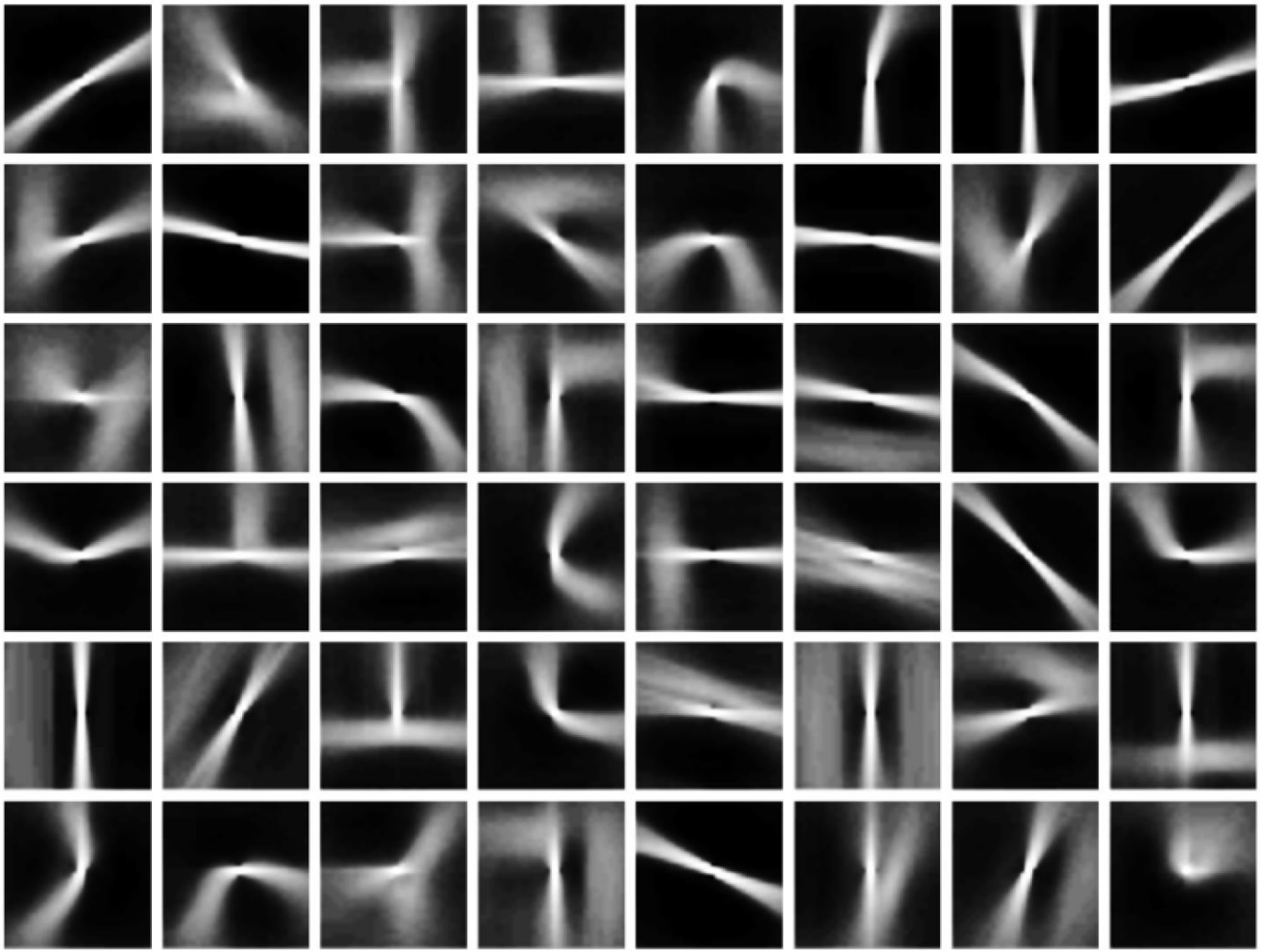
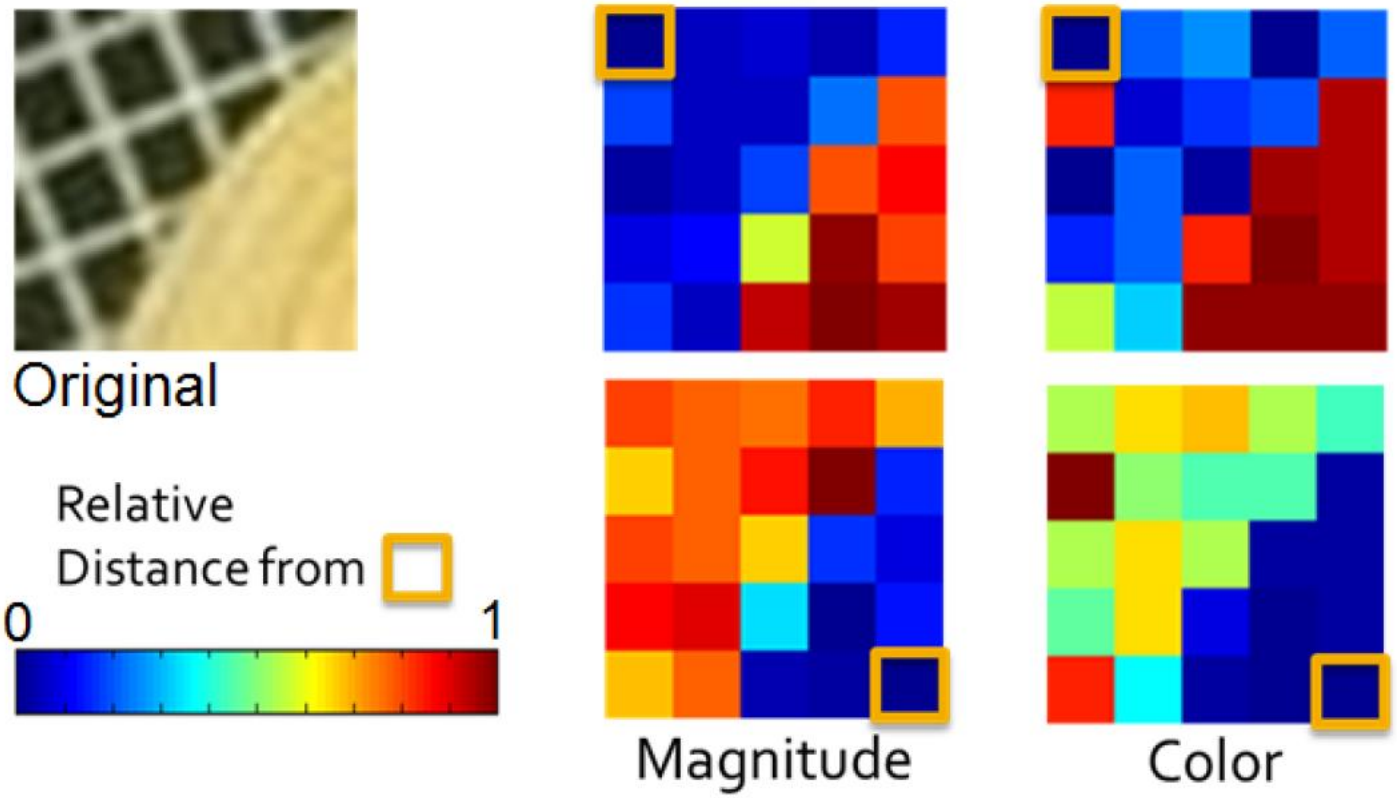


Image Features – 21350 dimensions!

- 35x35 patches centered at every pixel
- 35x35 “channels” of many types:
  - Color (3 channels)
  - Gradients (3 unoriented + 8 oriented channels)
    - Sigma = 0, T heta = 0, pi/2, pi, 3pi/2
    - Sigma = 1.5, Theta = 0, pi/2, pi, 3pi/2
    - Sigma = 5
  - Self Similarity
    - 5x5 maps of self similarity within the above channels for a particular anchor point.

