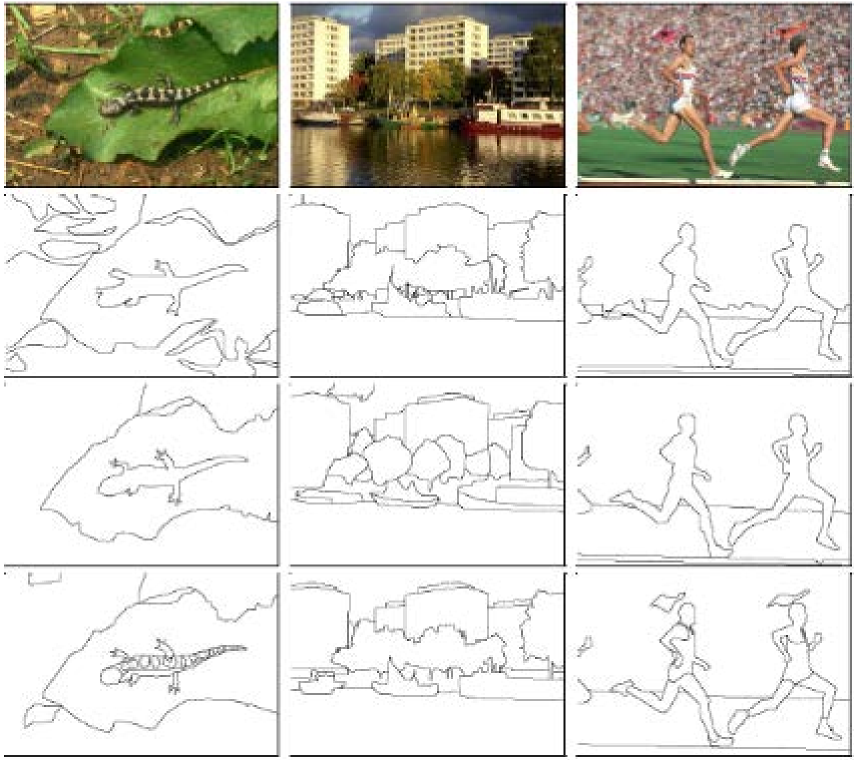
LEARNING-BASED EDGE AND CONTOUR DETECTION

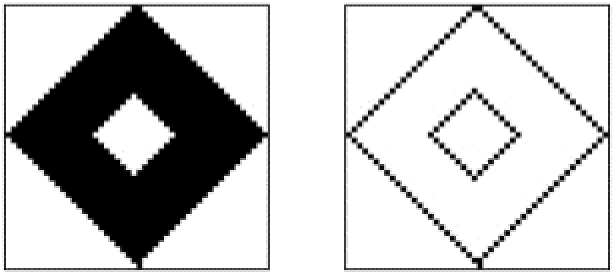
C.-C. Jay Kuo
University of Southern California

Problem Definition

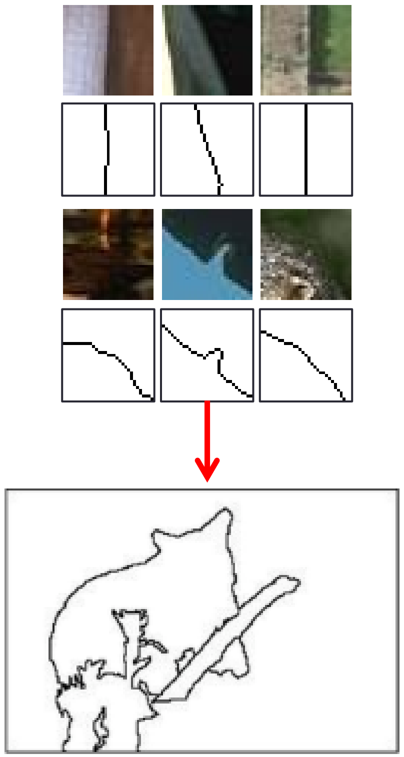
- Find those Visually Salient Contours to help image understanding
- Indicate the intersection of different meaningful regions



Edge v.s. Contour


(a) Input image (b) Edge detection

- Low level v.s. Mid level vision task
- Edge detection
  - Sharp changes in image brightness
  - Differential operation capture the discontinuities
  - Pixel-based

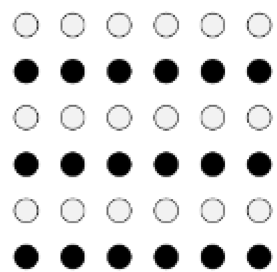


- Contour Detection
  - Contour/Boundary is generalized definition of edge
  - Synthesis ability of human vision system
  - Patch-based

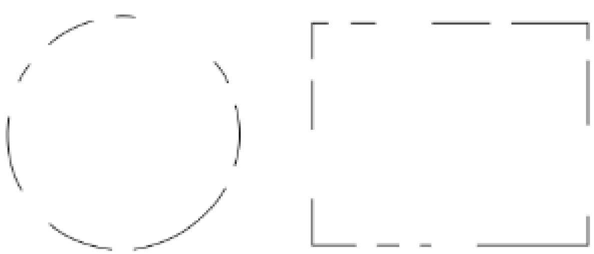
Motivation

- Human Vision System: Gestalt Laws
  - ➢Human is prone to group low-level image components

Proximity



Similarity

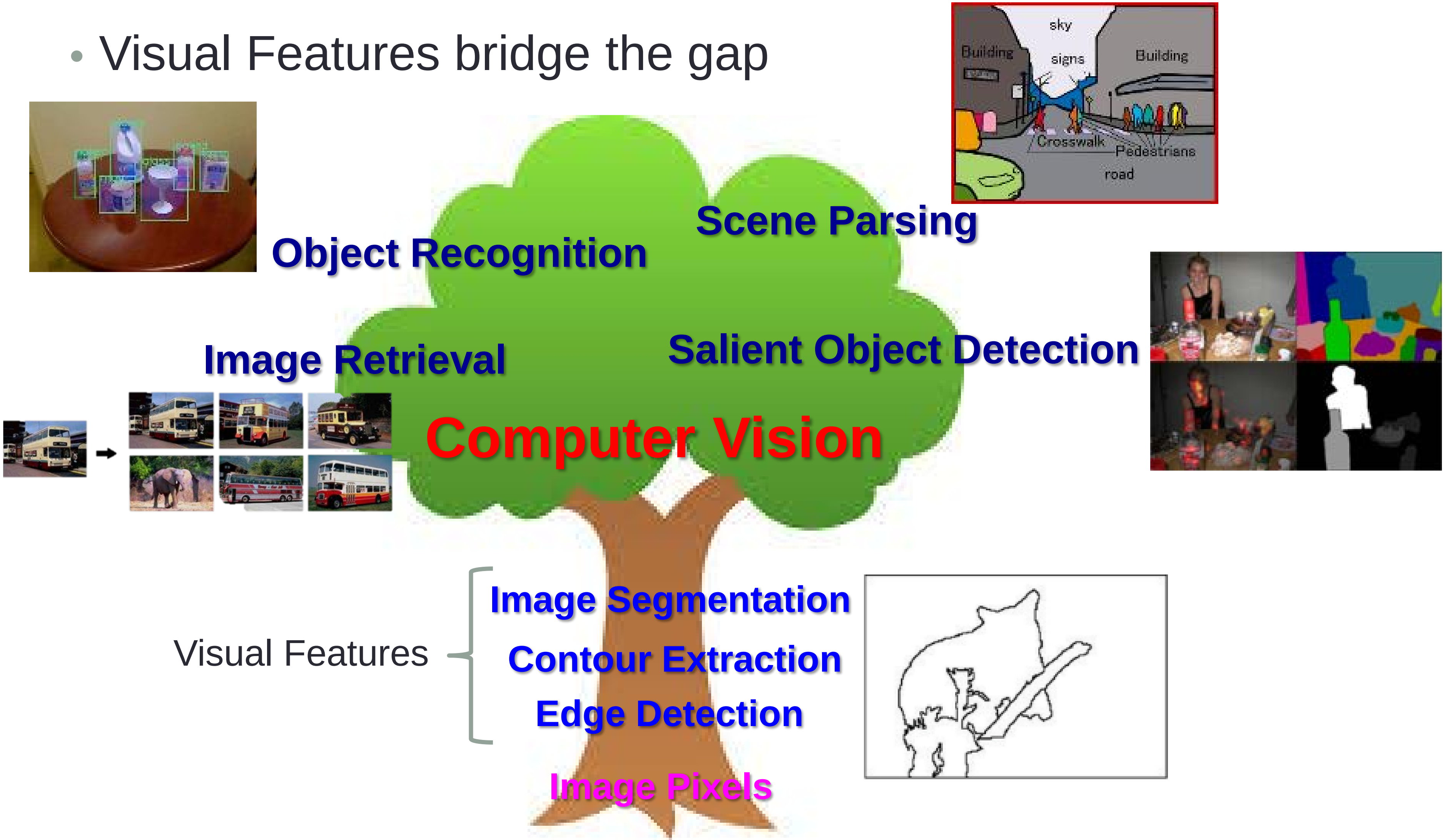


Closure

- Primitive features such as **edges, contours, corners, and regions** are much related to human visual perception
- Important role for **image interpretation** in computer

Motivation

- Visual Features bridge the gap

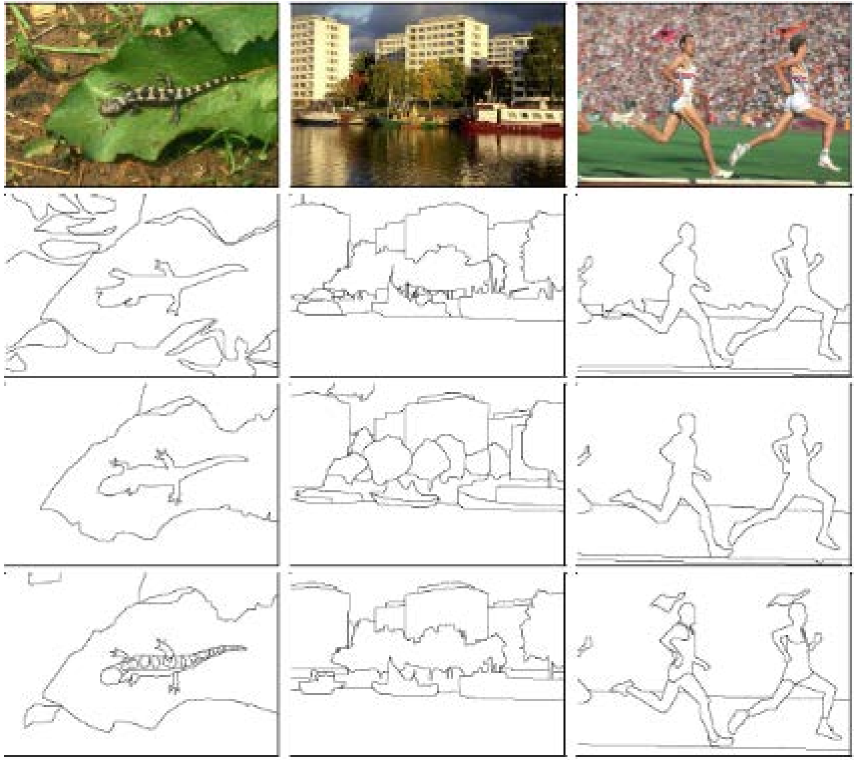


How to evaluate?

- Hard to define an edge/contour…

Problem Definition

- Find those Visually Salient Contours ? to help image understanding
- Indicate the intersection of different meaningful regions ?



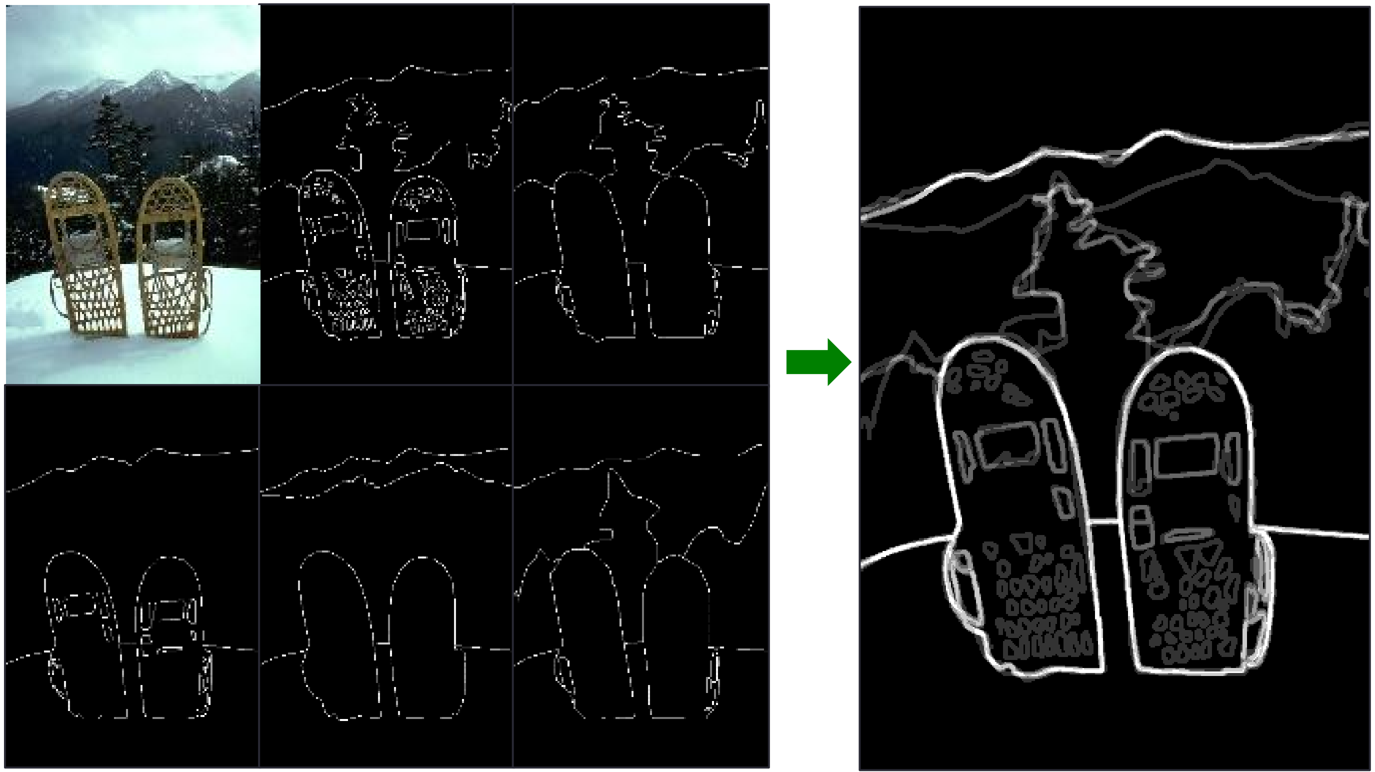
How to evaluate?

