Evaluating Segmentation

David Martin
dmartin@cs.bc.edu
Computer Science Department
Boston College

How do you know when a segmentation algorithm is good?

How do you know when a segmentation algorithm is good?

(1) The results look good on these two images.*

* Not acceptable!

How do you know when a segmentation algorithm is good?

(2) Higher performance on the standard vision application test suite as a result of using the improved segmentation sub-system.*

* see, e.g. “A Complete Implementation of the Human Visual Cortex”, Fowlkes, Martin, Sharon, Shi, CVPR 2020.

How do you know when a segmentation algorithm is good?

(3) It performed well on a standard segmentation benchmark.*

* Let's talk!

Benchmarks are Good

Requirements:

- A clear problem specification.
- A representative dataset.
- An evaluation methodology.

Successes:

- Handwritten digit recognition (MNIST)
- Face recognition (FERET, CMU PIE)
- Object recognition (COIL, Trademarks, Web Images)

Layered Benchmarks: A Necessary Result of Complexity

High-level end-to-end application-level benchmarks

- Systems: Database TPS on TPC workload
- Vision: DARPA-sponsored robot race across So. Cal. desert

Mid-level sub-system benchmarks

- Systems: Lock manager, block cache, TCP stack, query planner
- Vision: Tracking, segmentation, pose estimation

Low-level micro-benchmarks

- Systems: CPU functional units/cache, disc seeks
- Vision: Edge detection, optical flow

Layered Benchmarks: Where are the challenges?

High-level end-to-end application-level benchmarks

- Hard: Specification, Dataset
- Easy: Measurement

Mid-level sub-system benchmarks

- Hard: Specification, Dataset, Measurement
- Easy: None

Low-level micro-benchmarks

- Hard: Measurement
- Easy: Dataset, Specification

Def: Segmentation

(1) A division of the pixels of an image into disjoint groups (where the groups correspond to objects or parts of objects).

Def: Segmentation

(1) A division of the pixels of an image into disjoint groups.

(2) The partition given by a binary pixel affinity function s, where s(i,j)=1 when pixels i and j belong together, and s(i,j)=0 otherwise.

Step #1: Define Segmentation

(1) A division of the pixels of an image into disjoint groups.

(2) The partition given by a binary pixel affinity function s(i,j).

(3) The set of edge elements (edgels) denoting segment boundaries. An edgel has a location and orientation.

Step #2: Representative Dataset

Step #2.5: Groundtruthing

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)

Dataset Summary

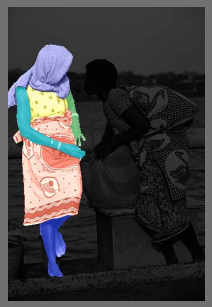
- 30 subjects, age 19-23
  - 17 men, 13 women
  - 9 with artistic training
- 8 months
- 1,458 person hours
- 1,020 Corel images
- 11,595 Segmentations
  - 5,555 color, 5,554 gray, 486 inverted/negated

What is your s(i,j)?

Do you even have one?