Self-similarity features

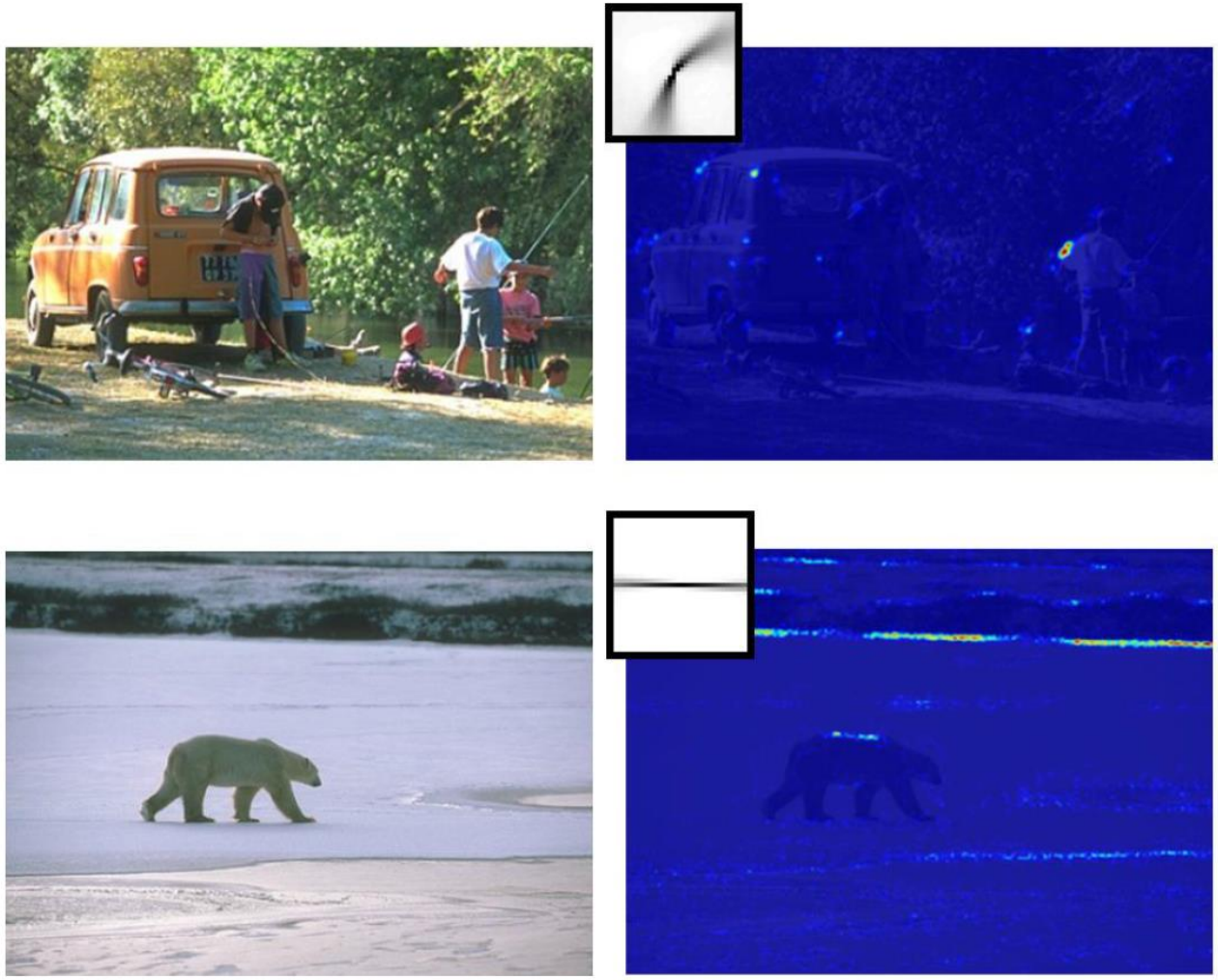


Self-similarity features: The L1 distance from the anchor cell (yellow box) to the other 5 x 5 cells are shown for color and gradient magnitude channels. The original patch is shown to the left.

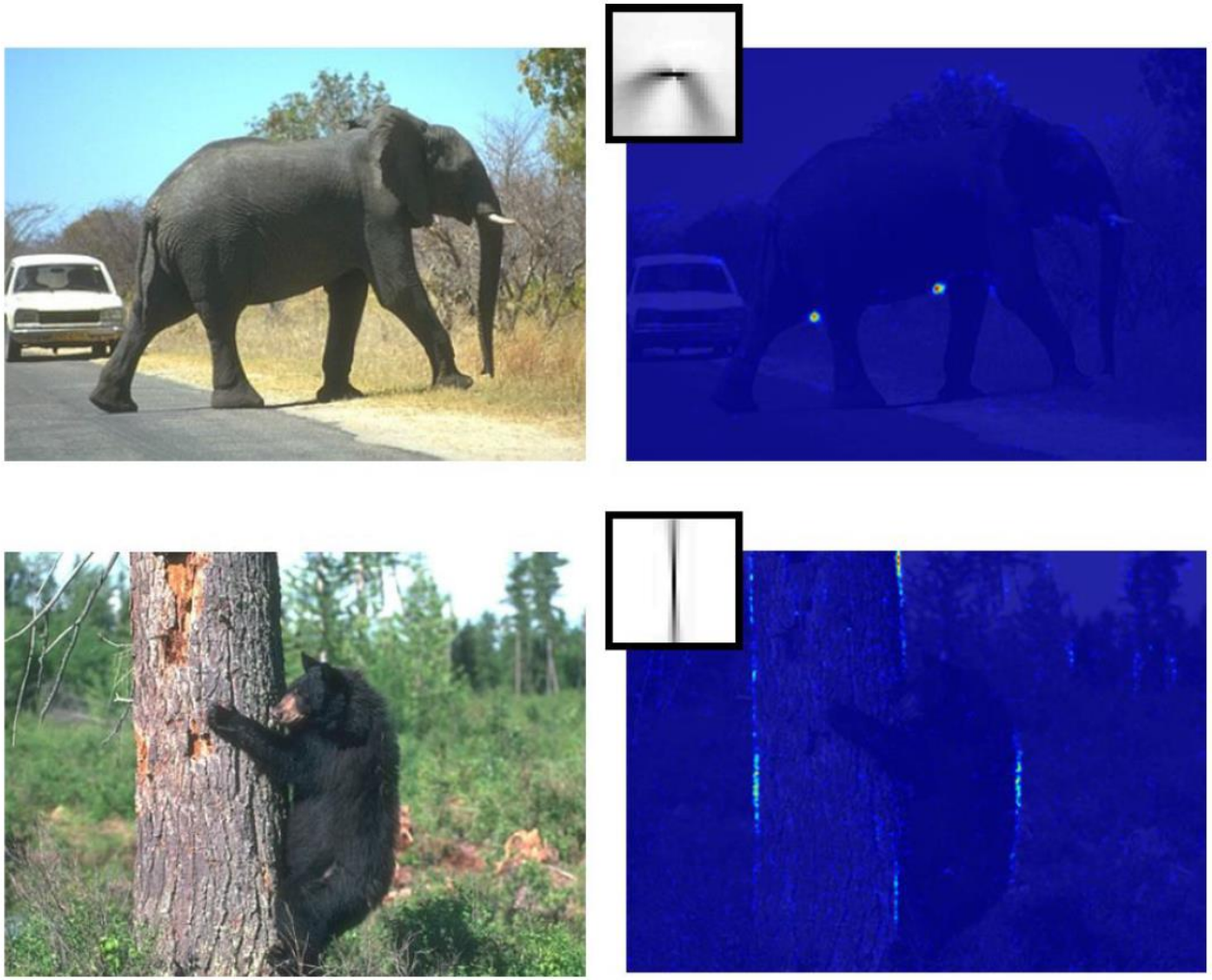
Learning

- Random Forest Classifiers, one for each sketch token + background, trained 1-vs-all
- Advantages:
  - Fast at test time, especially for a non-linear classifier.
  - Don’t have to explicitly compute independent descriptors for every patch. Just look up what the decision tree wants to know at each branch.