- Hard to define an edge/contour…
  - Peak gradient magnitude?
  - Discontinuity between color/texture?
  - Boundary of an object?
  - Even the edge sketched by human are different from person to person?

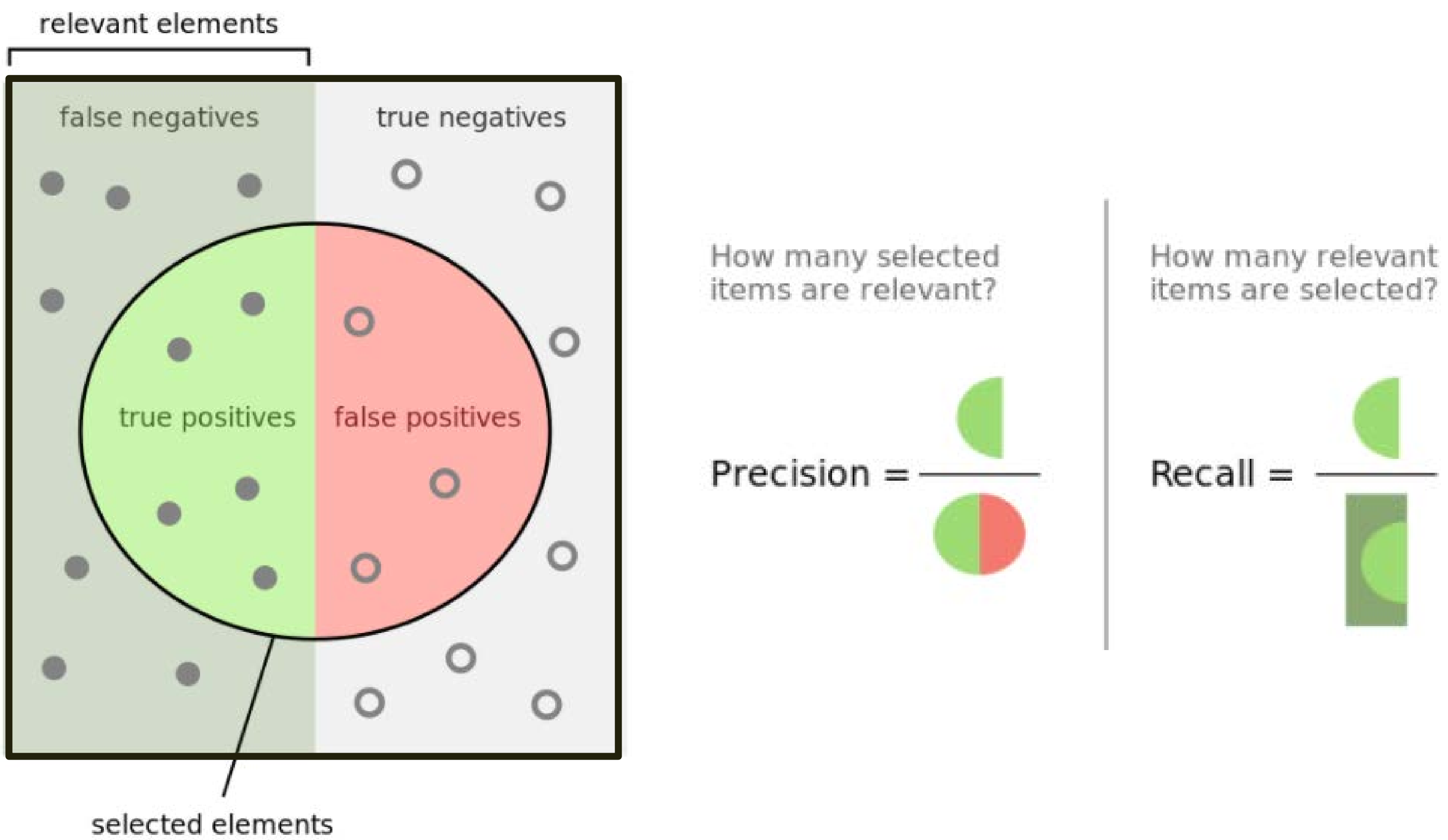
- For the time being, we utilize the segmentation ground truth for our goal
  - Not aim to match the taste of these subjective results
  - Need to be generalized further
  - Dataset Bias

Ground Truth Dataset

- Berkeley Segmentation Dataset 500
  - 500 images (200 train + 100 validation + 200 test)
  - Each image was annotated by five subjects on average
  - Served as evaluation of both "Segmentation" and "Contour"



Visualization of Precision and Recall



More about Evaluation Metric (1)

Evaluation Method

Precision: $\dfrac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$

Recall = $\dfrac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$

More about Evaluation Metric (2)

How to balance the two?

$$\frac{1}{2}\left(\frac{1}{P}\downarrow + \frac{1}{R}\downarrow\right) = \frac{1}{F}\downarrow$$

$$\frac{1}{2}\,\frac{P+R}{PR} = \frac{1}{F}$$

$$F = \frac{2PR}{P+R}$$

$P\uparrow, R\uparrow\ F\uparrow$

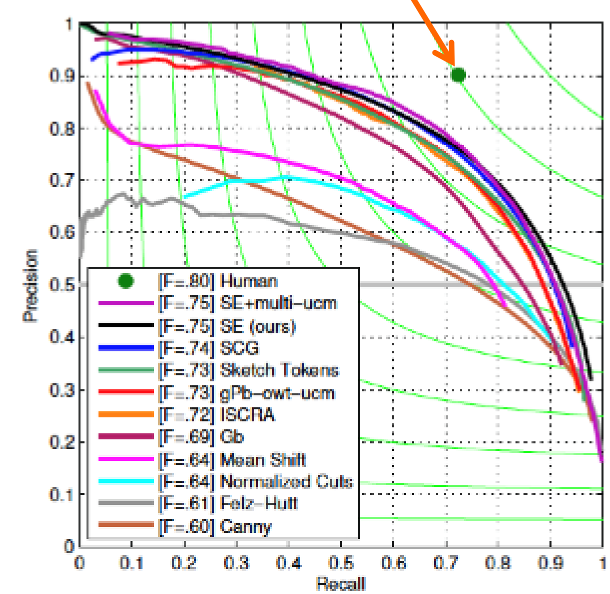
Evaluation Metric

- F-measure (Precision-Recall)
  - Precision(P) = True Positive / (True Positive + False Positive)
  - Recall(R) = True Positive / (True Positive + False Negative)

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

  - Average Precision (AP): Area under the F-curve

Max Score = 0.8



- Optimal Dataset Scale (ODS)
  - Choose optimal threshold for the test set
- Optimal Image Scale (OIS)
  - Choose optimal threshold for each image

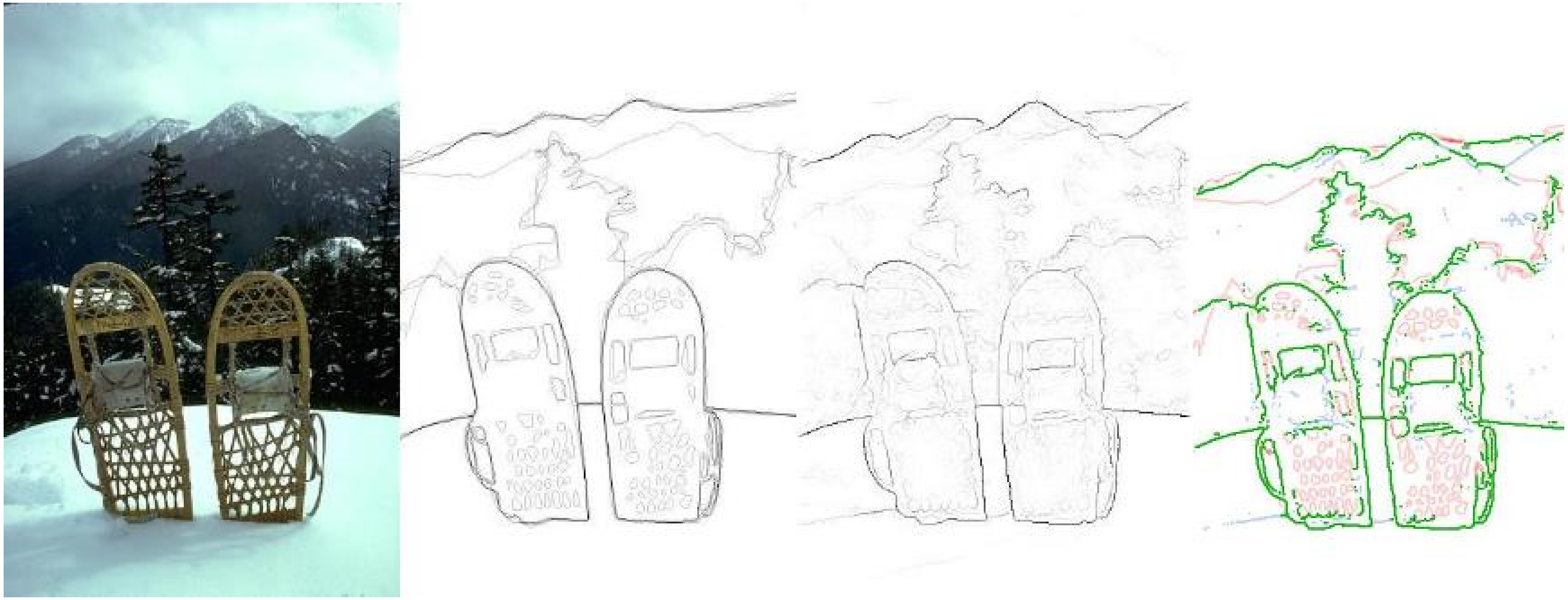
Evaluation Metric

- Error in visualization
  - True Positive: Green
  - False Positive: Blue
  - False Negative: Red

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Precision = G/(G+B)

Recall = G/(G+R)

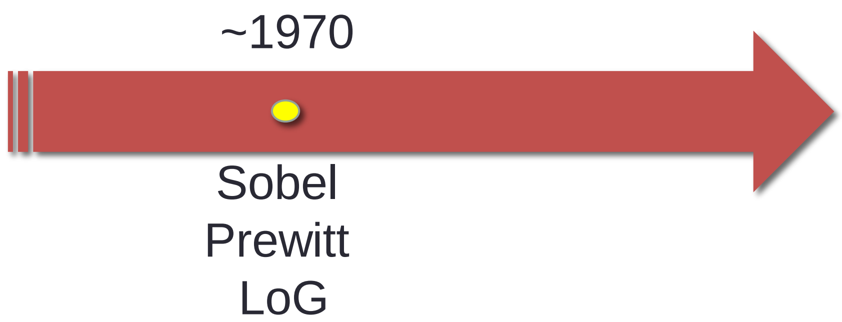


Input Ground Truth Contour Output Evaluation

Classic Methods

Prior Research Work

- Differentiation Based (HVS)



- Machine Learning Based (CVS)

Traditional

- Very local information about "**Edge**"
- Focus on **brightness discontinuities**
- **Differential operation** capture the strength and position

- (a) Prewitt, (b) Sobel, (c) Laplacian of Gaussian
  - ➢Local Maxima of gradient magnitudes are recorded as edges

a

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

b

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

c

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Traditional

- Results about “Edge” Detection

Input

Sobel

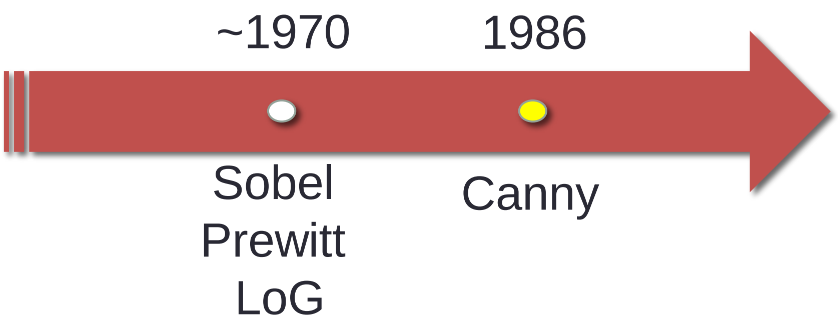
Prewitt

LoG

- Problems:
  - ➢Sensitive to noise
  - ➢Weak Localization
  - ➢Pixel-wise detection

Prior Research Work

- Differentiation Based (HVS)



- Machine Learning Based (CVS)

Canny Edge Detector

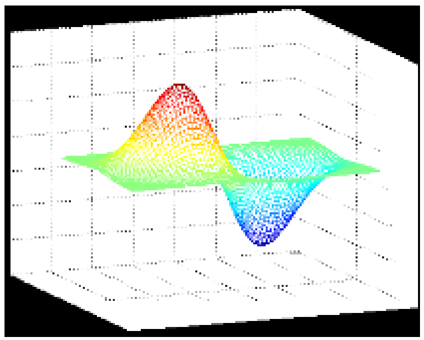
Canny Edge Detector

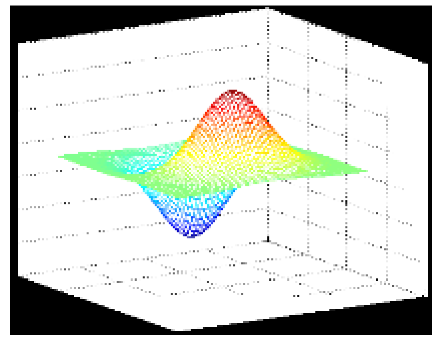
- Utilize Post-processing to refine edge maps
  - Consider the connectivity of “contour”