Percept Tree


Basket Water Rail Trees Sky



Left Woman

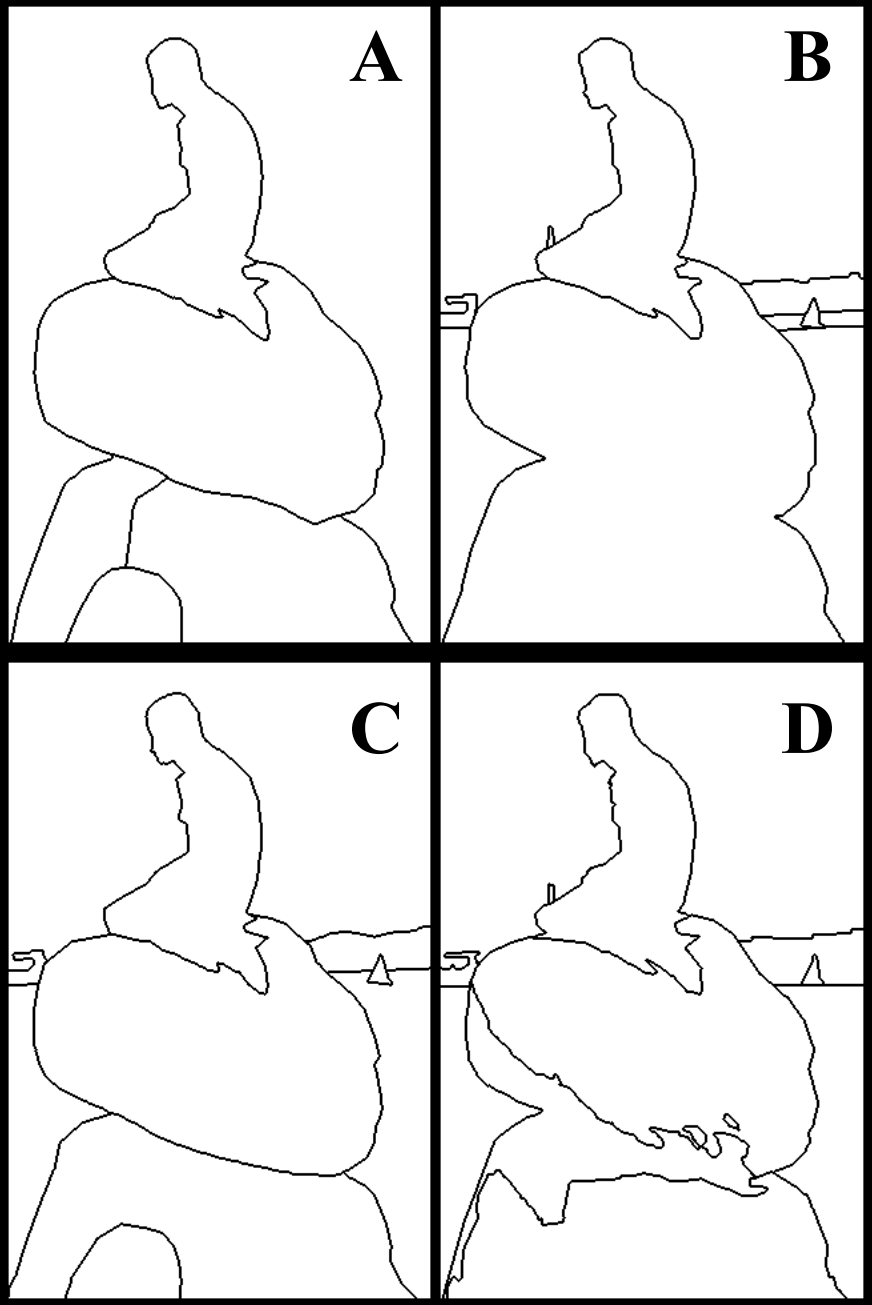
Head Shirt Skirt Arm Arm Legs

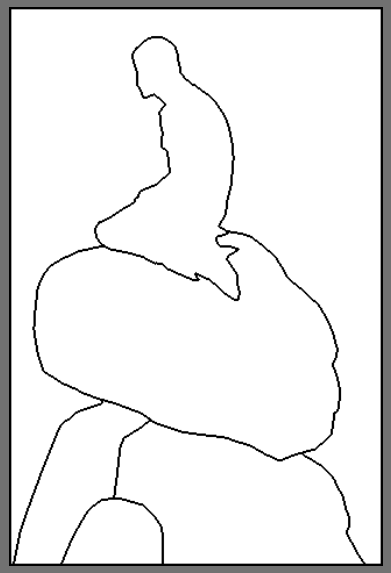
Bridge

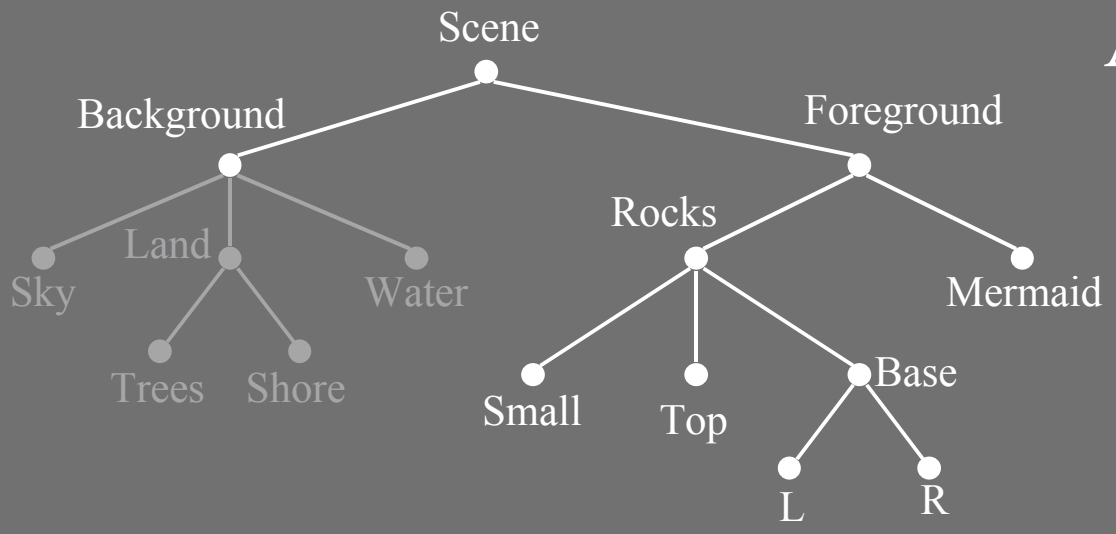