Detections of individual sketch tokens



Detections of individual sketch tokens

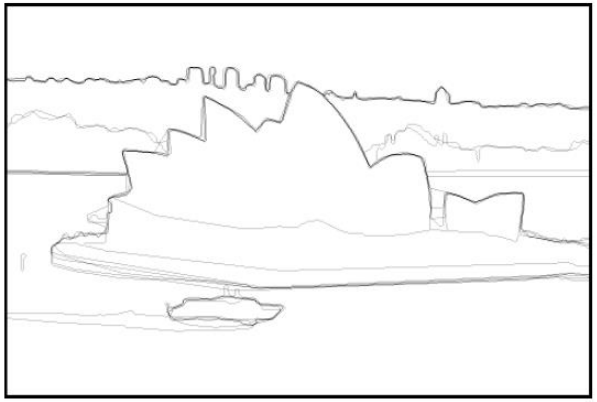
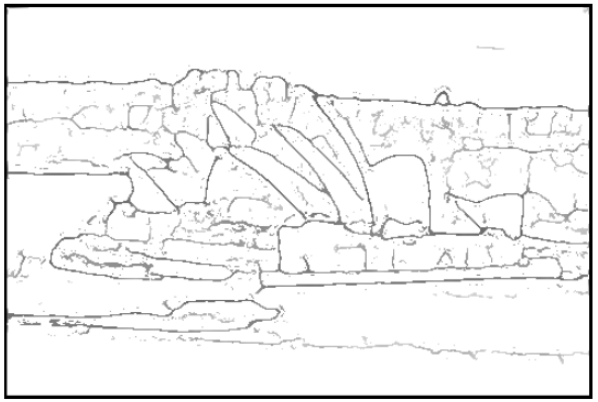


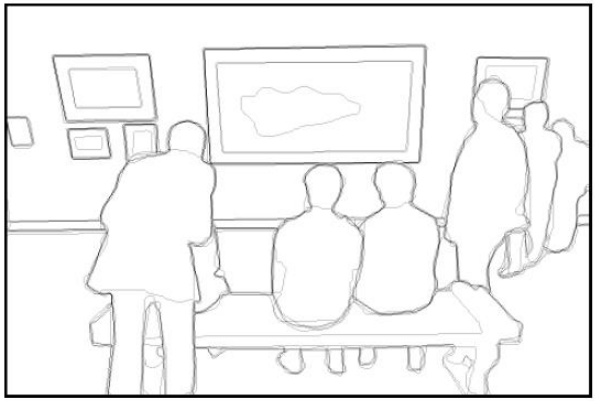
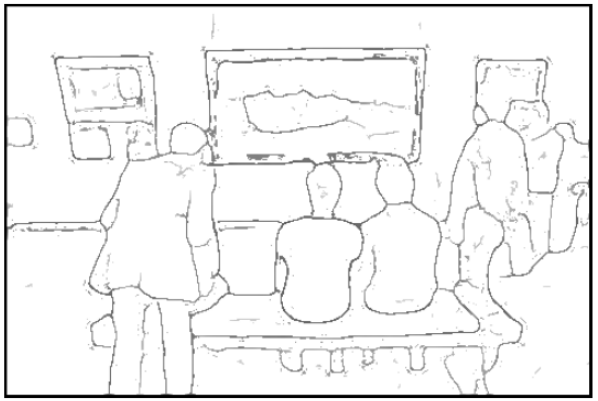
Combining sketch token detections

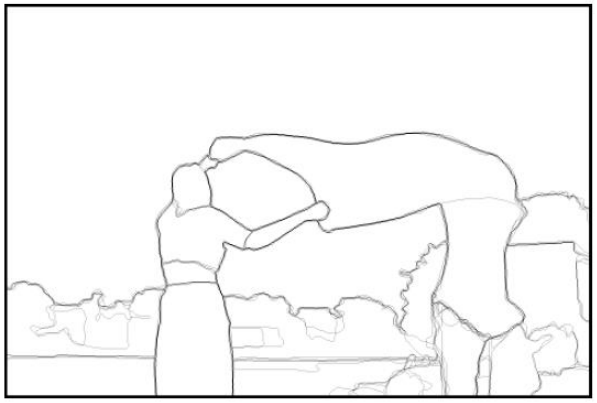
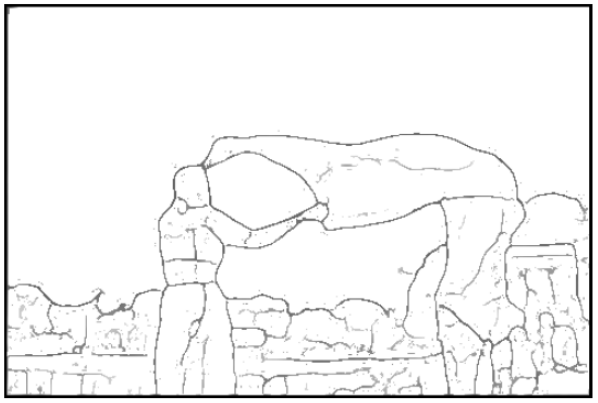
- Simply add the probability of all non-background sketch tokens

- Free parameter: number of sketch tokens
  - k = 1 works poorly*, k = 16 and above work OK.

*Actually, the entire sketch clustering idea might not be important.

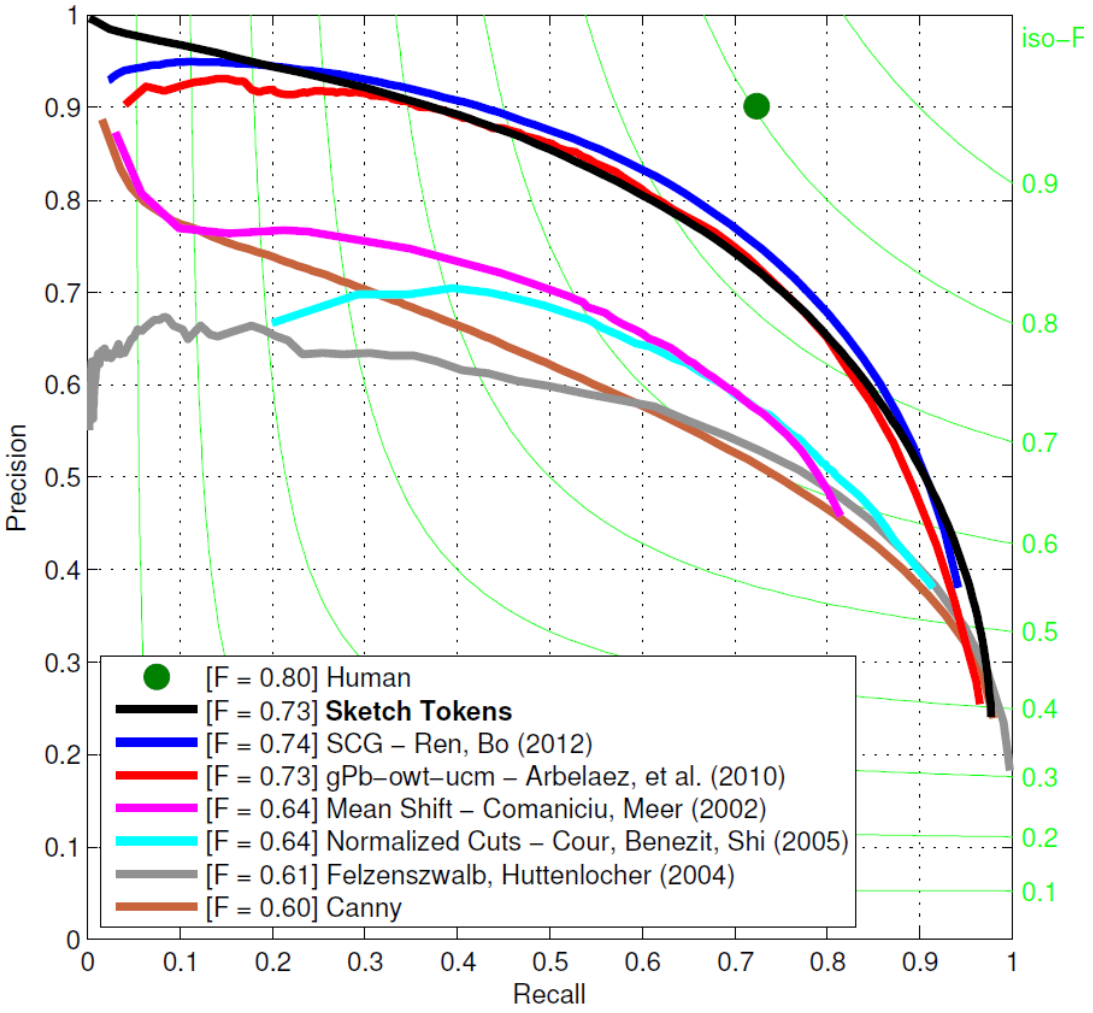



Input Image Ground Truth Sketch Tokens

Evaluation on BSDS

Method	ODS	OIS	AP	Speed
Human	.80	.80	-	-
Canny	.60	.64	.58	1/15 s
Felz-Hutt [12]	.61	.64	.56	1/10 s
gPb (local) [1]	.71	.74	.65	60 s
SCG (local) [24]	.72	.74	.75	100 s
Sketch tokens	**.73**	**.75**	**.78**	**1 s**
gPb (global) [1]	.73	.76	.73	240 s
SCG (global) [24]	.74	.76	.77	280 s