- Three Main Steps
  - Convolution with derivative of Gaussian
  - Non-maximum Suppression
  - Hysteresis Thresholding

Canny Edge Detector

- Convolution with derivative of Gaussian





$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

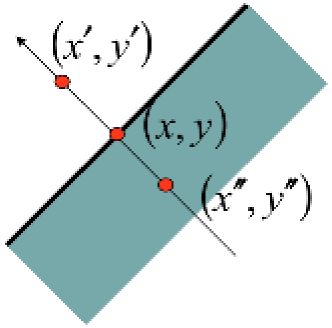
S_x

S_y

Canny Edge Detector

- Non-maximum Suppression (nms)

➢ Suppress the pixels in 'Gradient Magnitude Image' which are not local maximum

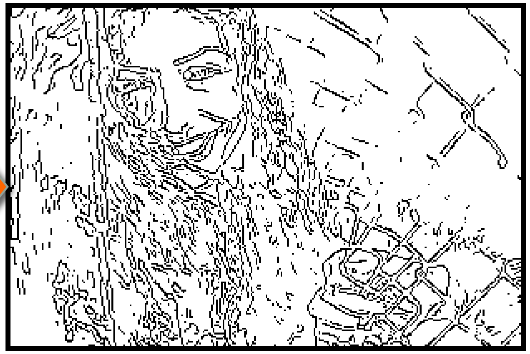


$$M(x,y)=\begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\nabla S|(x',y') \\ & \& \, |\nabla S|(x,y) > |\nabla S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$

$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

After Suppression

M



$$M \geq Threshold = 25$$

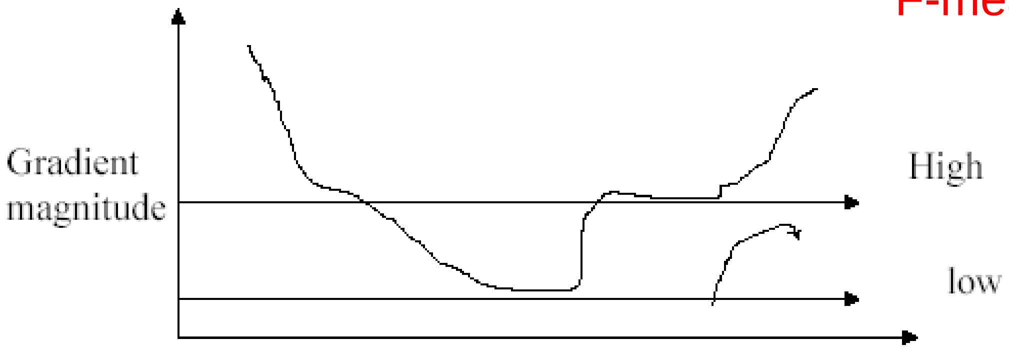
Canny Edge Detector

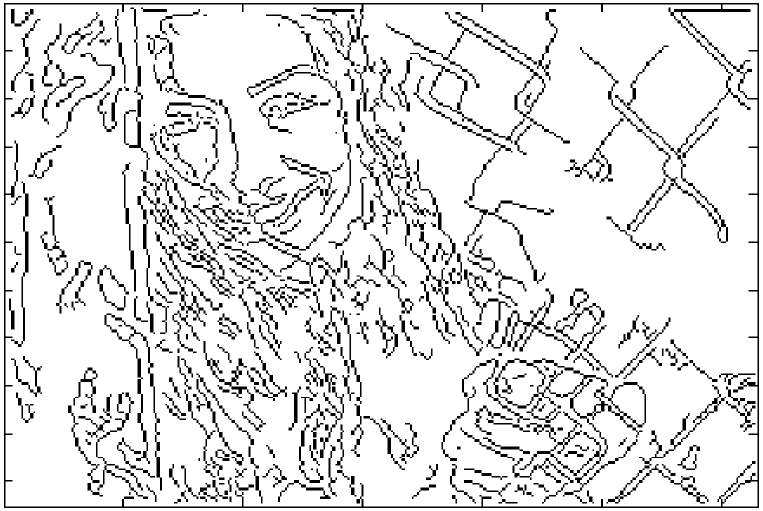
- Hysteresis Thresholding
  - Choose two thresholds: “high” and “low”
  - Above “high”: Edge
  - Below “low”: Non-Edge
  - Between “high” and “low”: Whether it connect to “Edge” or not

F-measure: 0.6

$High = 35$

$Low = 15$



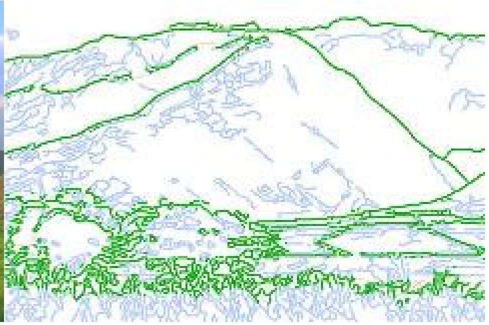


Connectivity of Contour

Canny Edge Detector

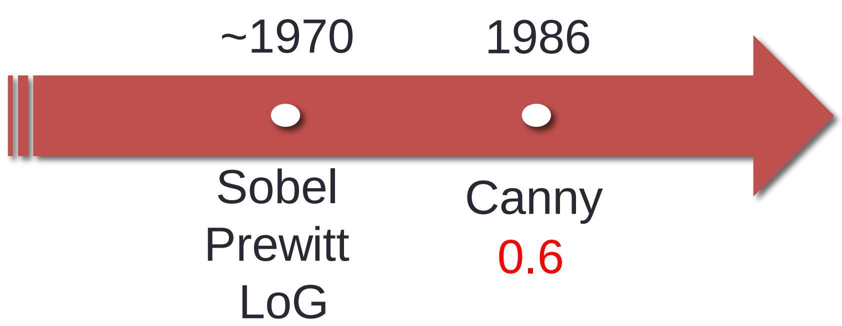
- Weakness
  - ➢ Sensitive to textured regions
    - Ambiguity for understanding
  - ➢ Not enough for image interpretation
    - Gradient on luminance only

- Mostly used as a pre-processing step
  - ➢ Low-level cues still play an important role
  - ➢ When low precision -> High Recall



Prior Research Work

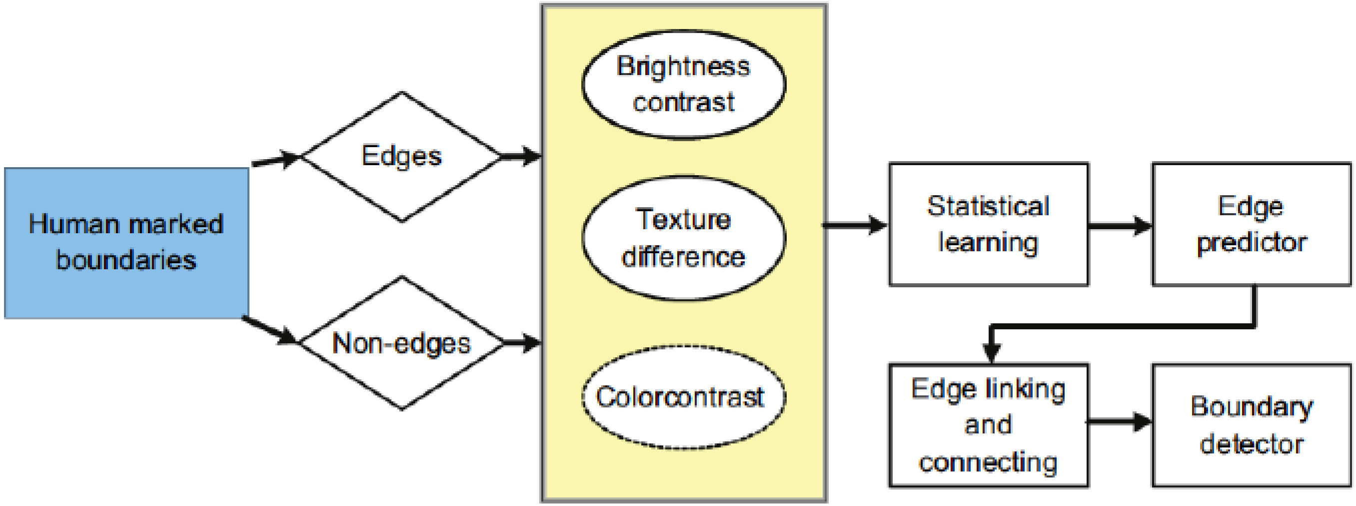
- Differentiation Based (HVS)



- Machine Learning Based (CVS)

Learning Based

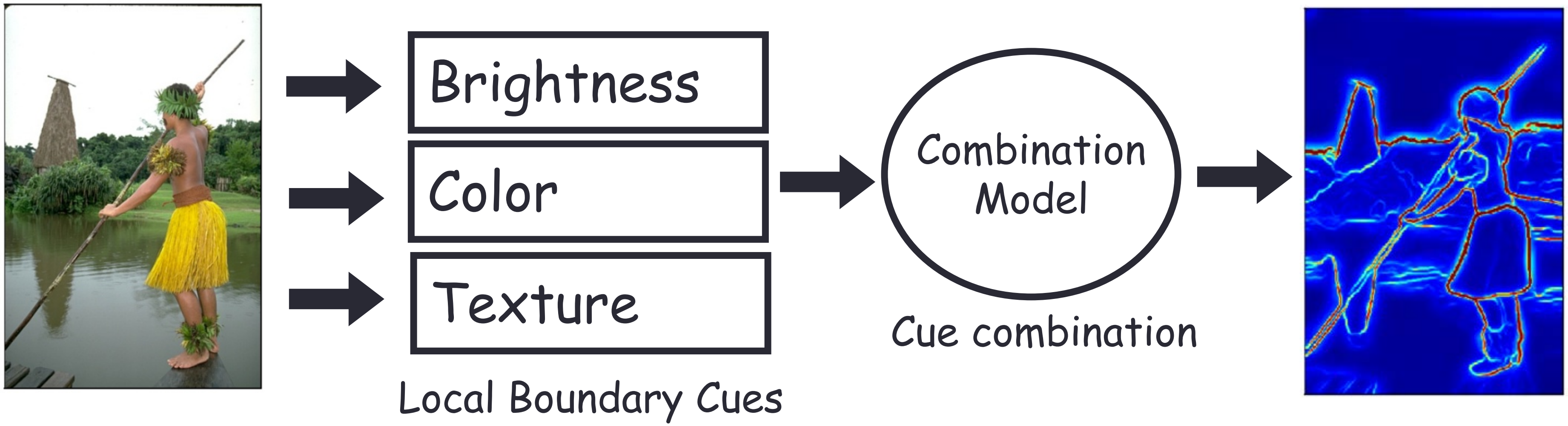
- Ground Truth driven approach **(Discriminative Model)**
  - Labels are created from human-generated sketch
  - Aim to mimic human perception with shortcut
- Features
  - Extracted from a patch which represents the center pixel
- The output is an edge confidence map
- Post-processed with non-maximal suppression



Probability of Boundary (Pb) and Its Two Variants (MS-Pb and gPb)

Probability-of-Boundary (Pb)

- Learning based on local features:
  - Brightness Gradient
  - Texture Gradient
  - Color Gradient



Martin et al. “Learning to detect natural image boundaries using local brightness, color, and texture cues” IEEE Trans Pattern Anal. Mach. Intell.26 (5) (2004) 530–549.

Probability-of-Boundary (Pb)

- Color: a, b
- Brightness: L
- Texture: textons (Convolve with 17 filters)

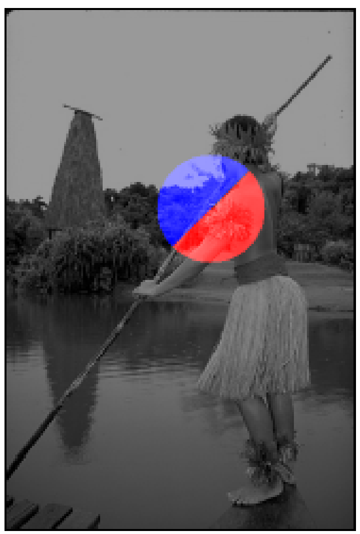
Filters for creating textons

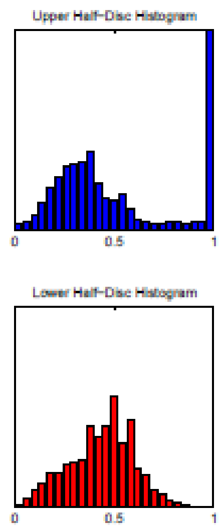
Channel

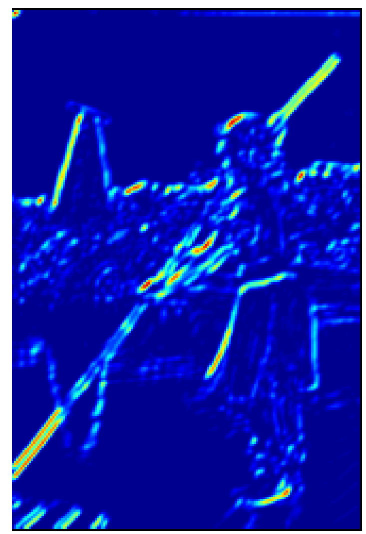
J. Malik "Representing and Recognizing the Visual Appearance Materials using Three-dimensional Textons" 2001 IJCV

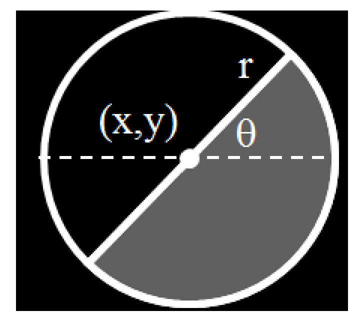
Probability-of-Boundary (Pb)

- Oriented gradient of histogram
  - Put disks with different scales (r) and orientations (θ)
  - Calculate the histogram difference between two half disks









χ^2 histogram difference operator

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

Martin et al. “Learning to detect natural image boundaries using local brightness, color, and texture cues” IEEE Trans Pattern Anal. Mach. Intell.26 (5) (2004) 530–549.