Box Walk Curb Road

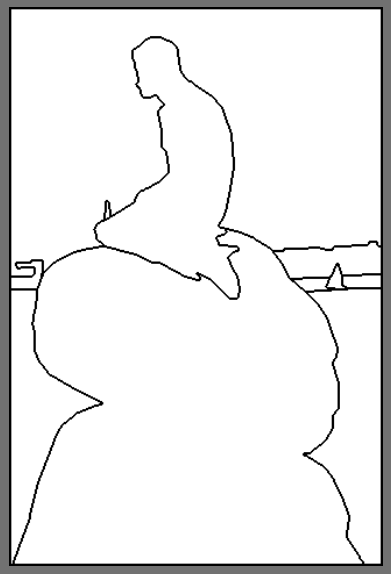
Right Woman

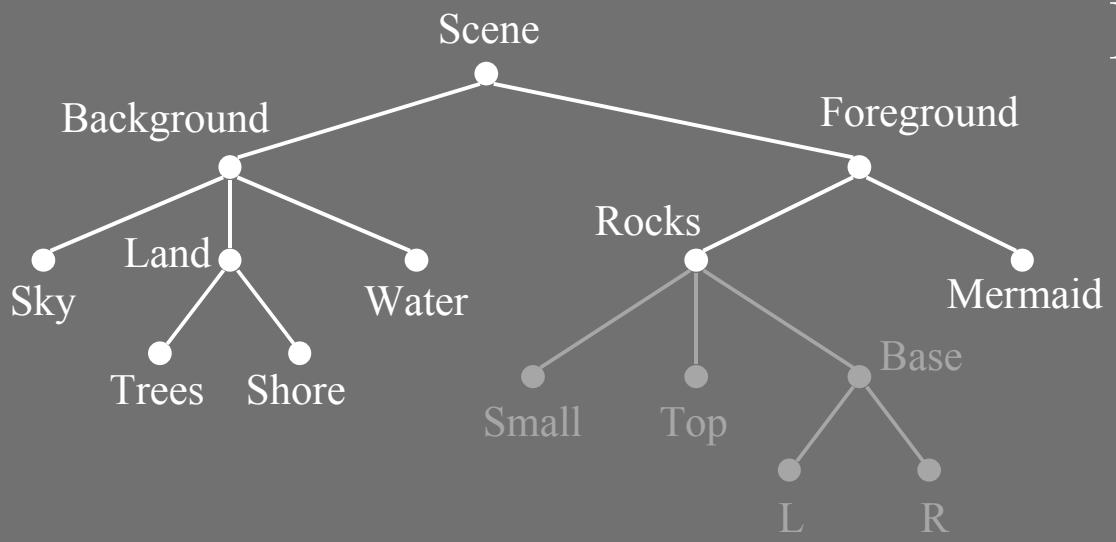
Head Shirt Pack Arms Bag Skirt Leg Leg

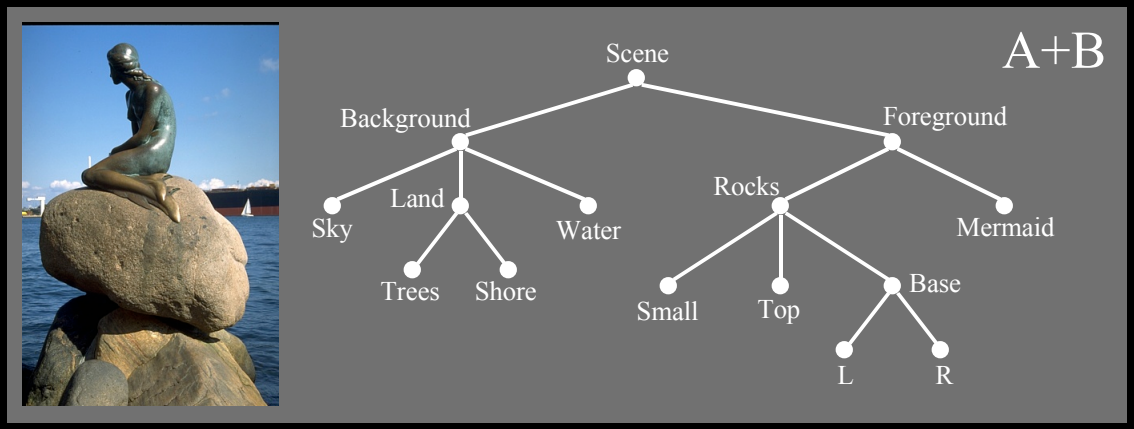