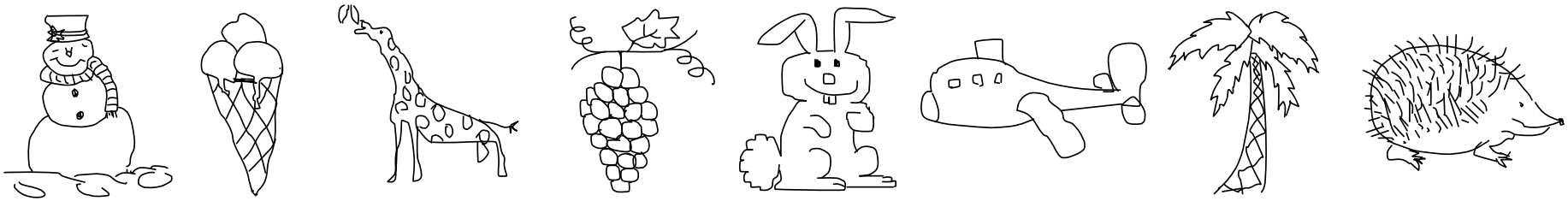
Evaluation on BSDS



Summary

- Distinct from previous work, cluster the *human annotations* to discover the mid-level structures that you want to detect.
- Train a classifier for every sketch token.
- Is as accurate as any other method while being 200 times faster and using no global information.

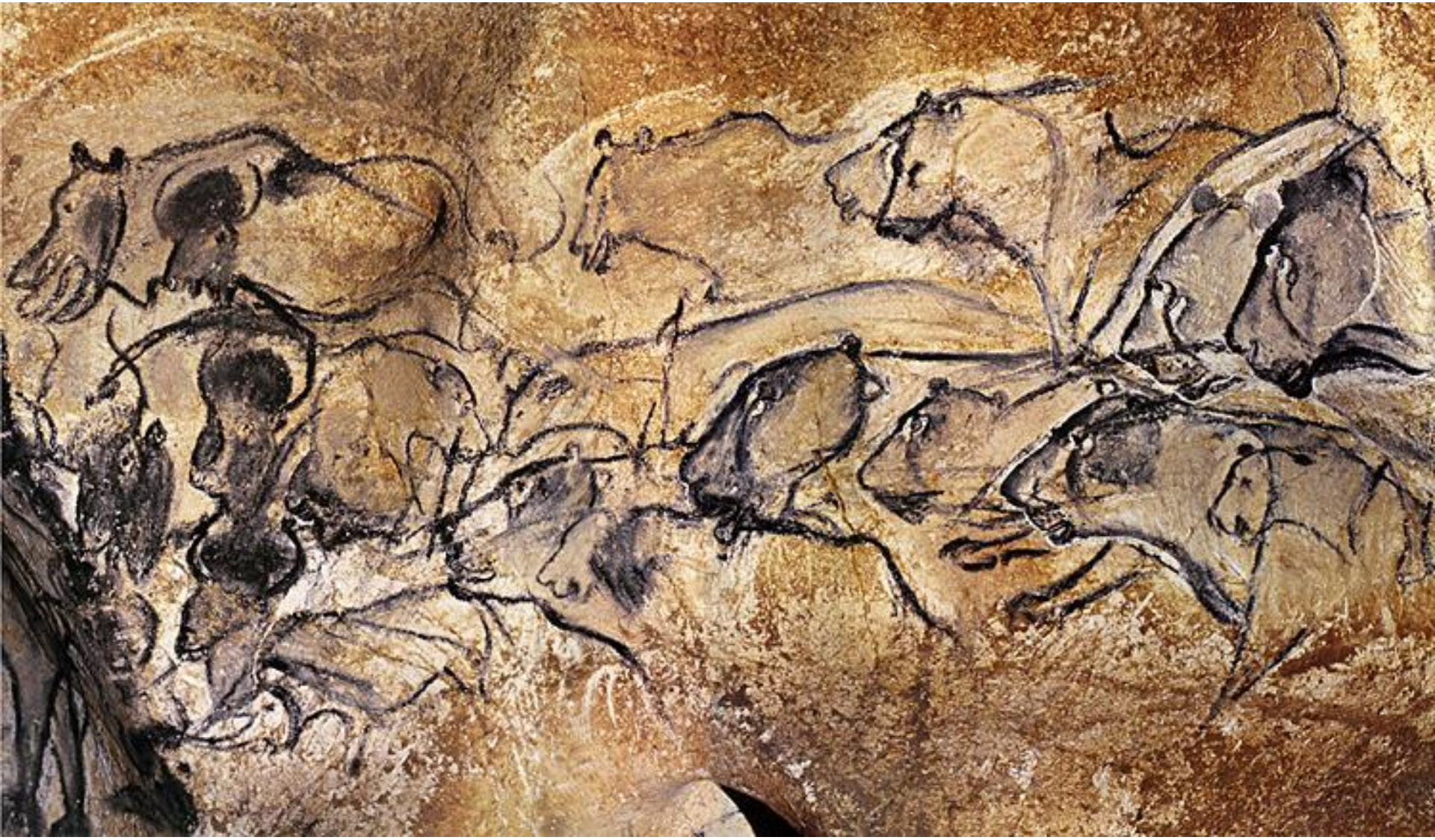
How Do Humans Sketch Objects?



Mathias Eitz, James Hays, and Marc Alexa. Siggraph 2012

Sketches Are Important

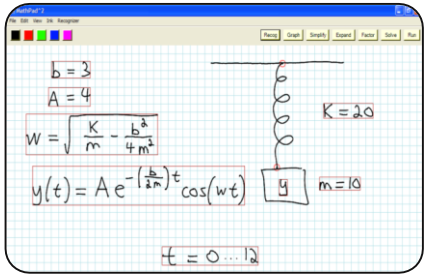
20,000 years ago (Lascaux, France)



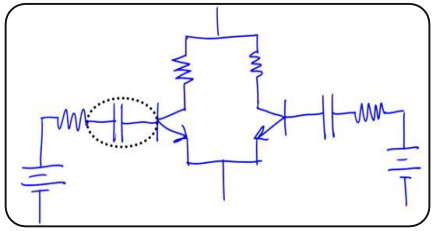
Sketches Are Important

Today

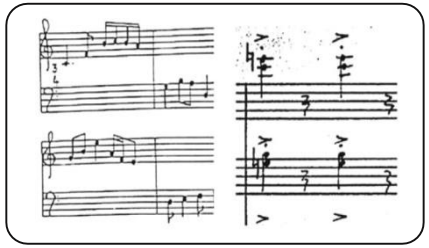
Prior Work: Domain-specific Recognition



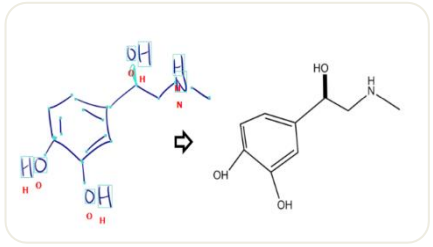
[LaViola and Zeleznik 2004]



[Sezgin and Davis 2008]

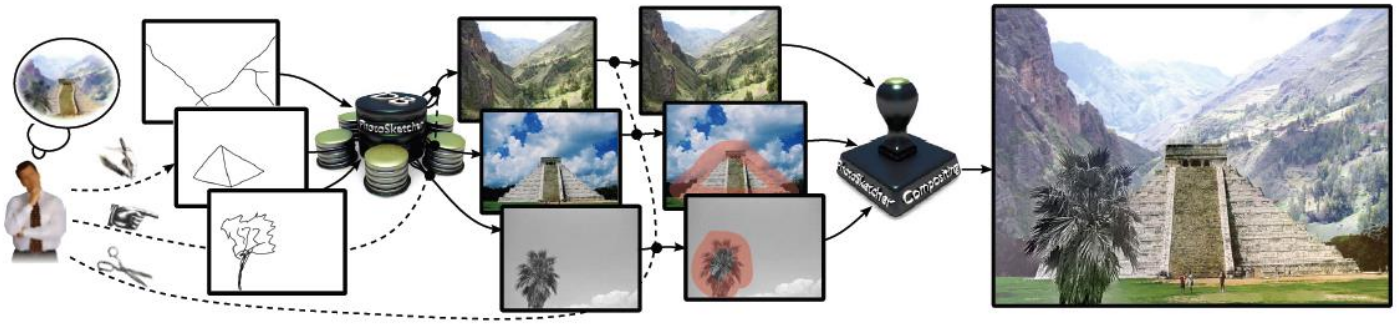


[Rebelo et al. 2009]