Probability-of-Boundary (Pb)

- Local Cue Combination

$$mPb(x,y,\theta) = \sum_{s}\sum_{i} \alpha_{i,s} G_{i,\sigma(i,s)}(x,y,\theta)$$

Learned from the training data

$\theta : [0, \pi)$

$\theta = 0$

•

•

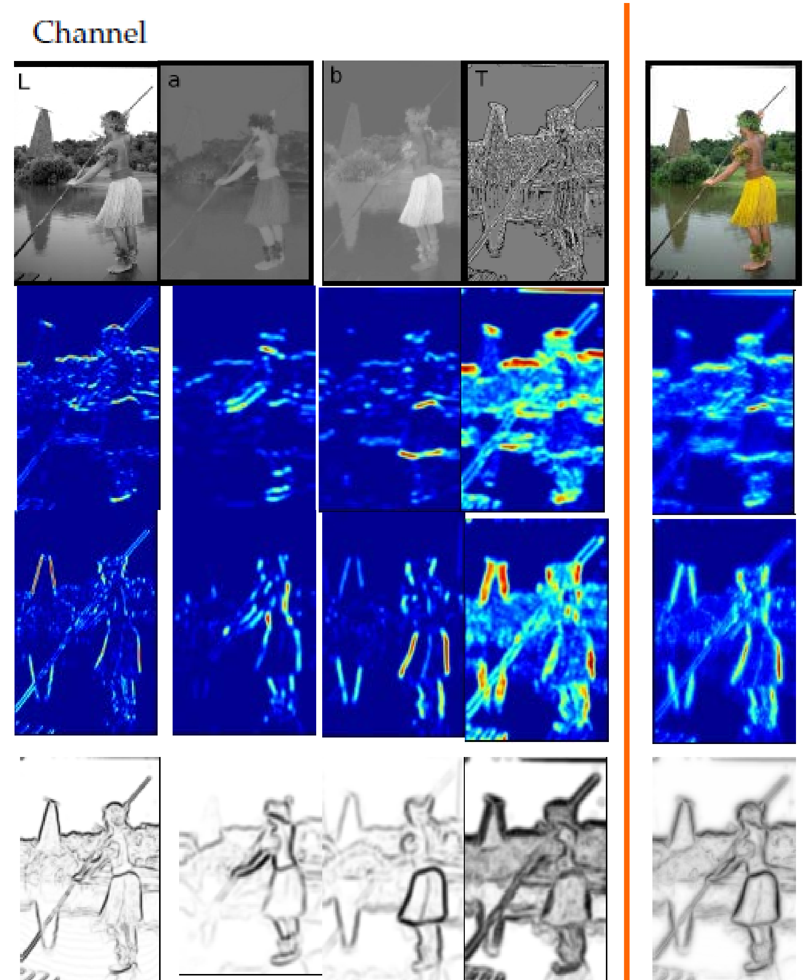
$\theta = \frac{\pi}{2}$

•

•

•

$G(x,y)$



$mPb(x,y)$

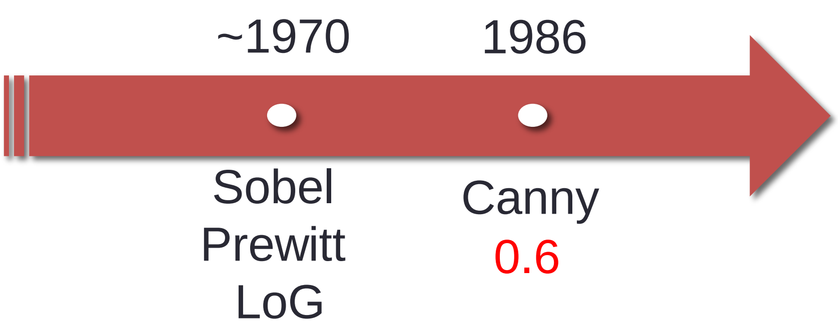
F-measure: 0.65

Maximum response over eight orientation

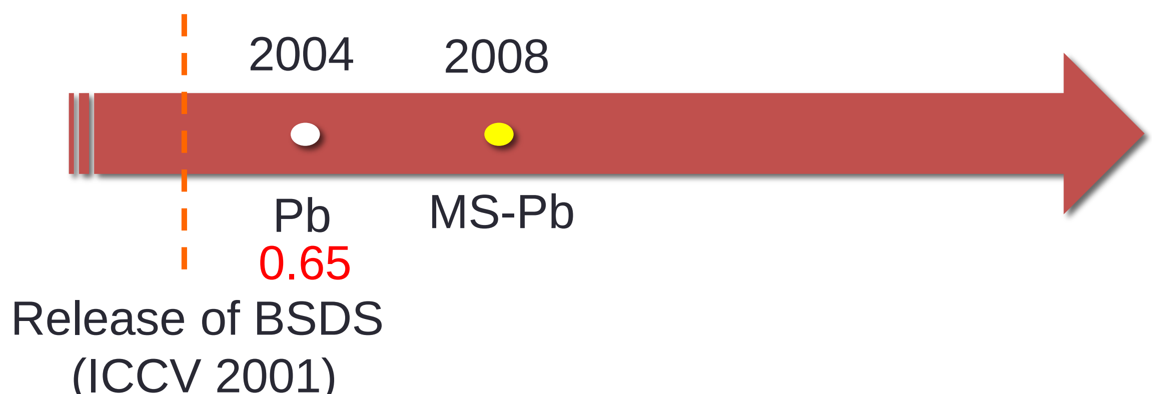
Martin et al. “Learning to detect natural image boundaries using local brightness, color, and texture cues” IEEE Trans Pattern Anal. Mach. Intell.26 (5) (2004) 530–549.

Prior Research Work

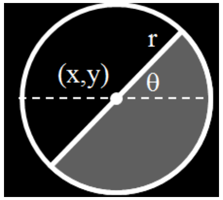
- Differentiation Based (HVS)



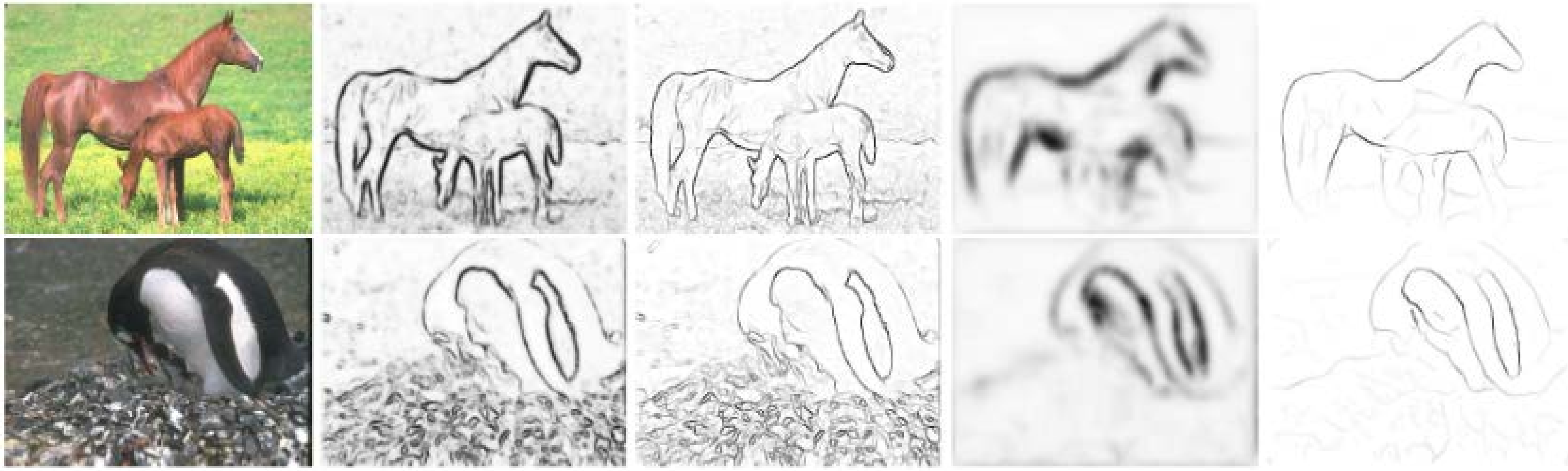
- Machine Learning Based (CVS)



Multi-Scale Probability-of-Boundary (MS-Pb)



- **Multi-Scale** Approach:
  - Small-scale output: (small radius)
    - Could capture detailed structures but suffers from false positives
  - Large-scale output: (large radius)
    - Reliable output but poor in localization

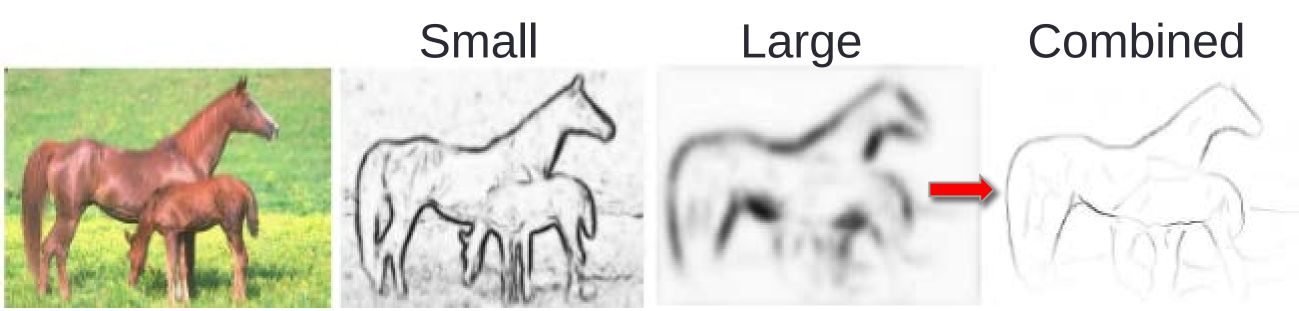


Input Small After nms Large After nms

"Multi-scale Improves Boundary Detection in Natural Images" in ECCV08

Multi-Scale Probability-of-Boundary (MS-Pb)

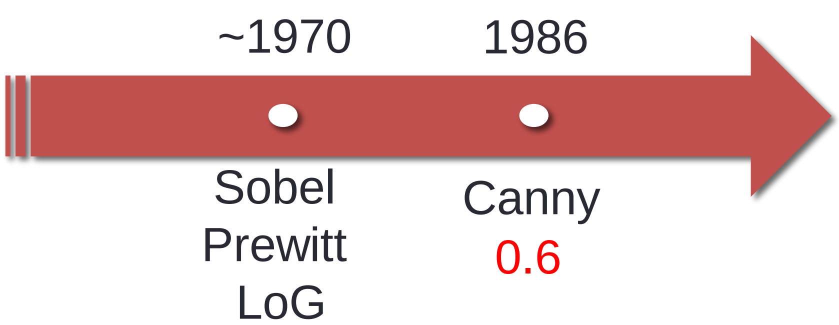
- Learning from ground truth to make the decision
  - Linear Classifier for every edge point in the finest scale
- Discriminative features between large-scale edges and small-scale edges
  - Contrast
  - Localization
- Here Multi-scale is more like **post-processing** method



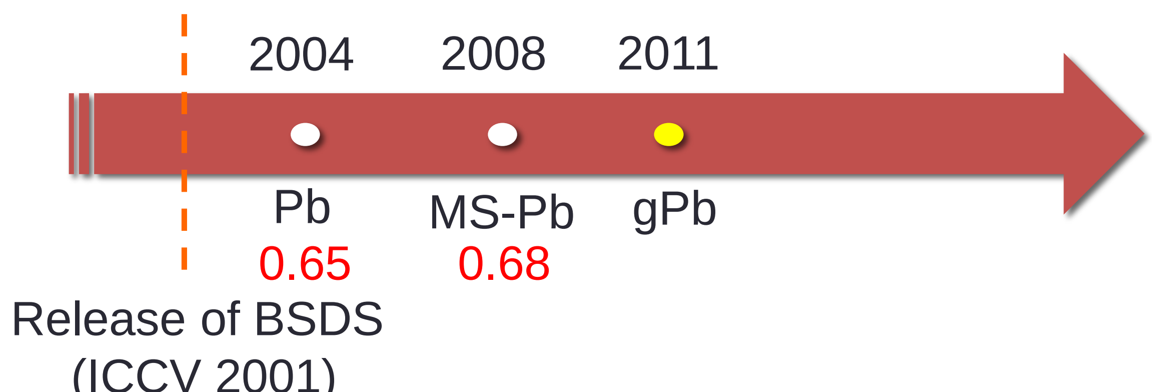
F-measure: 0.65 -> 0.68

Prior Research Work

- Differentiation Based (HVS)