A
B
C
D

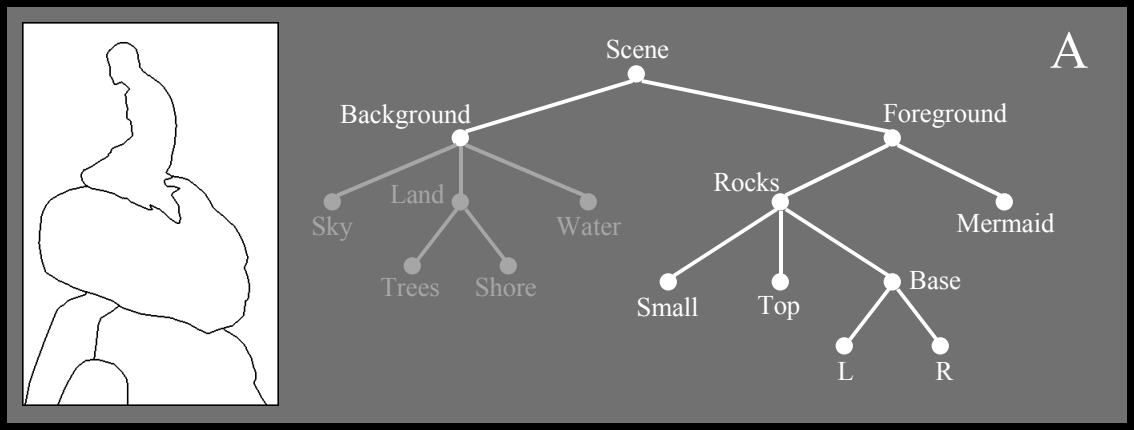


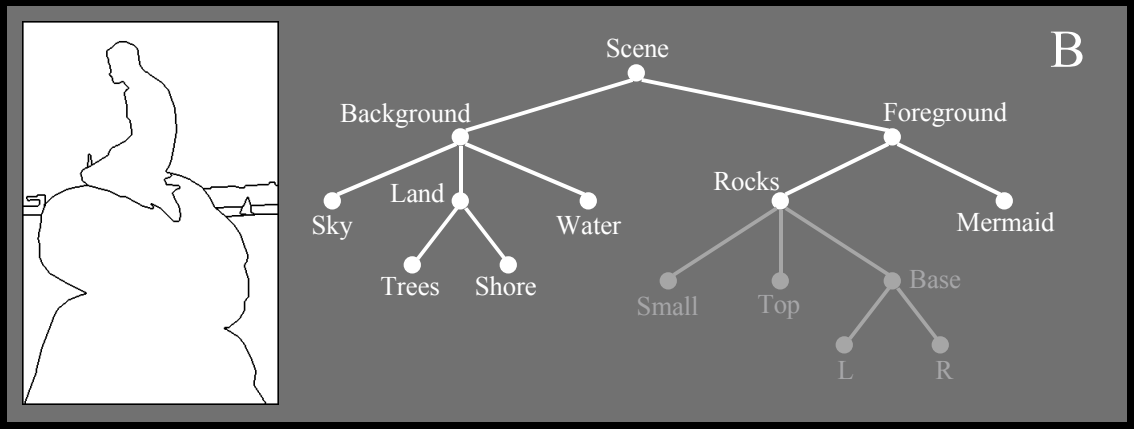

A
Scene
Background
Foreground
Sky
Land
Water
Rocks
Mermaid
Trees
Shore
Small
Top
Base
L
R