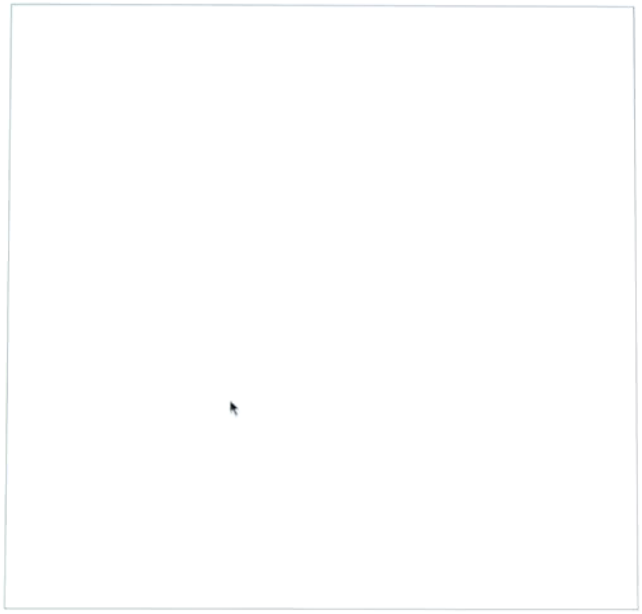
[Ouyang and Davis 2011]

Prior Work: Sketch-based image retrieval



- PhotoSketcher. Eitz, Richter, Hildebrand, Boubekeur, and Alexa. CGA 2011.
- Sketch2Photo. Chen, Cheng, Tan, Shamir, Hu. Siggraph Asia 2009

Our work

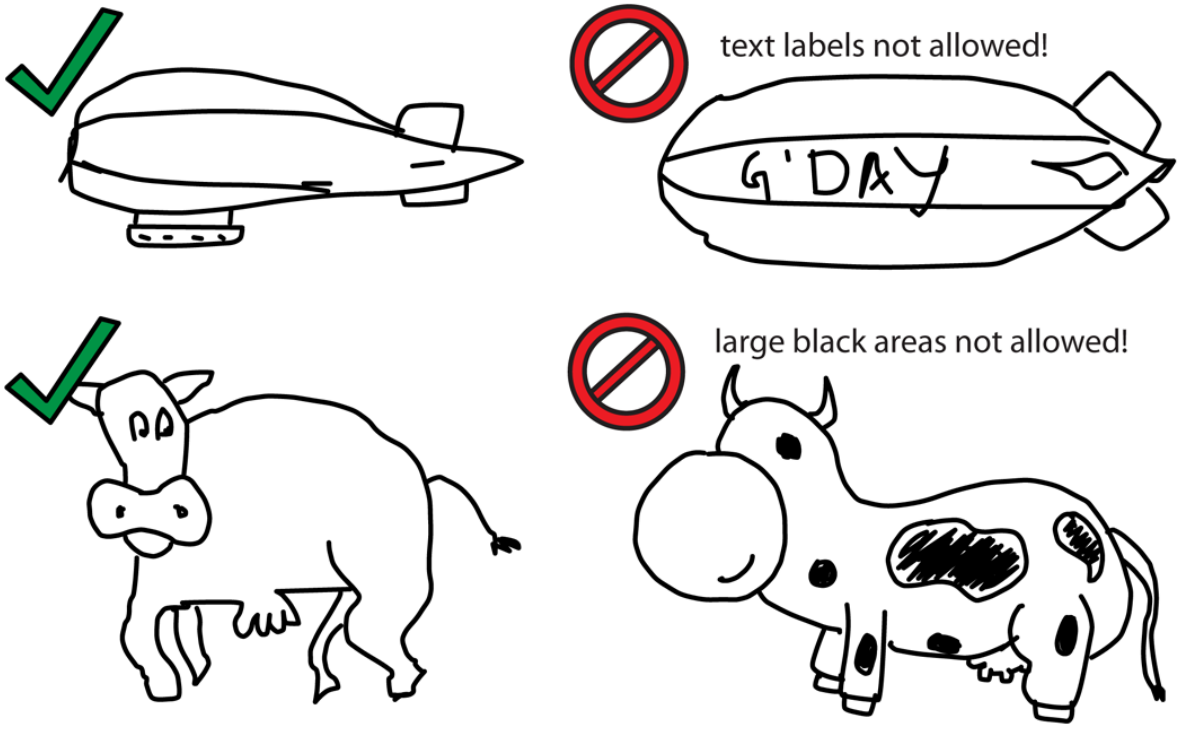


How Do Humans Sketch Objects?

- Need many example sketches from a variety of humans
- We used amazon Mechanical Turk

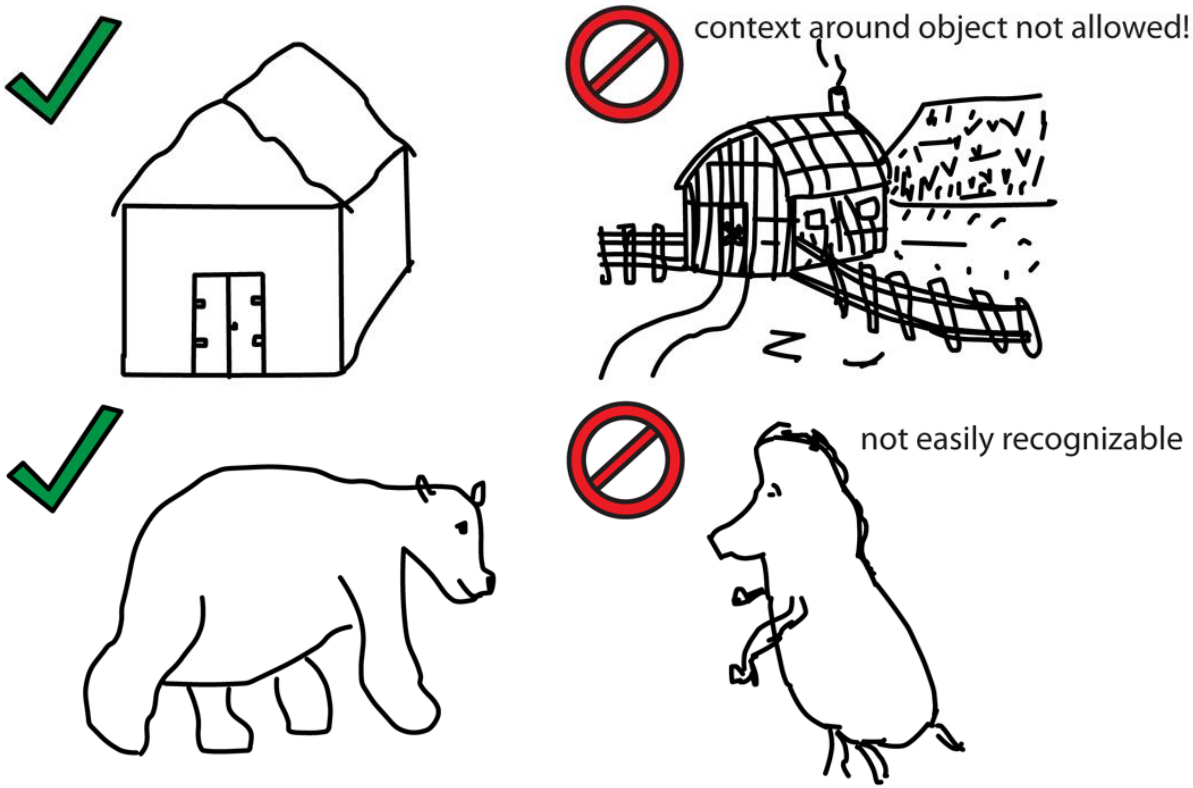
"Please sketch an image that is clearly recognizable to other humans as belonging to the following category: airplane"

How Do Humans Sketch Objects?