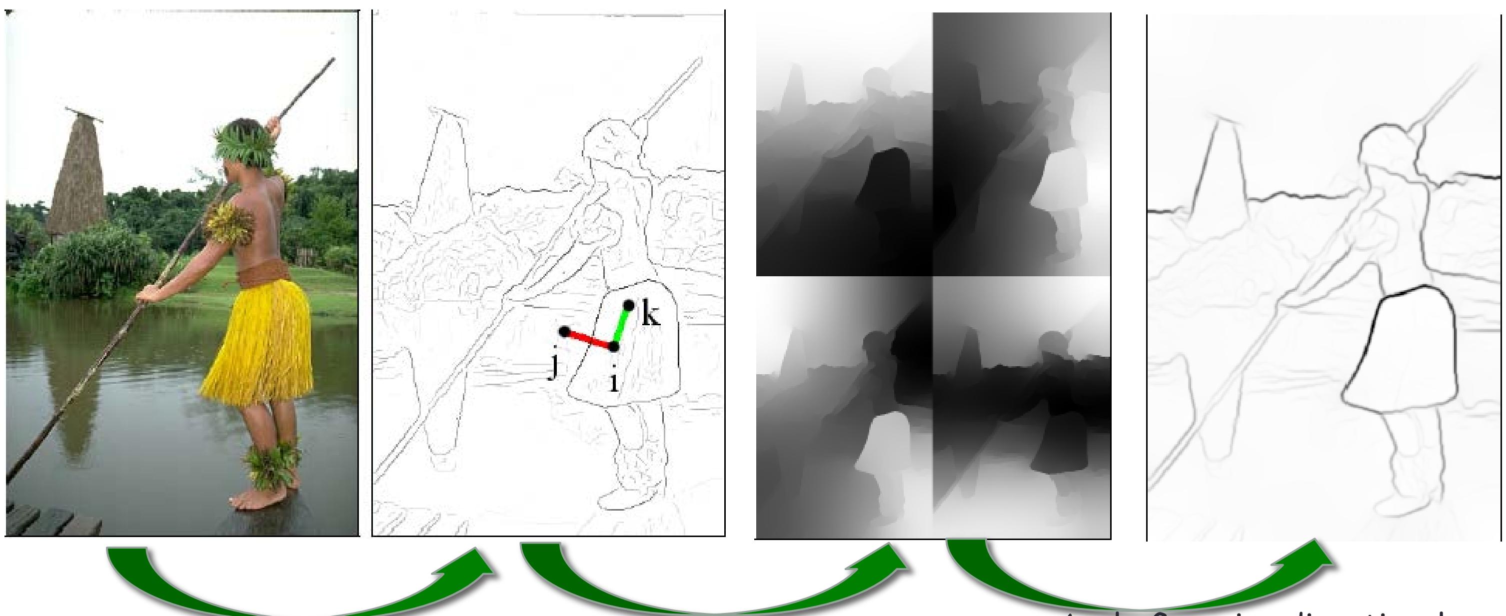


- Machine Learning Based (CVS)



Global Probability-of-Boundary (gPb)

- Extension of Pb algorithm
- Use spectral clustering to obtain **global information**
  - Distance measure based on Pb soft edge map



Affinity Matrix

$$W_{ij} = \exp\left(-\max_{p\in\overline{ij}}\{mPb(p)\}/\rho\right)$$

Eigenvectors capture important boundaries

Apply Gaussian directional filters and sum

Spectral Boundary

$$sPb(x,y,\theta) = \sum_{k=1}^{n} \frac{1}{\sqrt{\lambda_k}} \cdot \nabla_\theta \mathbf{v_k}(x,y)$$

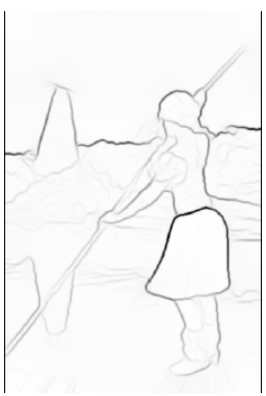
Arbeleaz et al. “Contour Detection and Hierarchical Image Segmentation” in PAMI 2011

Global Probability-of-Boundary (gPb)

- Combination by Logistic Regression
  - Globalization gPb (Pb + sPb)
  - Pb: local boundary (All edges) -> high recall
  - sPb: spectral boundary (most salient curves) -> high precision

$$gPb(x,y,\theta) = \sum_s \underbrace{\sum_i \beta_{i,s} G_{i,\sigma(i,s)}(x,y,\theta)}_{\text{Local Cues}} + \underbrace{\gamma \cdot sPb(x,y,\theta)}_{\text{Spectral cues}}$$

Learned from the training data



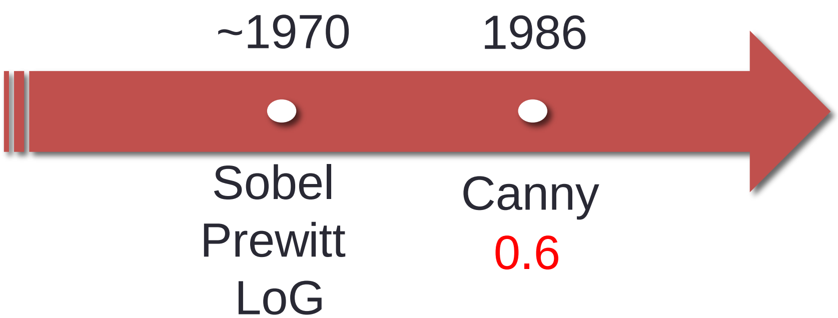
F-measure: 0.68 -> 0.70

Arbeleaz et al. "Contour Detection and Hierarchical Image Segmentation" in PAMI 2011

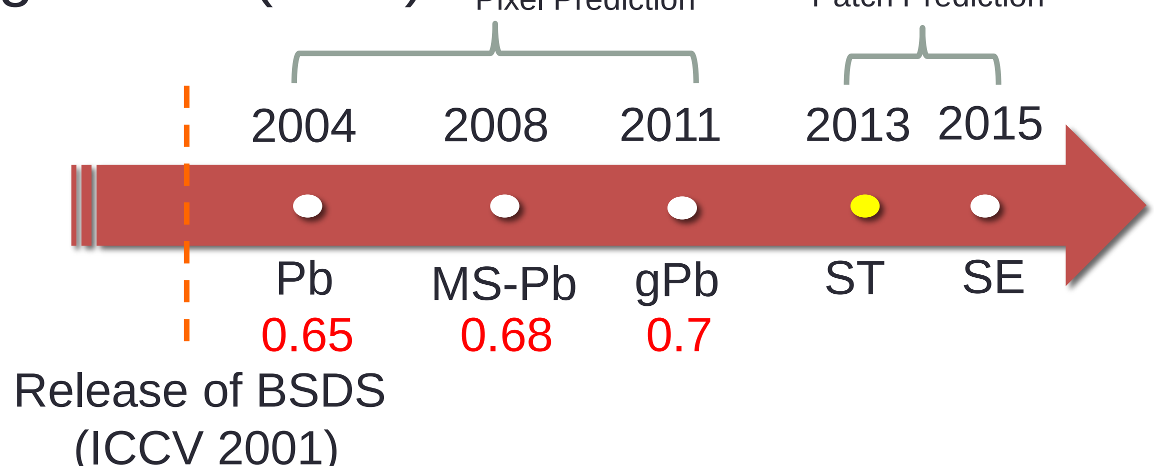
Sketch Token and Structured Edge

Prior Research Work

- Differentiation Based (HVS)

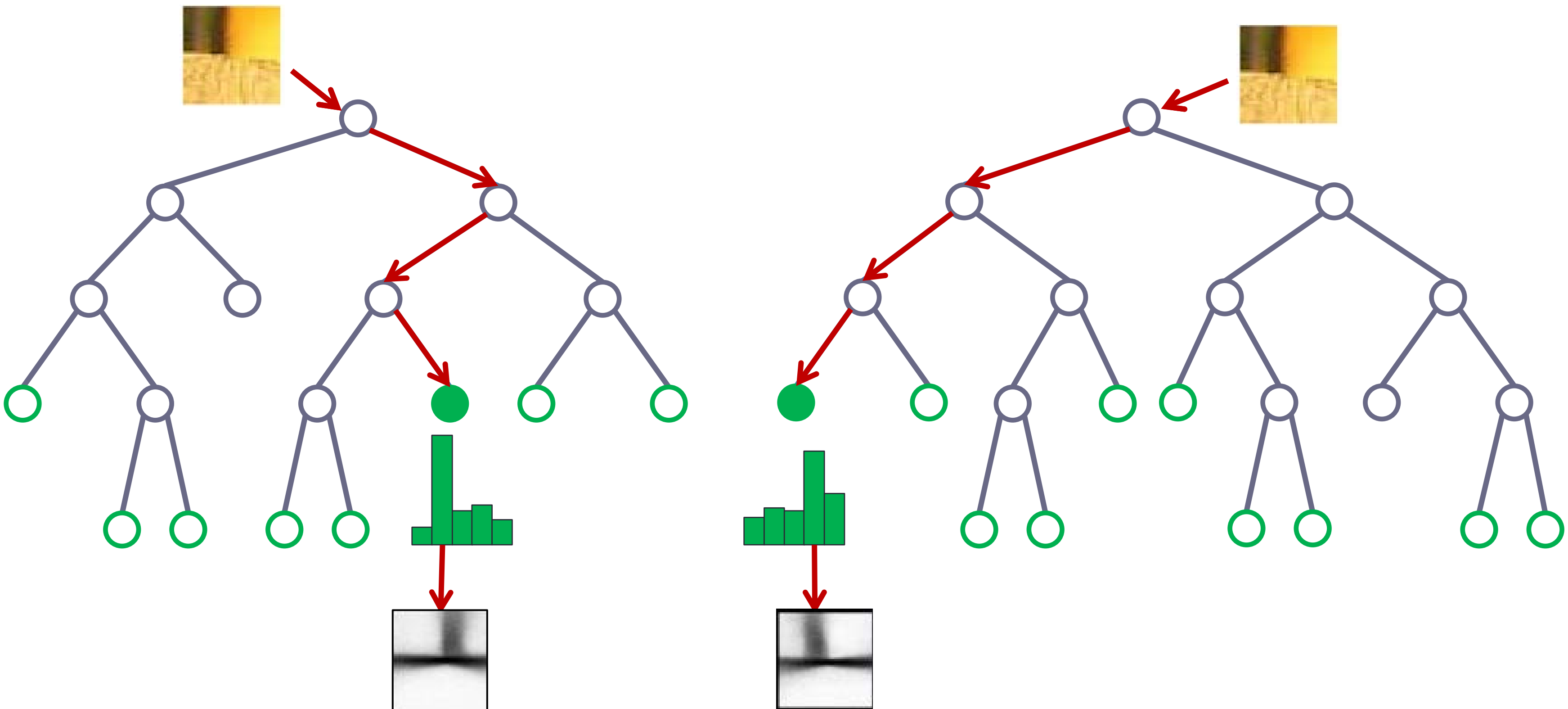


- Machine Learning Based (CVS)



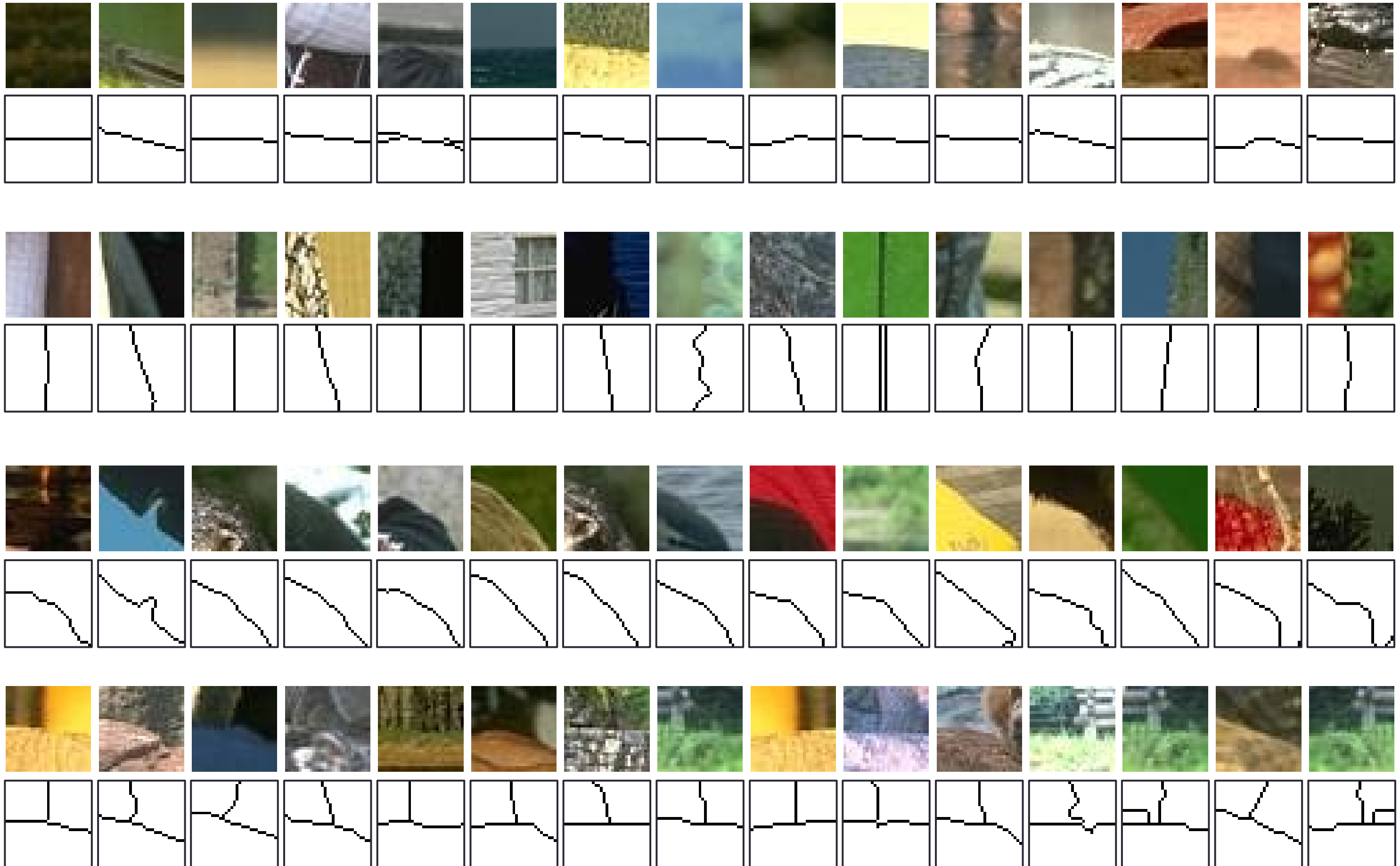
Why use Random Forest?