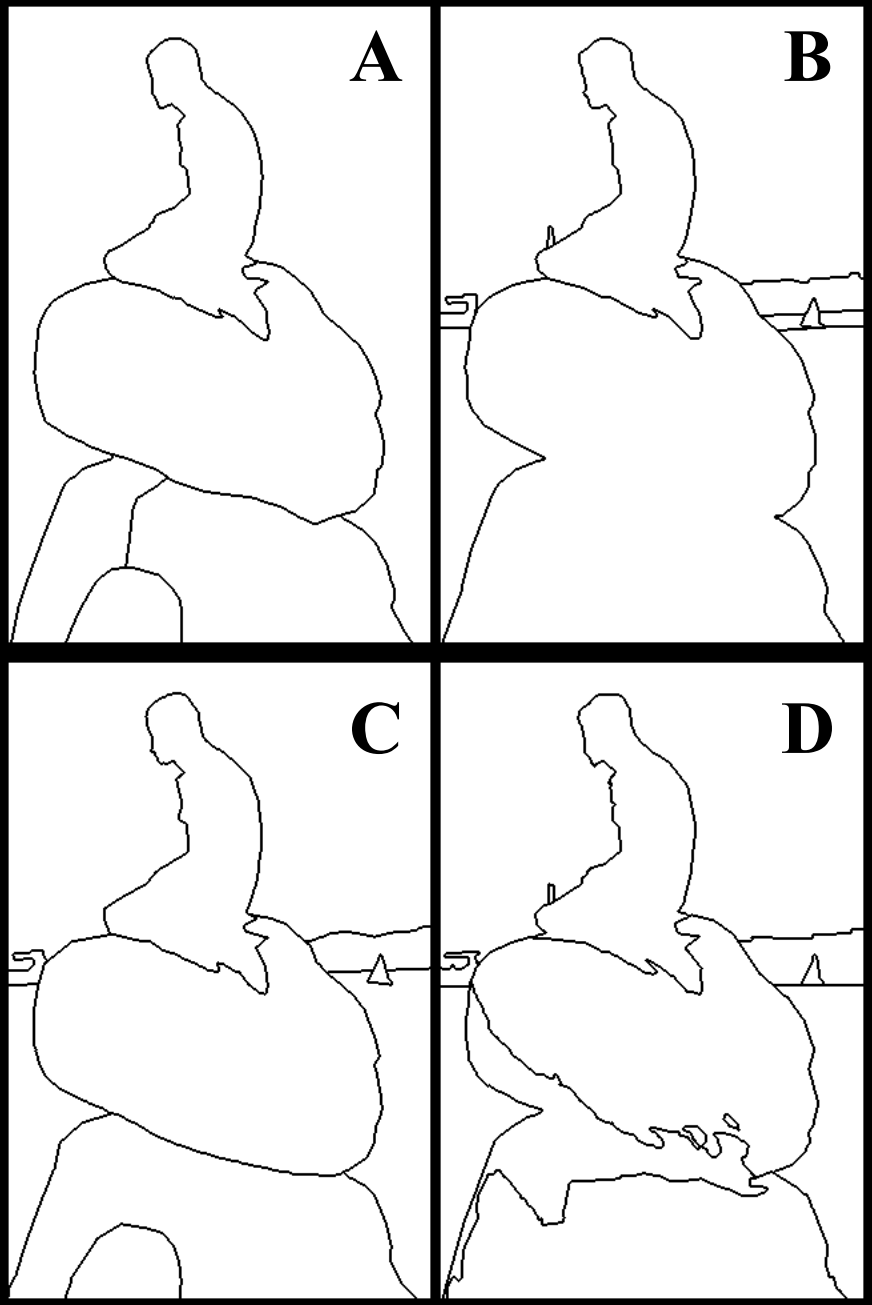



B
Scene
Background
Foreground
Sky
Land
Water
Rocks
Mermaid
Trees
Shore
Small
Top
Base
L
R


A+B
Scene
Background
Foreground
Sky
Land
Water
Rocks
Mermaid
Trees
Shore
Small
Top
Base
L
R


A
Scene
Background
Foreground
Sky
Land
Water
Rocks
Mermaid
Trees
Shore
Small
Top
Base
L
R