Mechanical Turk Instructions

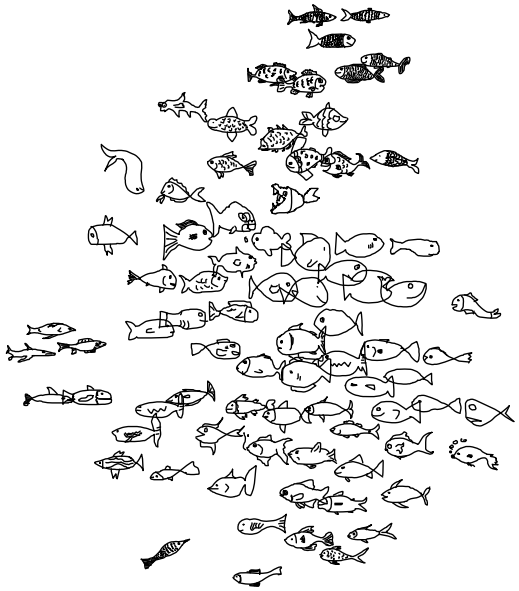
How Do Humans Sketch Objects?

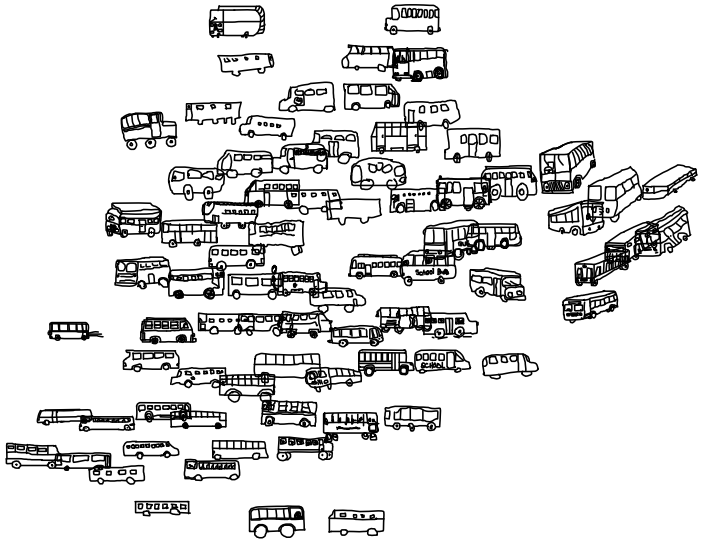


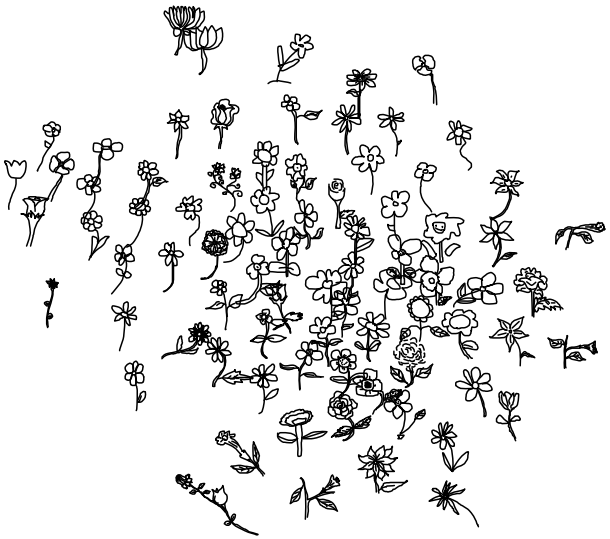
Mechanical Turk Instructions

How Do Humans Sketch Objects?

- 20,000 sketches in 250 categories
  - 1,350 unique participants, 741 hours drawing time