- Random Forest: Combination of decision trees
  - Low computation cost, High ability to select effective features
  - Not sensitive to the feature normalization
  - Sufficient diversity of trees could avoid the overfit problem



Random Forest Approach

- Two recent works: Sketch Token, Structured Edge
- Assumption: **Edge has structure**



Sketch Token (ST)

- Labels
  - Extract 35x35 patch from ground truth in the training dataset
    - The center pixel must be on the sketch
  - K-means clustering to categorize the “Tokens”
    - K = 150

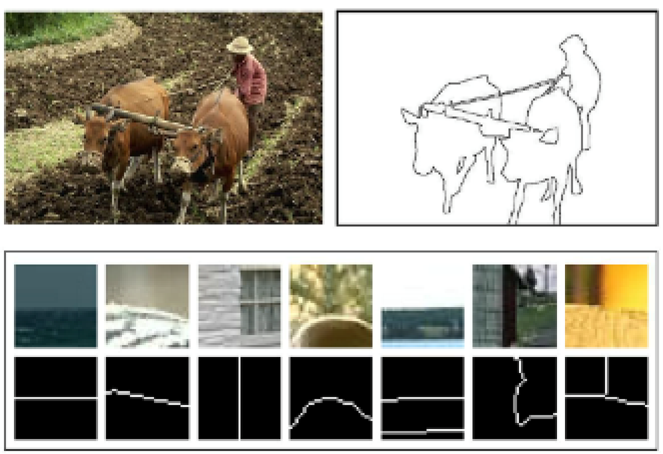


Figure 2. (Top) Example image and corresponding hand drawn sketch. (Bottom) Example image patches and their corresponding hand drawn contours.

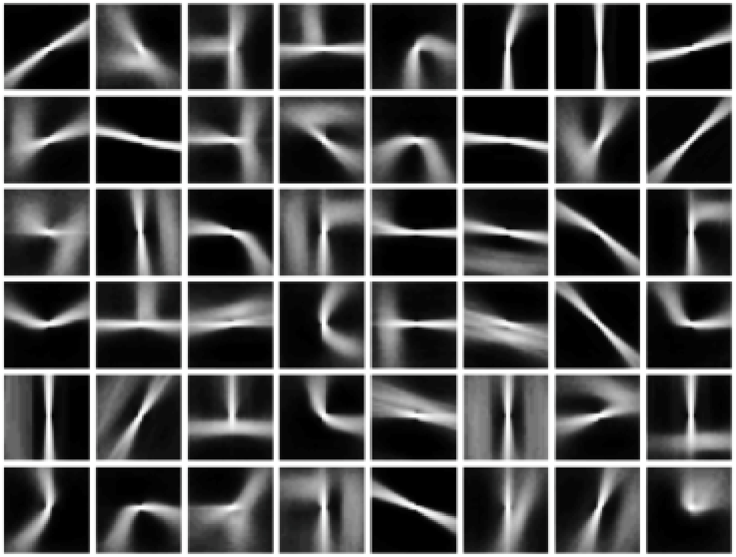
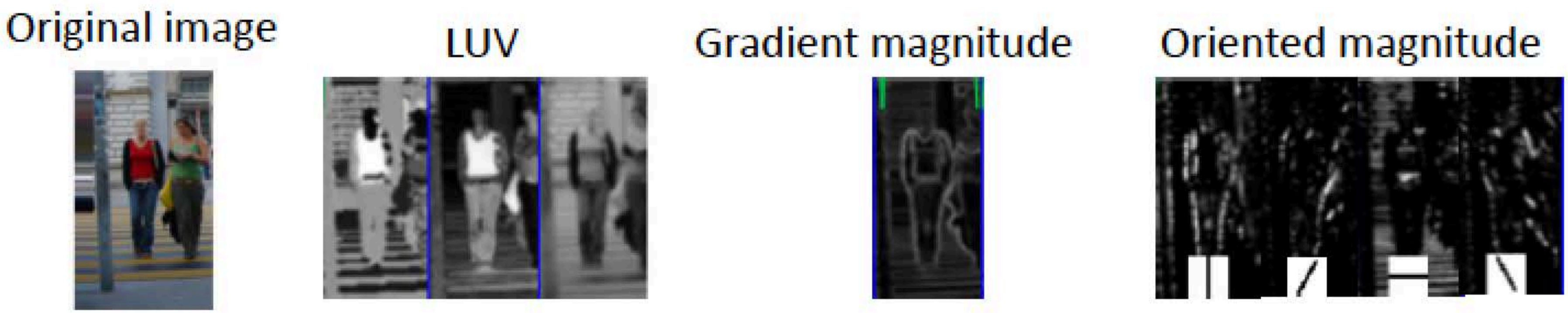


Figure 1. Examples of sketch tokens learned from hand drawn sketches represented using their mean contour structure. Notice the variety and richness of the sketch tokens.

J. Lim et al. “Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection” in CVPR 2013

Sketch Token (ST)

- Channel Features
  - CIE-LUV color space: 3 channels
  - Gradient magnitude: 3 channels
    - Blurred with σ=0, 1.5, 5 pixels
  - Oriented magnitude: 8 channels
    - Split 4 directions from gradient magnitude with σ=0, 1.5
  - Total basic feature channels = 3+3+8 = 14



J. Lim et al. "Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection" in CVPR 2013

Sketch Token (ST)

- Features (cont.)
  - Self-similarity
    - Aim to detect the “texture boundaries”
    - Down-sample to a 5x5 grid
    - Difference between every unit in the grid
      - C(5x5, 2) = 300 features per channel
  - Summary
    - Feature dimension = 35x35x14 + 300x14 = 21350
    - Computing the channels take only a fraction of a second

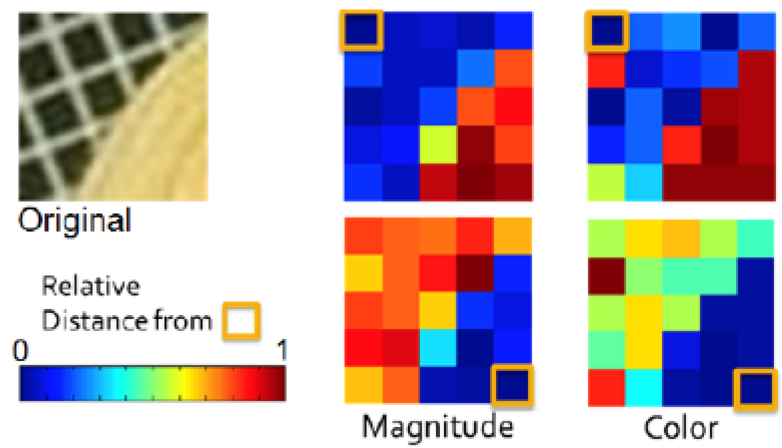
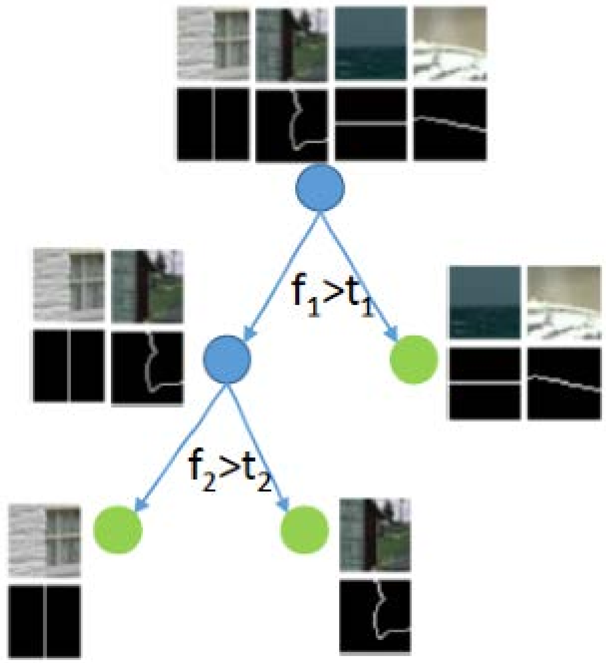


Figure 3. Illustration of the self-similarity features: The $L1$ distance $\sum_k |f_{ijk}|$ from the anchor cell (yellow box) to the other 5×5 cells are shown for color and gradient magnitude channels. The original patch is shown to the left.

J. Lim et al. “Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection” in CVPR 2013

Sketch Token (ST)

- How to train the random forest?
  - ➢For each node in each tree
    - – We want to gain more information after splitting
    - – Put the patches with same label to the same direction (Left or Right)



- Gini impurity (Entropy)

$$H(S) = \sum_y p_y(1-p_y)$$

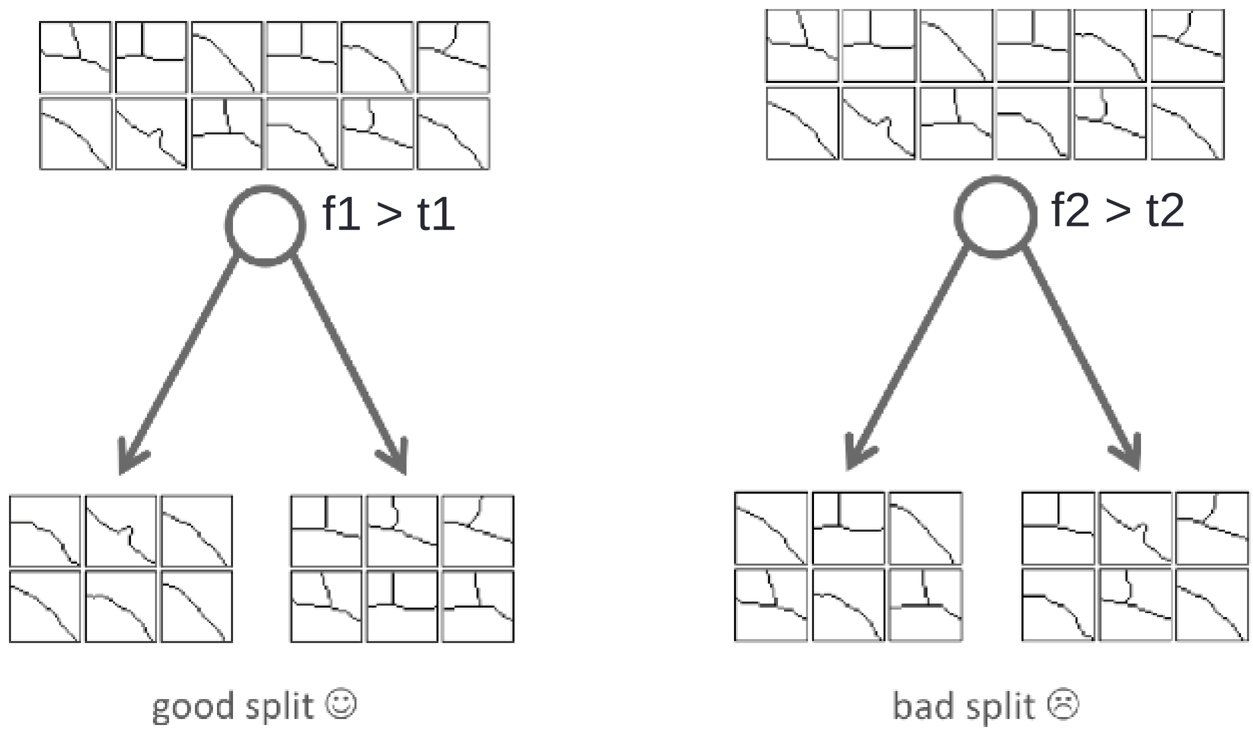
- Pick a feature (f) and a threshold (t) which maximize the information gain

$$I_j = \underbrace{H(S_j)}_{\text{The impurity before splitting}} - \underbrace{\sum_{k\in\{L,R\}} \frac{|S_j^k|}{|S_j|} H(S_j^k)}_{\text{The impurity after splitting}}$$

J. Lim et al. "Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection" in CVPR 2013

Sketch Token (ST)

- How to train the random forest?
  - For each node in each tree



	good split	bad split
Information Gain:	High	Low
Entropy:	Low	High

J. Lim et al. “Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection” in CVPR 2013

Sketch Token (ST)

- Random Forest Implementation
  - Randomly sample 150000 contour patches (positive patch)
    - 1000 per token class
  - Randomly sample 160000 “no contour” patches (negative patch)
    - 800 per training image
  - 25 trees are trained until every leaf node is pure enough

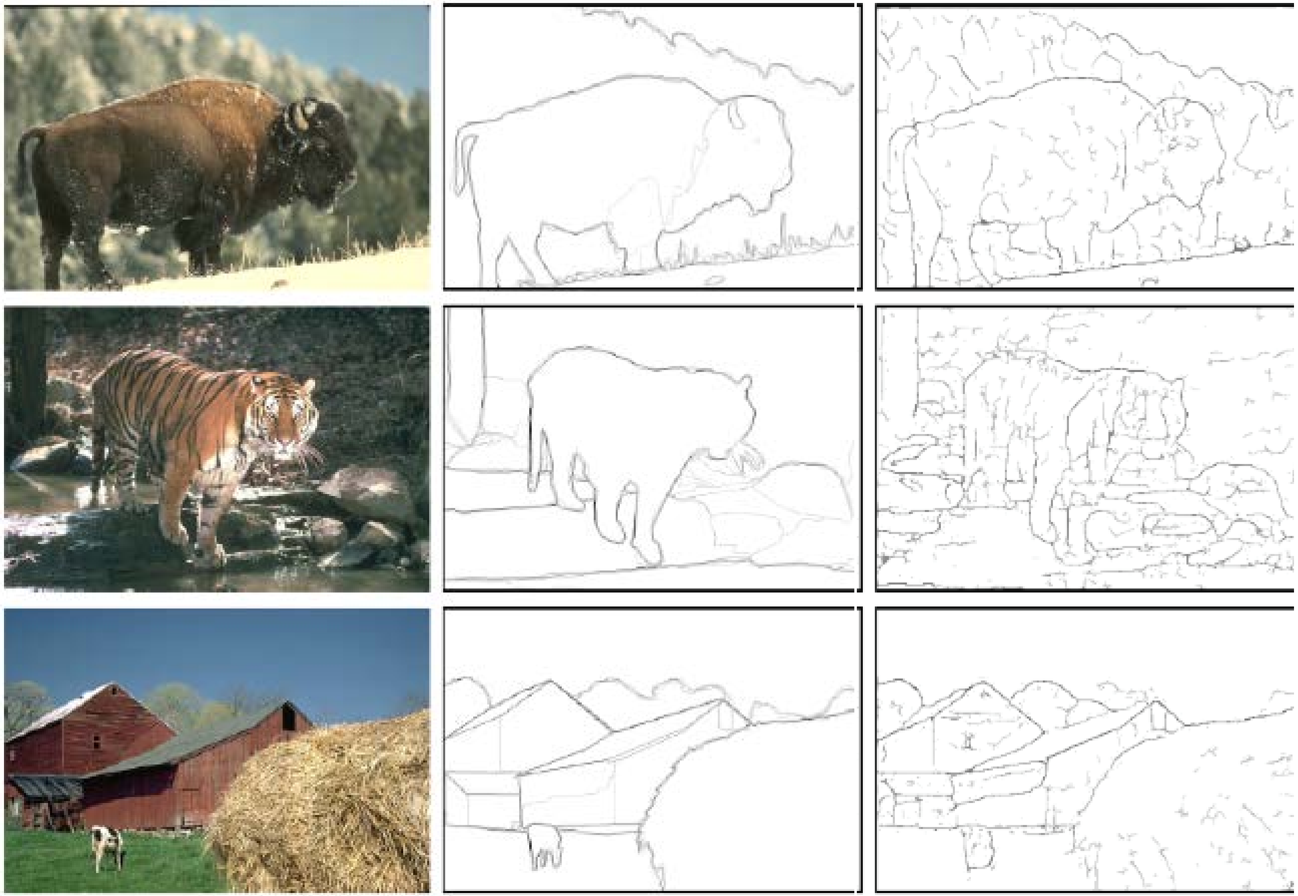
- How to predict the contour?
  - The results of 25 trees are averaged
  - Calculate the edge probability for each pixel