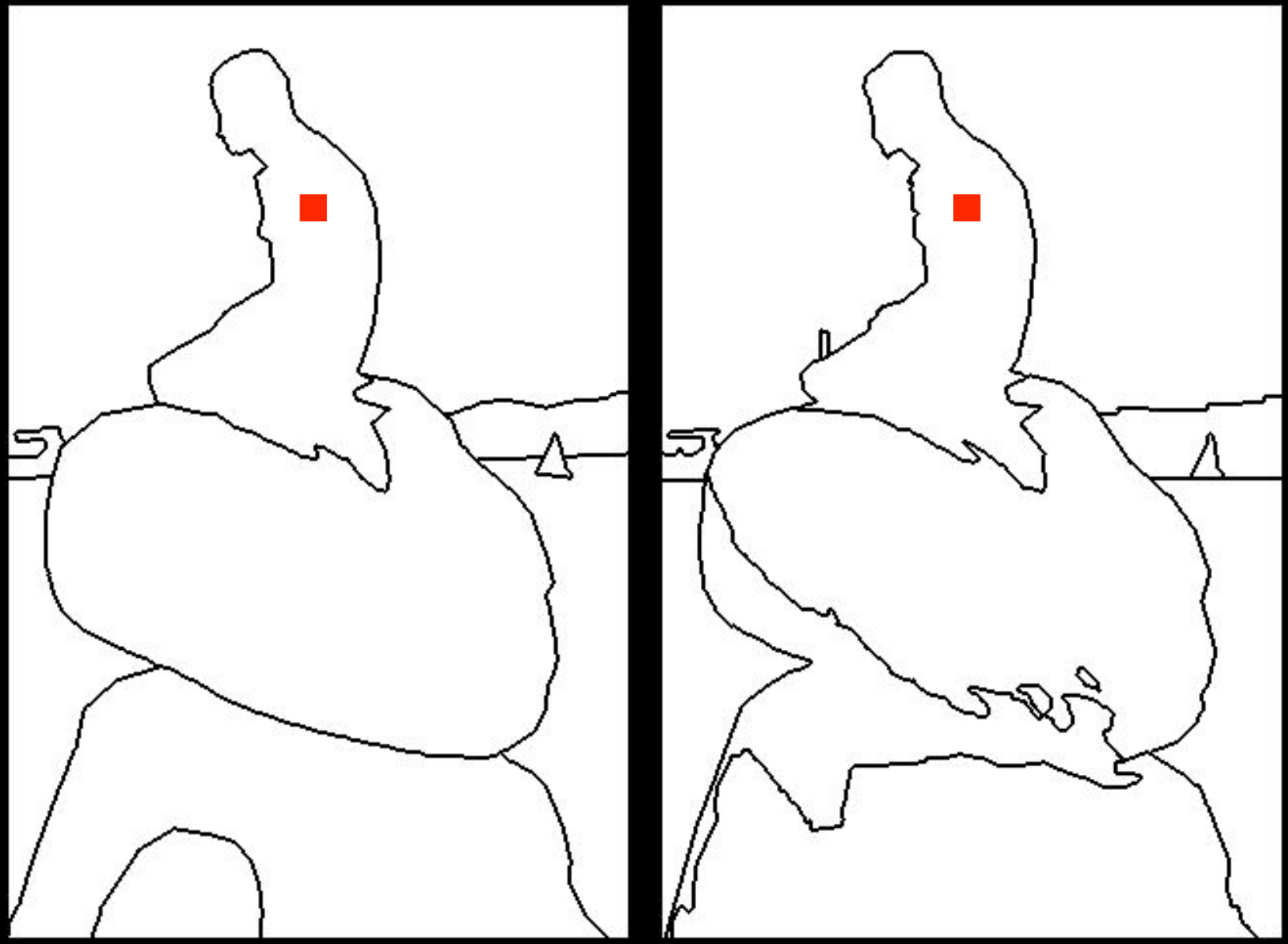

B
Scene
Background
Foreground
Sky
Land
Water
Rocks
Mermaid
Trees
Shore
Small
Top
Base
L
R