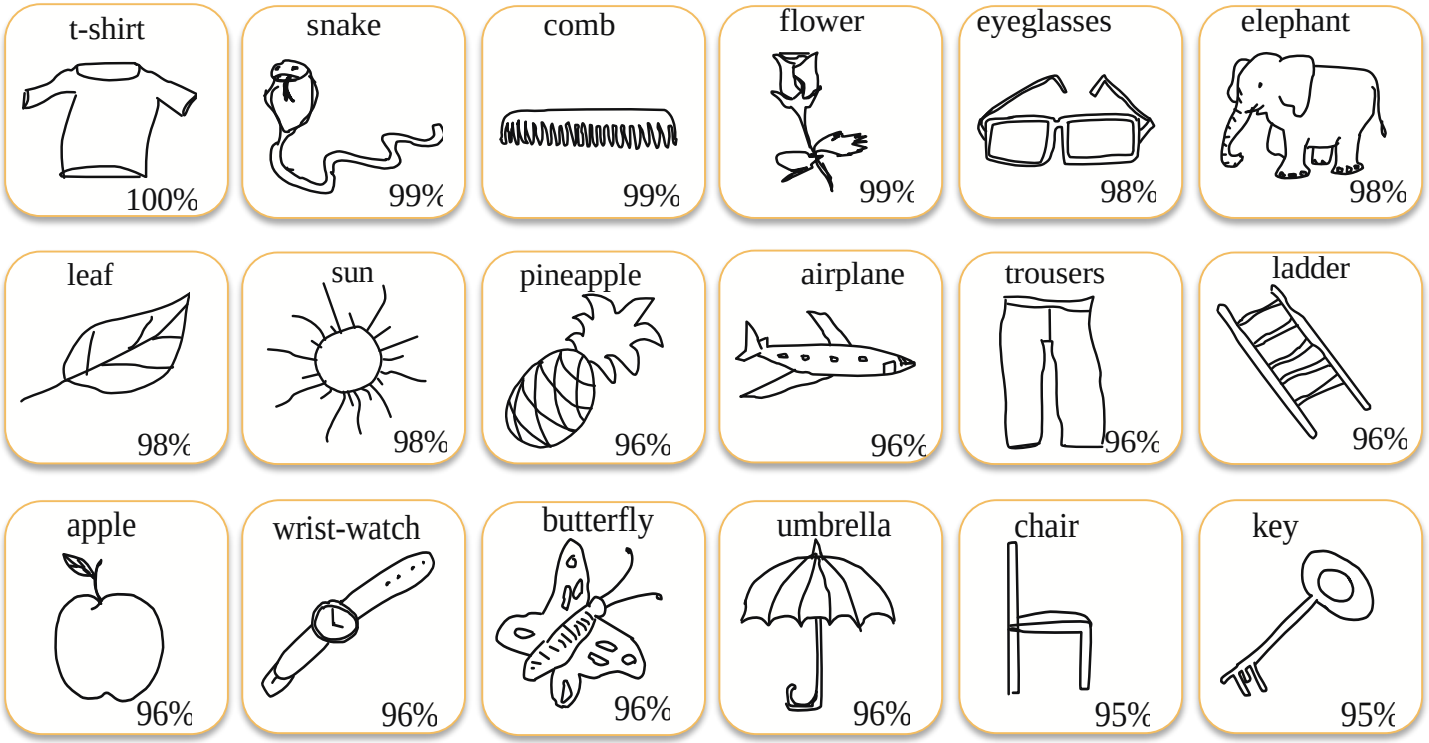




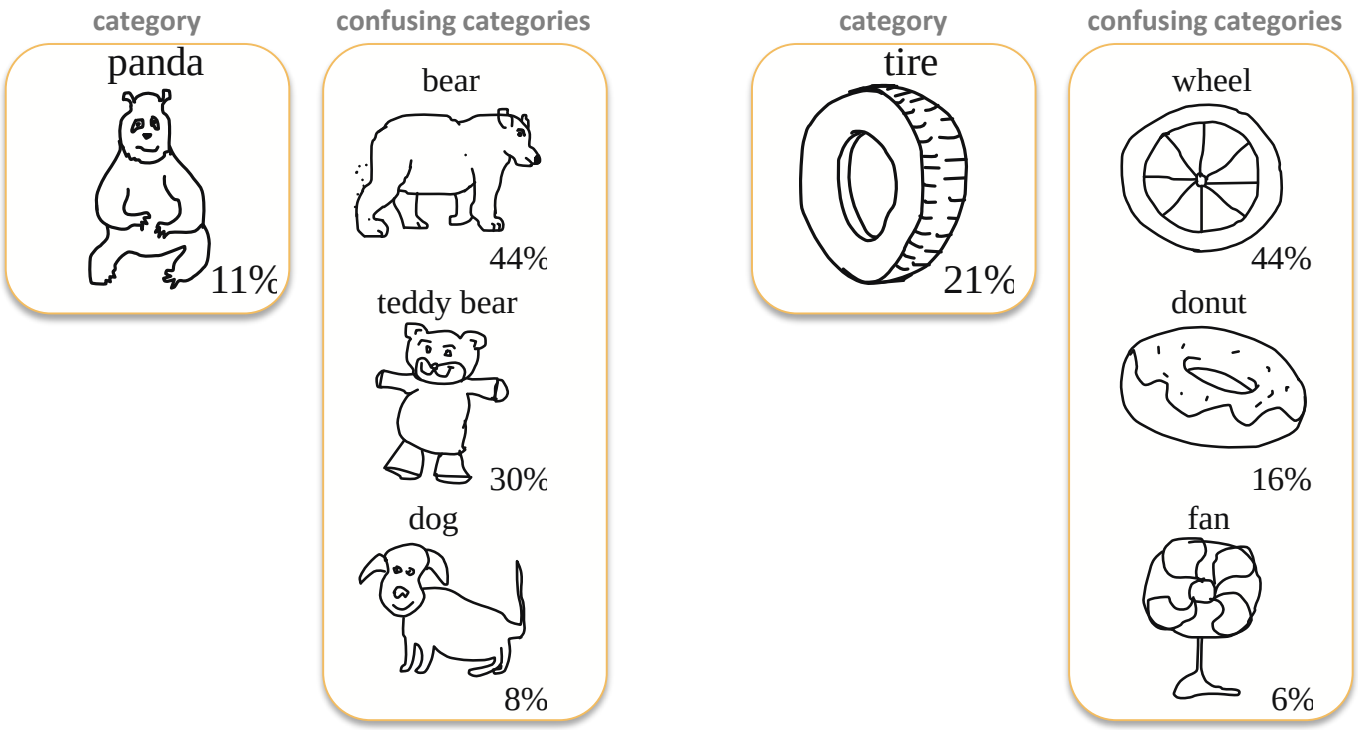


Human Sketch Recognition

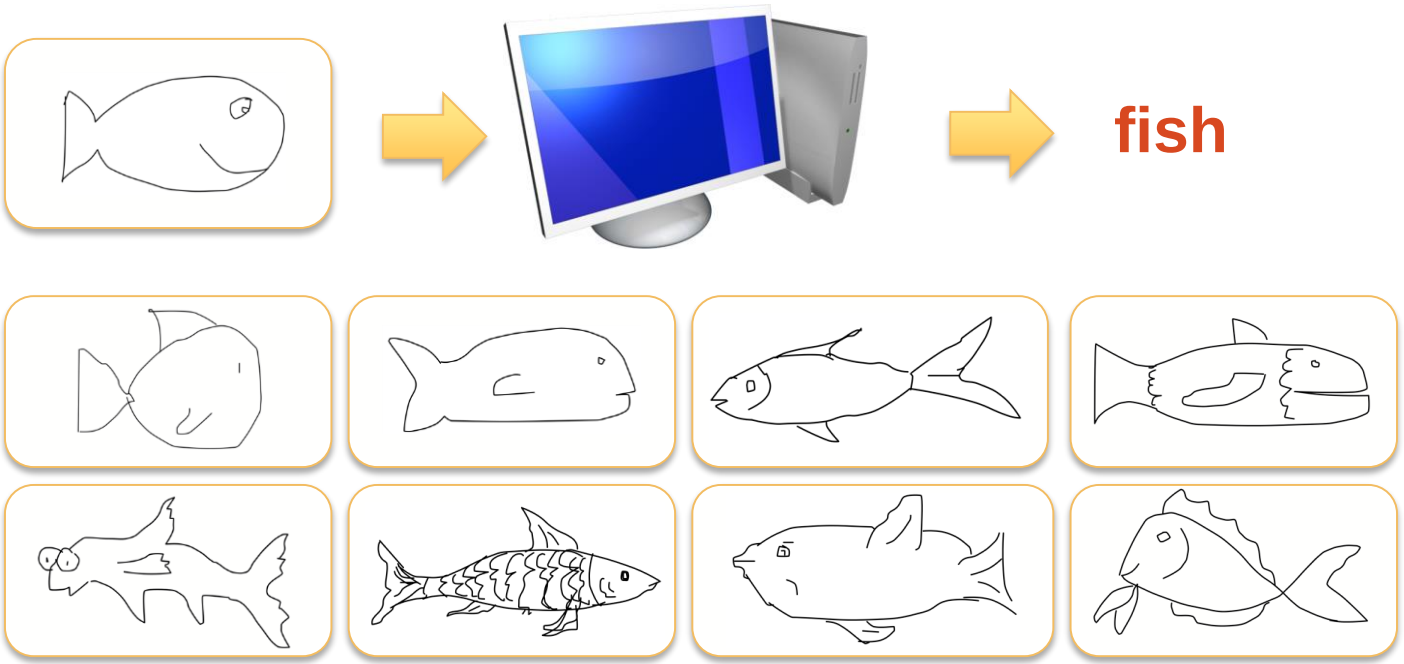
- 73% overall human recognition accuracy