J. Lim et al. “Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection” in CVPR 2013

Sketch Token (ST)

- Results

F-measure: 0.70 -> 0.73

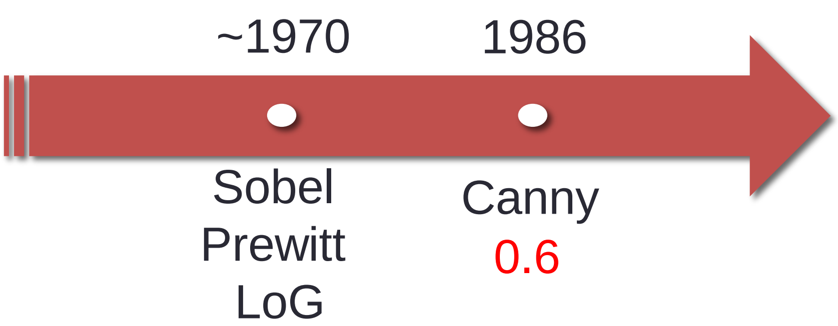


Input Ground Truth Sketch Token

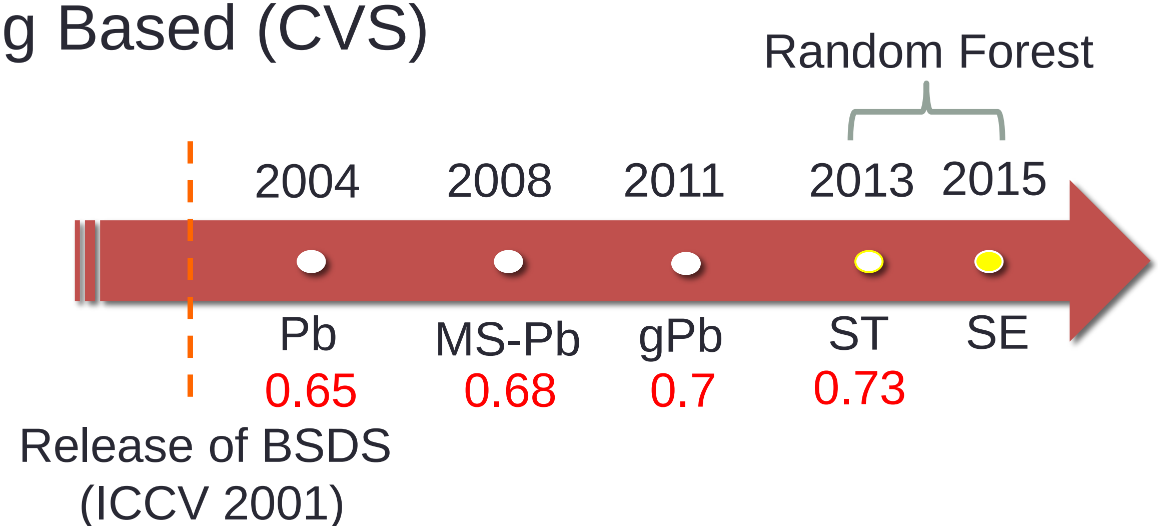
J. Lim et al. "Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection" in CVPR 2013

Prior Research Work

- Differentiation Based (HVS)



- Machine Learning Based (CVS)



Structured Edge (SE)

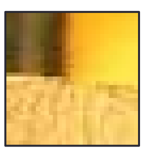
- Extension and Enhancement from Sketch Token (ST)
- **Use edge structure directly** instead of predefined labels

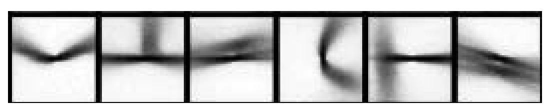
output space dimensionality

{ 0, 1 }

2

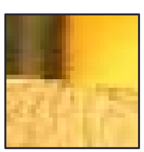
ST

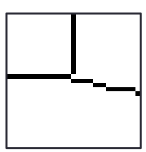


{  … }

151

SE





16x16 structure patch

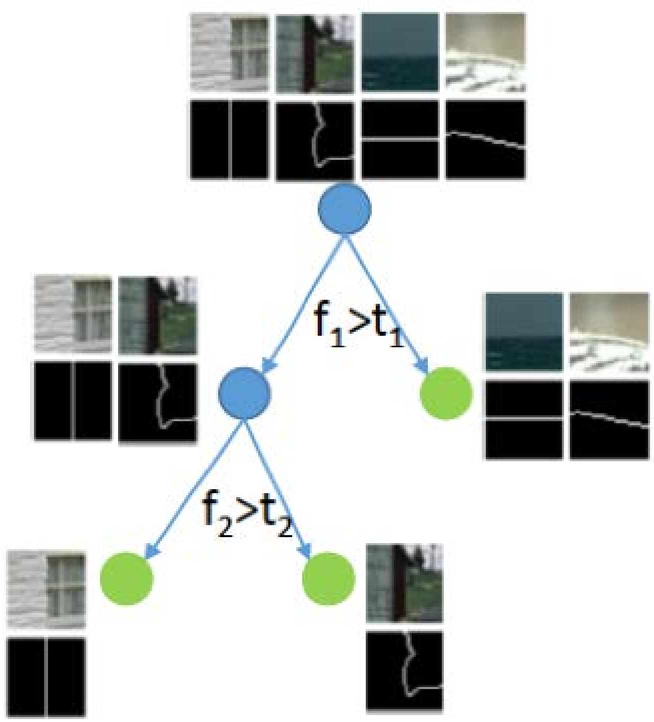
$\mathbf{2^{256}}$

Structured Output

P. Dollar et al. “Fast Edge Detection Using Structured Forests” in PAMI 2015

Structured Edge (SE)

- Main challenge of this enhanced framework
  - ➢ High output feature dimensions (**2**^{256})
  - ➢ Hard to calculate the information gain (too many labels)



- Gini impurity (Entropy) ?

$$H(S) = \sum_y p_y(1-p_y)$$

- <u>Pick a feature (f) and a threshold (t)</u> which maximize the information gain

$$I_j = \underbrace{H(S_j)}_{\text{The impurity before splitting}} - \underbrace{\sum_{k\in\{L,R\}} \frac{|S_j^k|}{|S_j|} H(S_j^k)}_{\text{The impurity after splitting}}$$

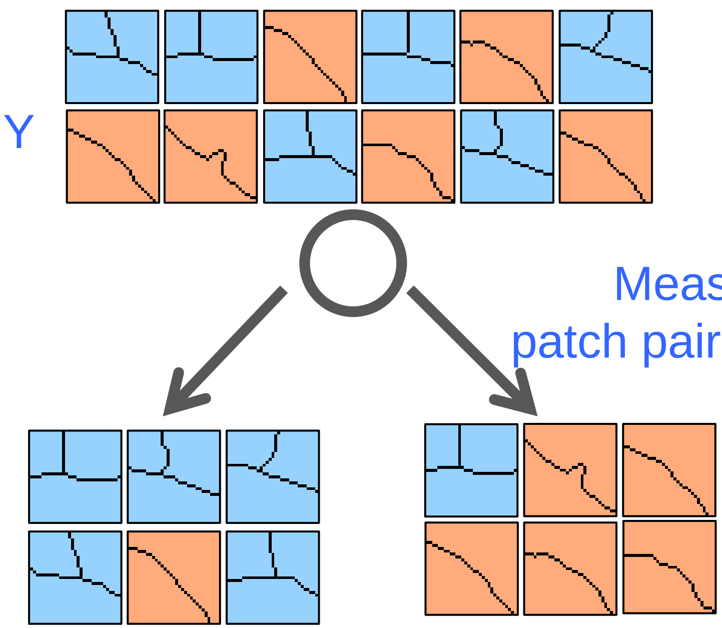
?

P. Dollar et al. “Fast Edge Detection Using Structured Forests” in PAMI 2015

Structured Edge (SE)

- Output Space Dimension Reduction Strategy
  - ➢ Output space Y is not random
    - – Patch is segmented into different regions by closed contours
  - ➢ Z encode the information whether every pair in Y belong to the same segment or not (C(256, 2) = 32640 dimensions)

Output Space
Segment Mask Y



Y -> Z

2^{256} 32640

Measure the similarity of
patch pairs to split into two classes

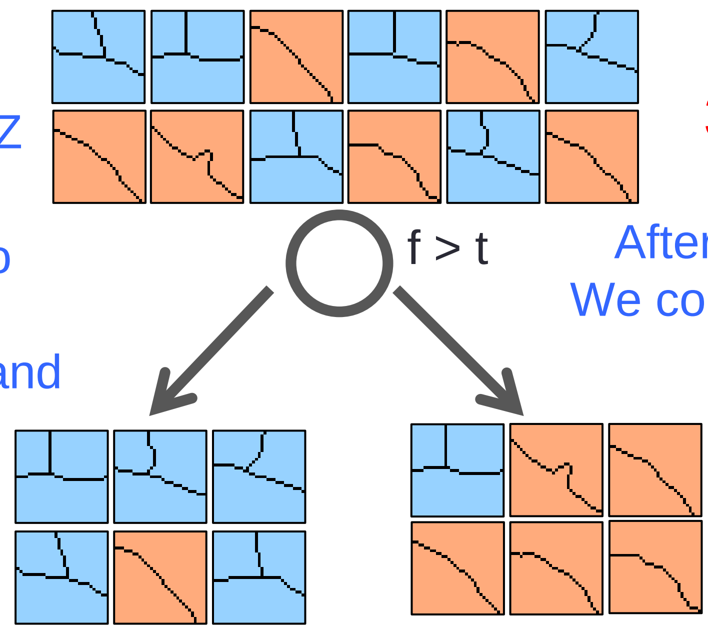
P. Dollar et al. “Fast Edge Detection Using Structured Forests” in PAMI 2015

Structured Edge (SE)

- Output Space Dimension Reduction Strategy
  - Sample 256 dimensions in Z then reduce to 5 dimensions by PCA
  - K-means (k=2) to classify into two classes (C) **for each node**

Output Space
Difference measure Z

Z -> C
32640 2



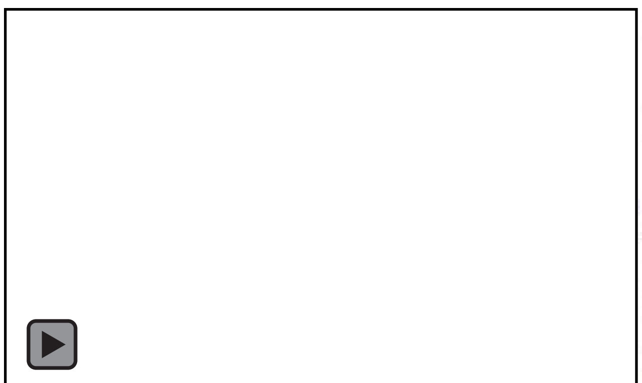
PCA and K-means to separate the patch into two classes (blue and orange label)

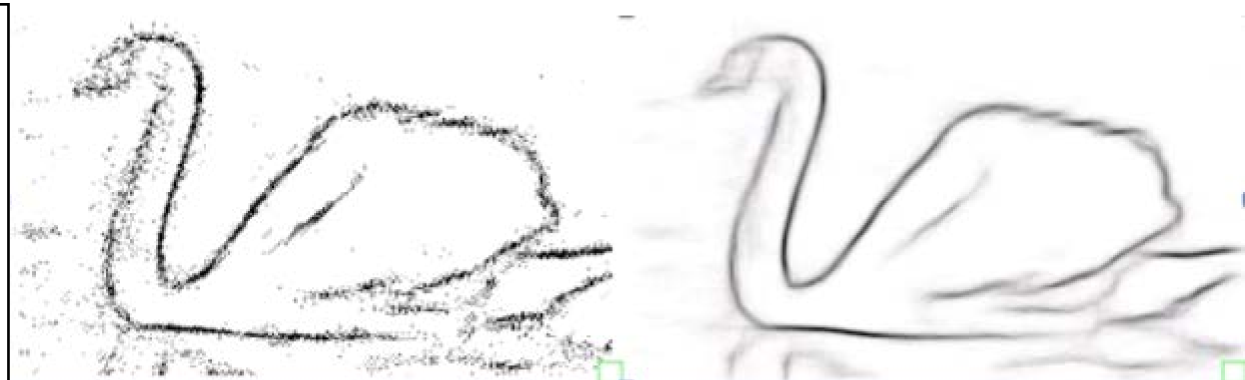
After defining the label
We could select the f and t