A
B
C
D

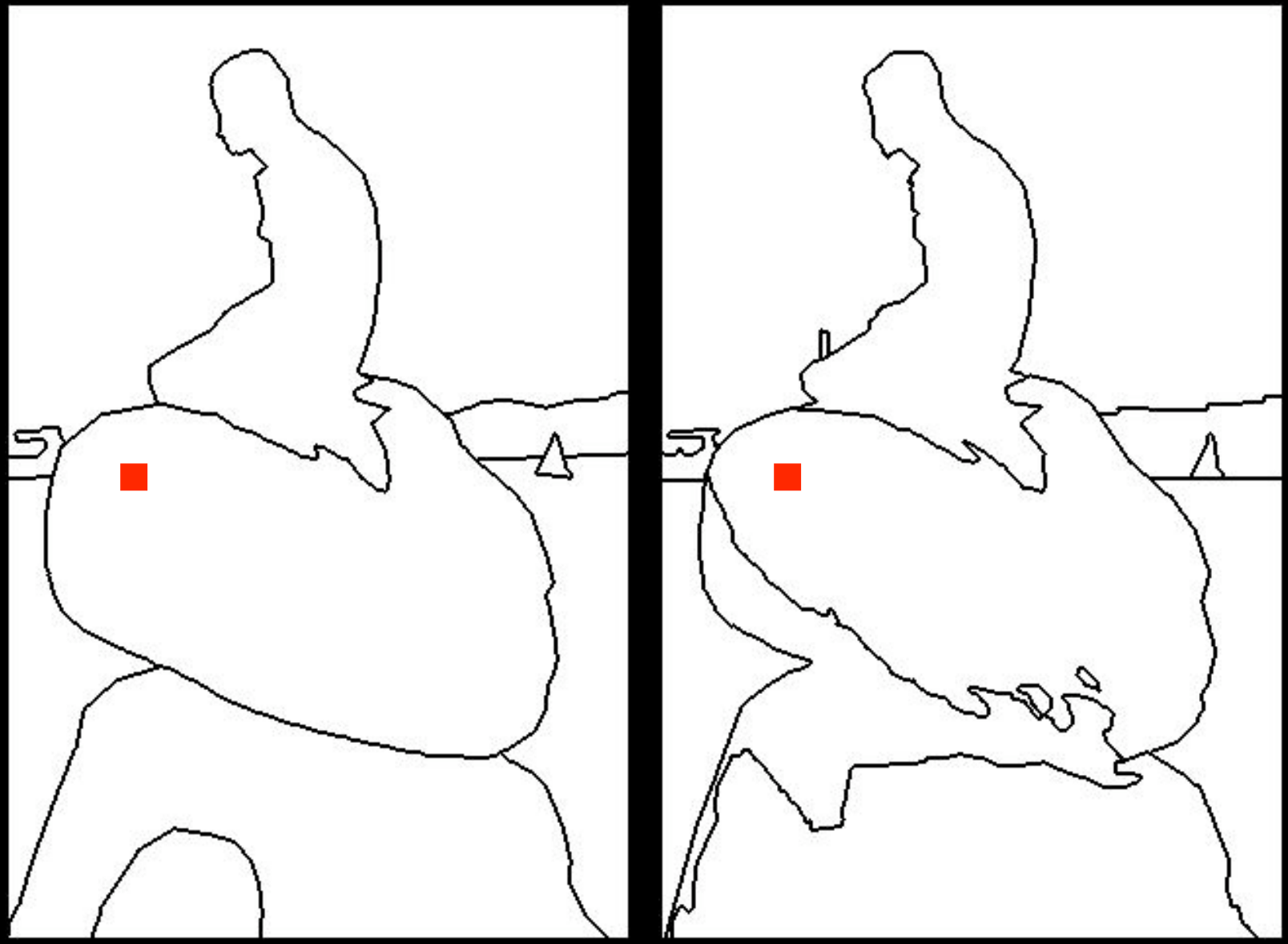
Step #3: Measuring Error

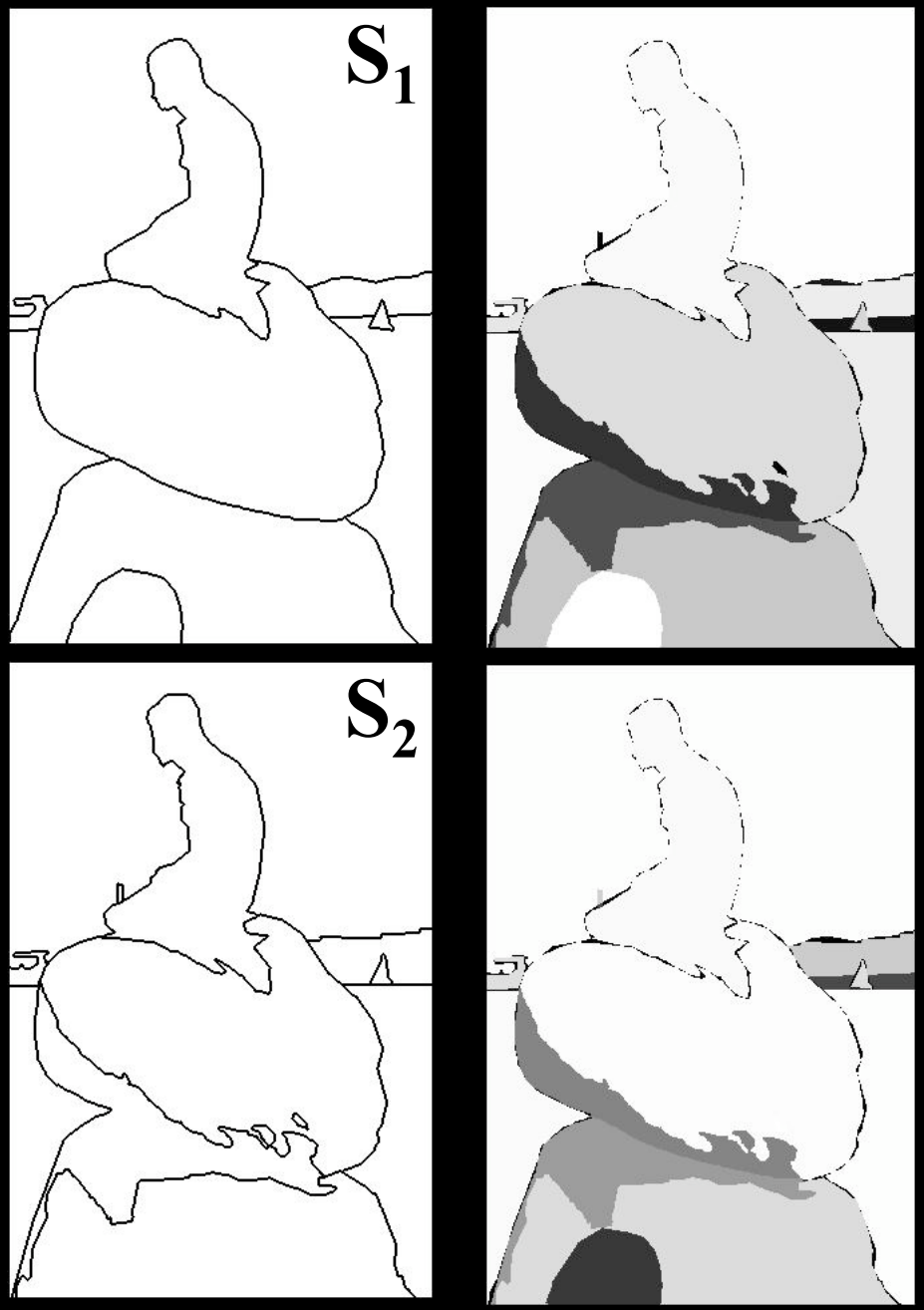
Approach: Region overlap

$$E(S_1, S_2, p_i) = |R(S_1, p_i) \setminus