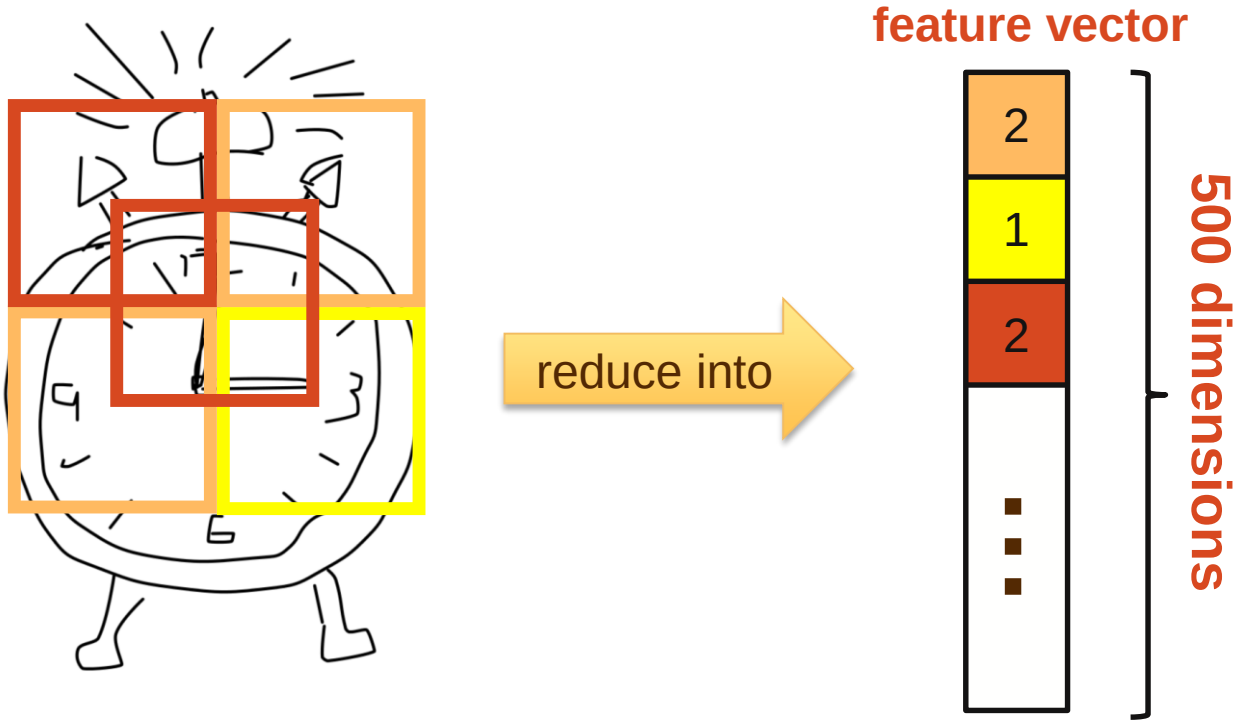
Human Sketch Recognition

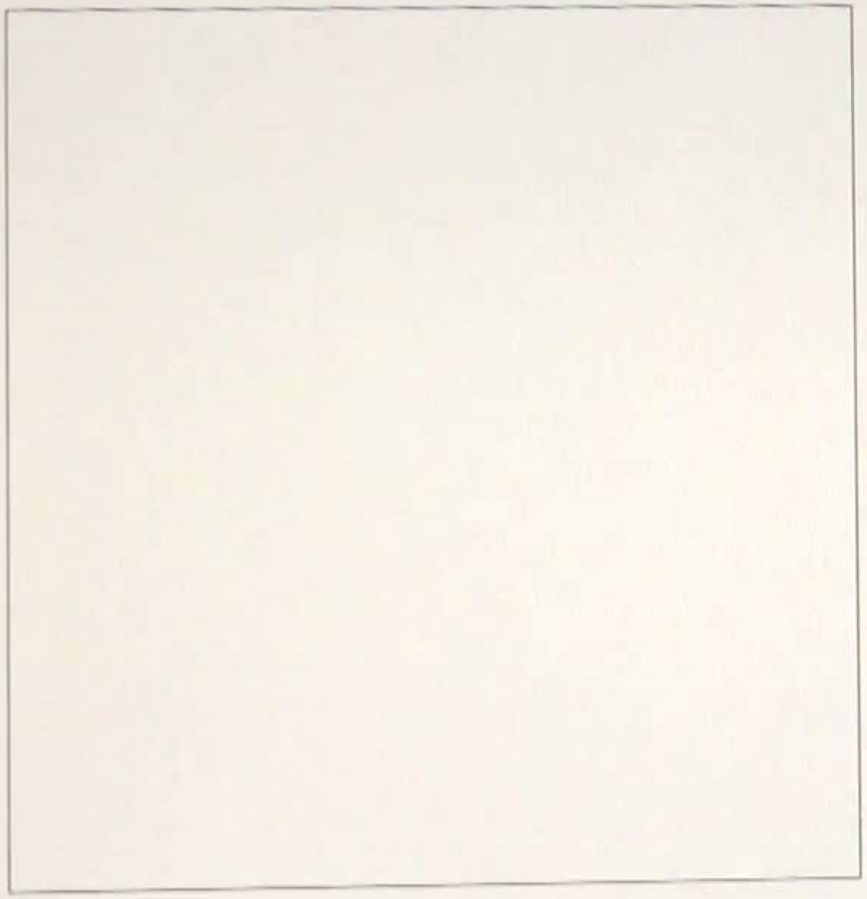


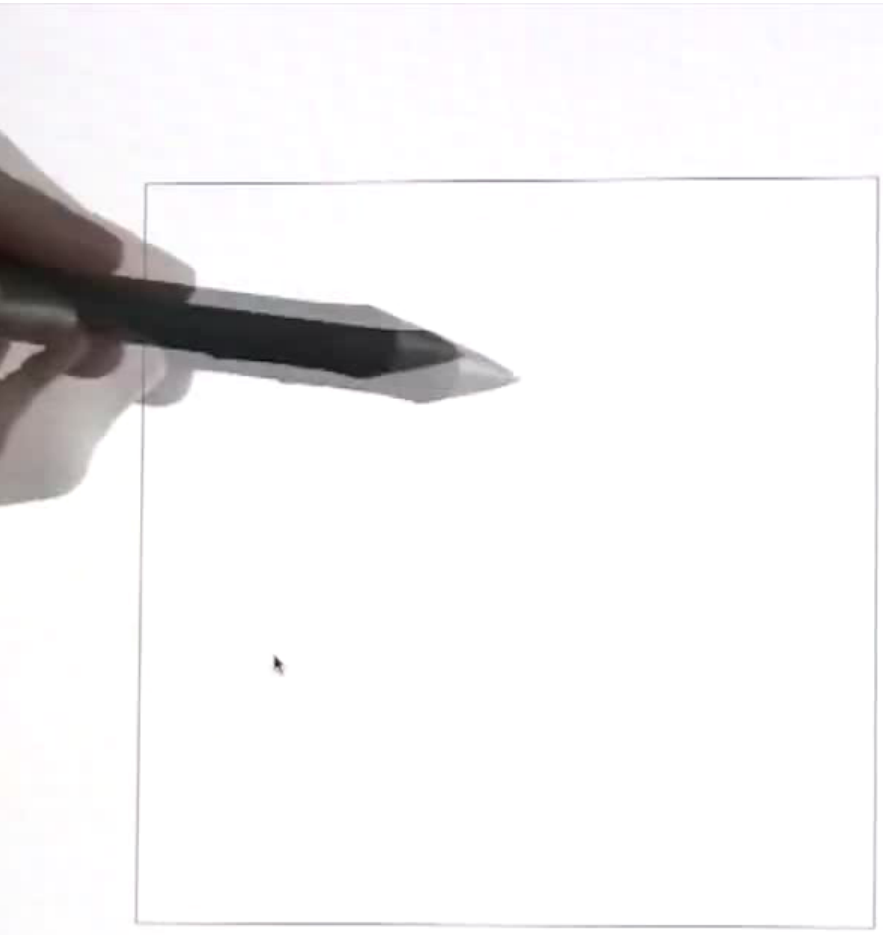
Sketch Recognition



Bag-of-Features Representation with SIFT-like features







Sketch Recognition

Does the system generalize beyond our AMT sketches?

flying bird

camel

sheep

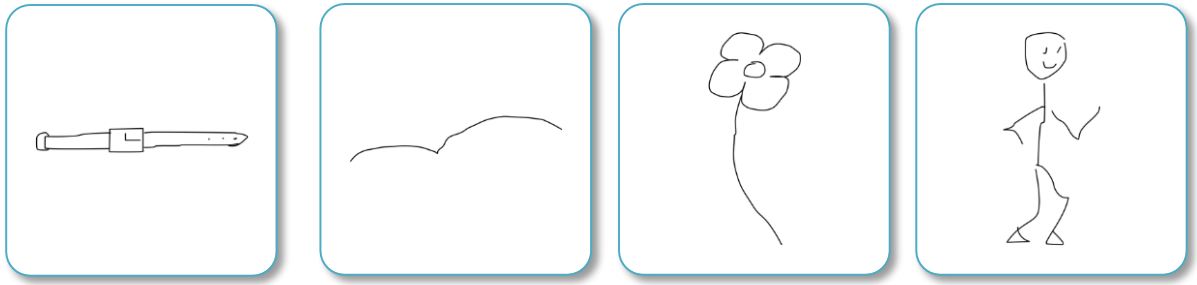
is: antilope

horse

Recognition Accuracy Over Time

- 44% bag of features + k nearest neighbors
- 56% bag of features + nonlinear SVM
- 65% deep learning features (no fine tuning)
- 65% Iterative-closest point (ICP) on nearest neighbors
- 68% bag of features + Fisher Vector encoding
  [Schneider and Tuytelaars. Siggraph Asia 2014]
- 73% humans
- 75% [Sketch-a-Net that Beats Humans. Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, Timothy Hospedales. BMVC 2015]
- ~75% [Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-Miller. ICCV 2015.]
- 80% fine-tuned GoogLeNet [Sketchy Database, Sangkloy et al. 2016]

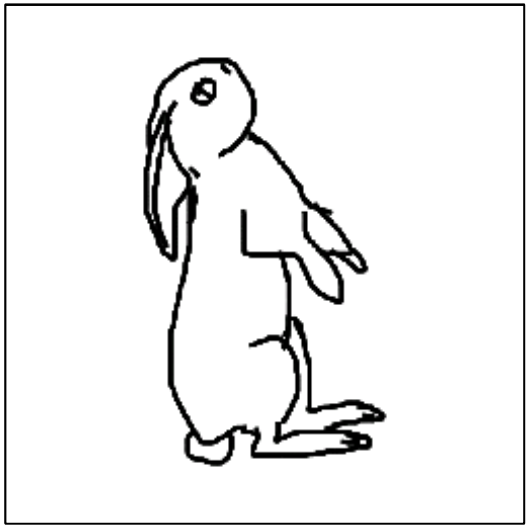
Conclusions



wristwatch flying bird flower person walking

People tend to agree on iconic representations which are often abstract and far from original geometry

Sketches are relatively easy to recognize with existing computer vision tools



Does recognizing sketches help us retrieve sketches?