P. Dollar et al. “Fast Edge Detection Using Structured Forests” in PAMI 2015

Structured Edge (SE)

- How to predict the contour structure?
  - ➢ Average the response from each tree to obtain the soft edge map
  - ➢ Sliding prediction window (16x16)



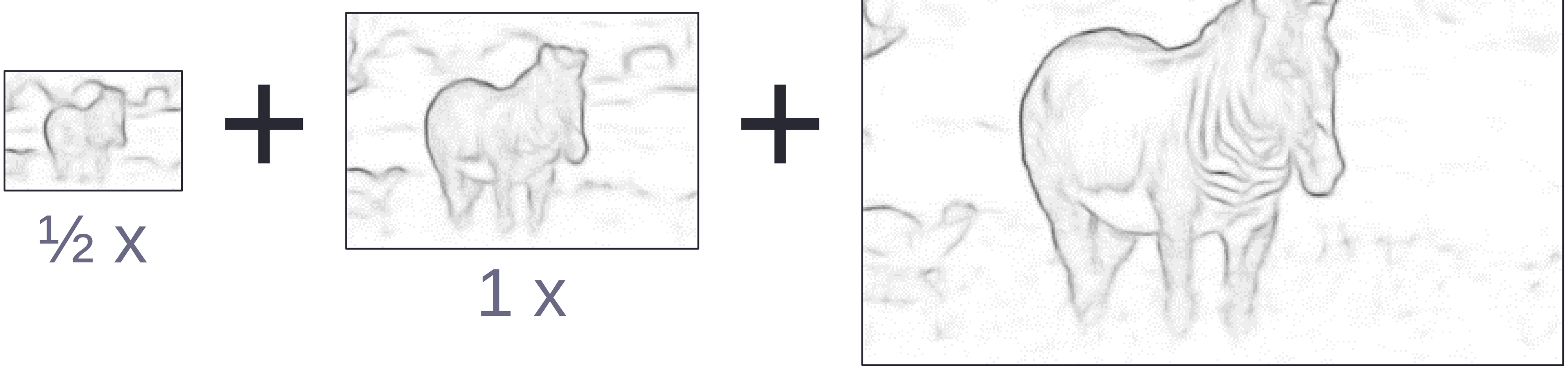


pixel output ☹ structured output ☺

P. Dollar et al. "Fast Edge Detection Using Structured Forests" in PAMI 2015

Structured Edge (SE)

- Multi-scale combination

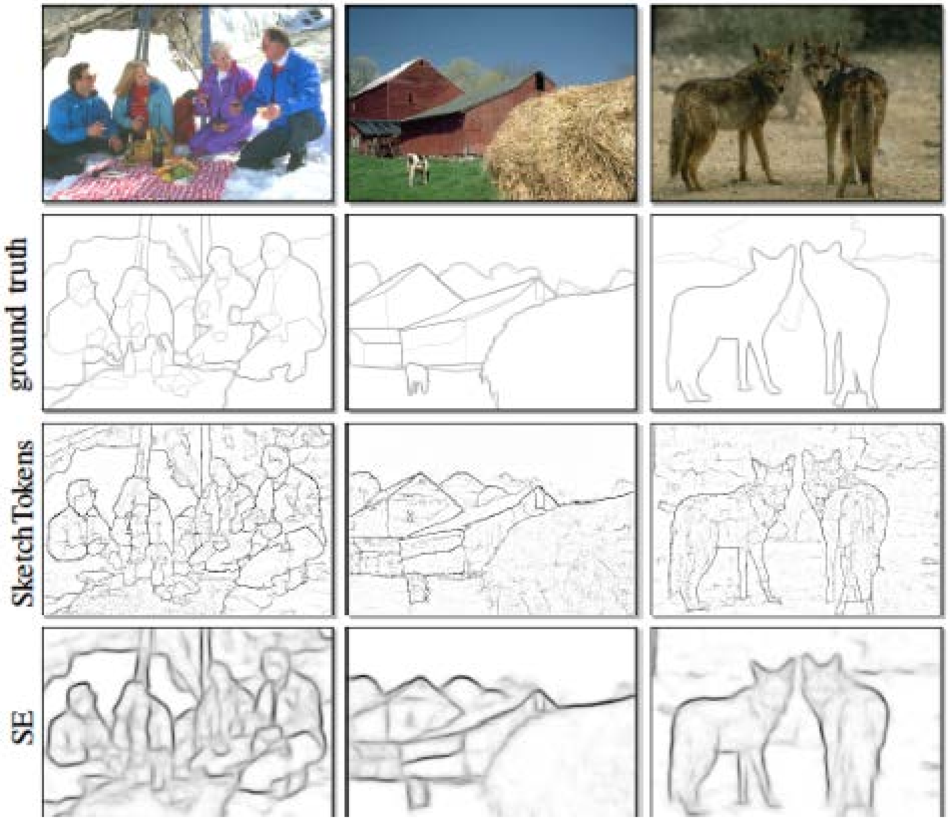


P. Dollar et al. “Fast Edge Detection Using Structured Forests” in PAMI 2015

Structured Edge (SE)

- Advantage
  - Very Fast computation
  - More accurate than Sketch Token (ST)

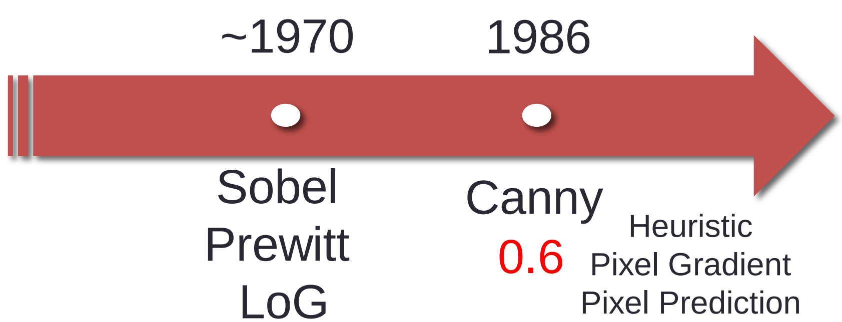
F-measure: 0.73 -> 0.74



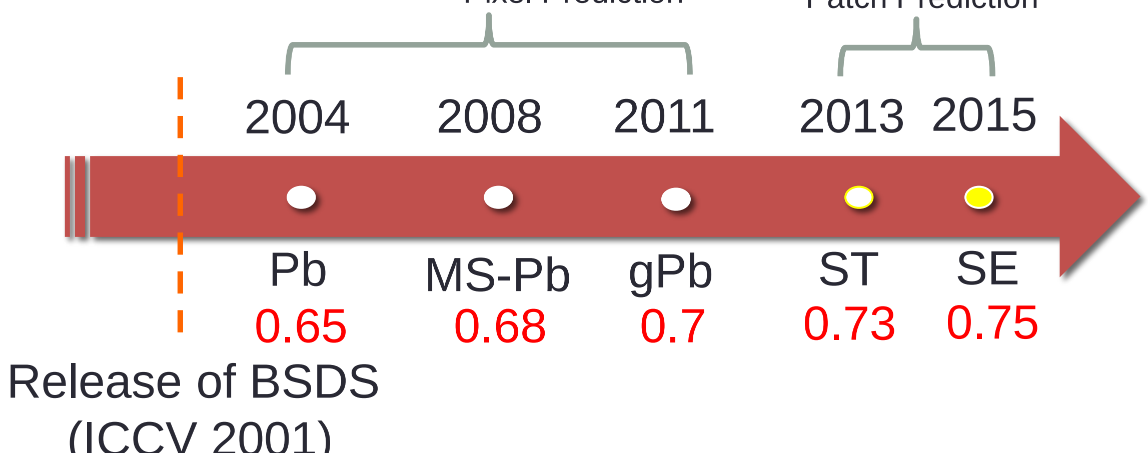
	ODS	OIS	AP	FPS
Human	.80	.80	-	-
Canny	.60	.64	.58	15
Felz-Hutt [11]	.61	.64	.56	10
Hidayat-Green [16]	.62†	-	-	20
BEL [9]	.66†	-	-	1/10
gPb + GPU [6]	.70†	-	-	1/2‡
gPb [1]	.71	.74	.65	1/240
gPb-owt-ucm [1]	.73	**.76**	.73	1/240
Sketch tokens [21]	.73	.75	**.78**	1
SCG [31]	**.74**	**.76**	.77	1/280
SE-SS, T=1	.72	.74	.77	**60**
SE-SS, T=4	.73	.75	.77	30
SE-MS, T=4	**.74**	**.76**	**.78**	6

Prior Research Work

- Differentiation Based (HVS)

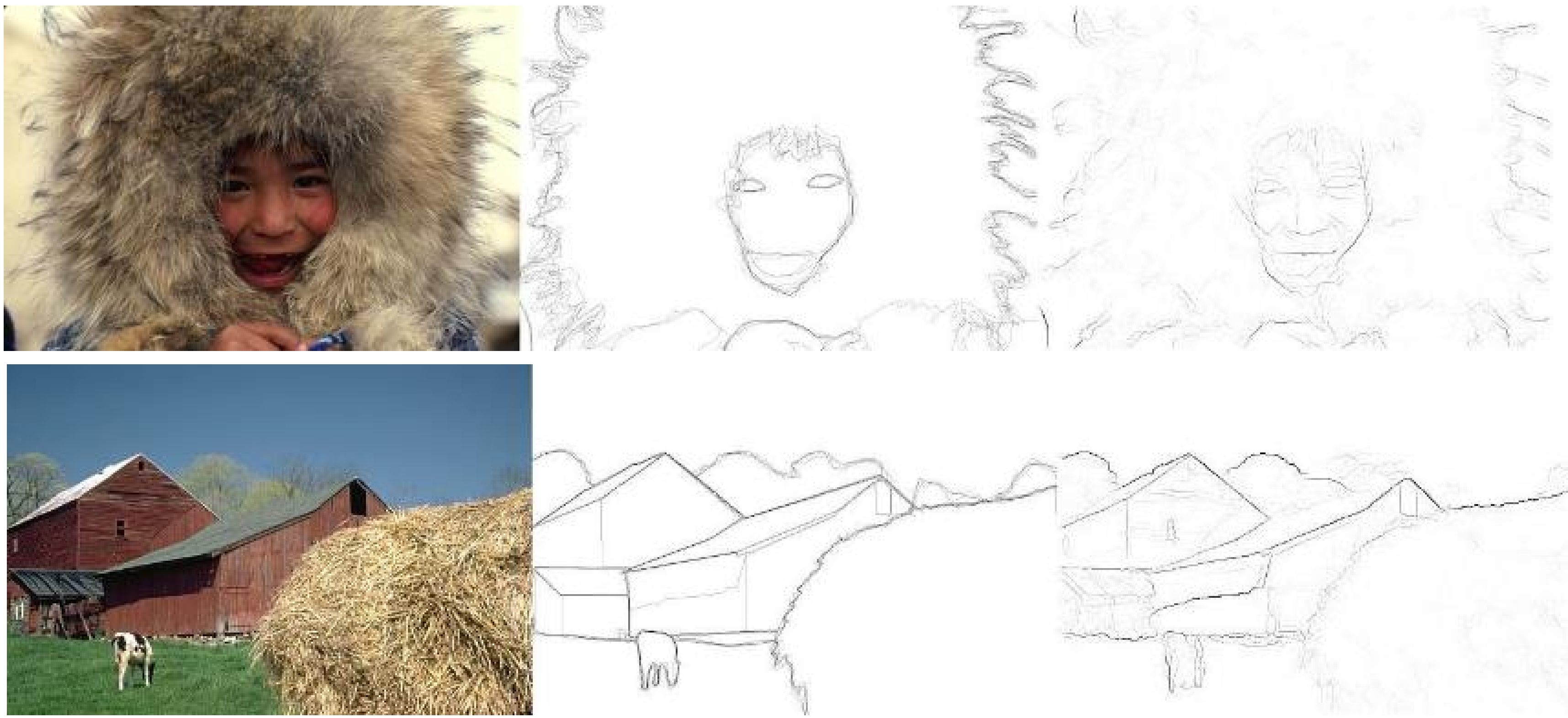


- Machine Learning Based (CVS)



Analysis of Structured Edge

- How good is Structured Edge Detector?
  - ➢ Texture parts can be ruled out
  - ➢ Suitable for every kind of local structure



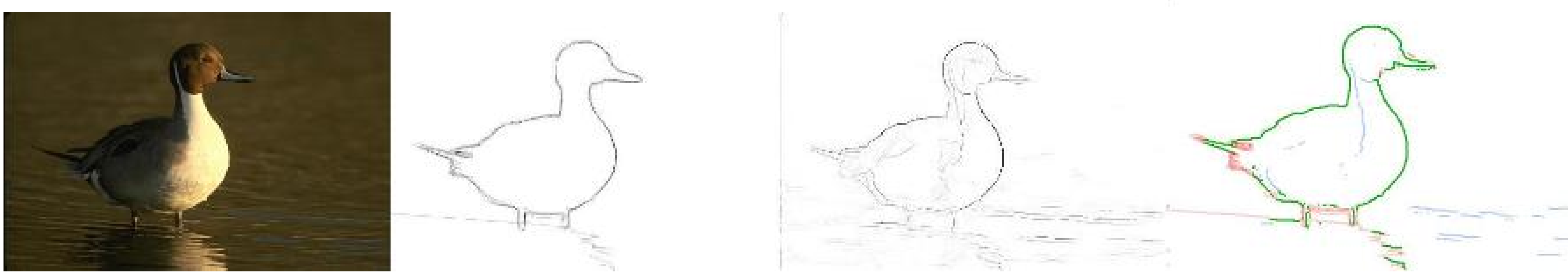
Ground Truth Structured Edge

Analysis of Structured Edge

- Where's the error?

Blue: Need to be suppressed

Red: Need to be recover



Conclusion and Future Work

- Local learning-based contour detector has been well developed
- Global information is required to obtain more accurate result
- Intrinsic evaluation problems exist in this field
- Future Directions
  - Find more discriminative features
  - Involve global information to refine the contour map
  - Propose a generative model just like Human Vision
  - Produce results that are suitable for downstream applications