R(S_2, p_i)| \;/\; |R(S_1, p_i)|$$



$$E(S_1, S_2, p_i) = |R(S_1, p_i) \setminus R(S_2, p_i)| \;/\; |R(S_1, p_i)|$$

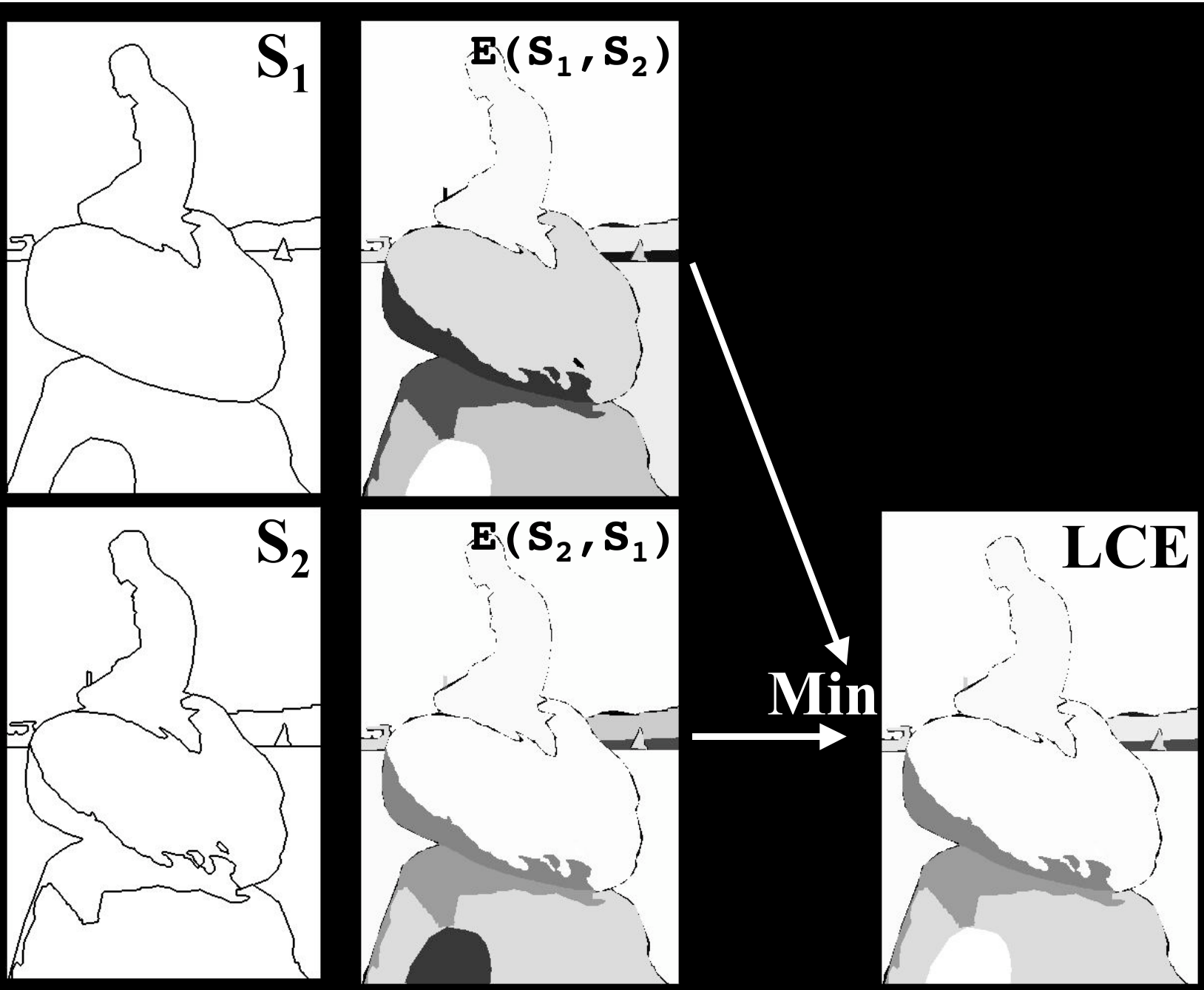




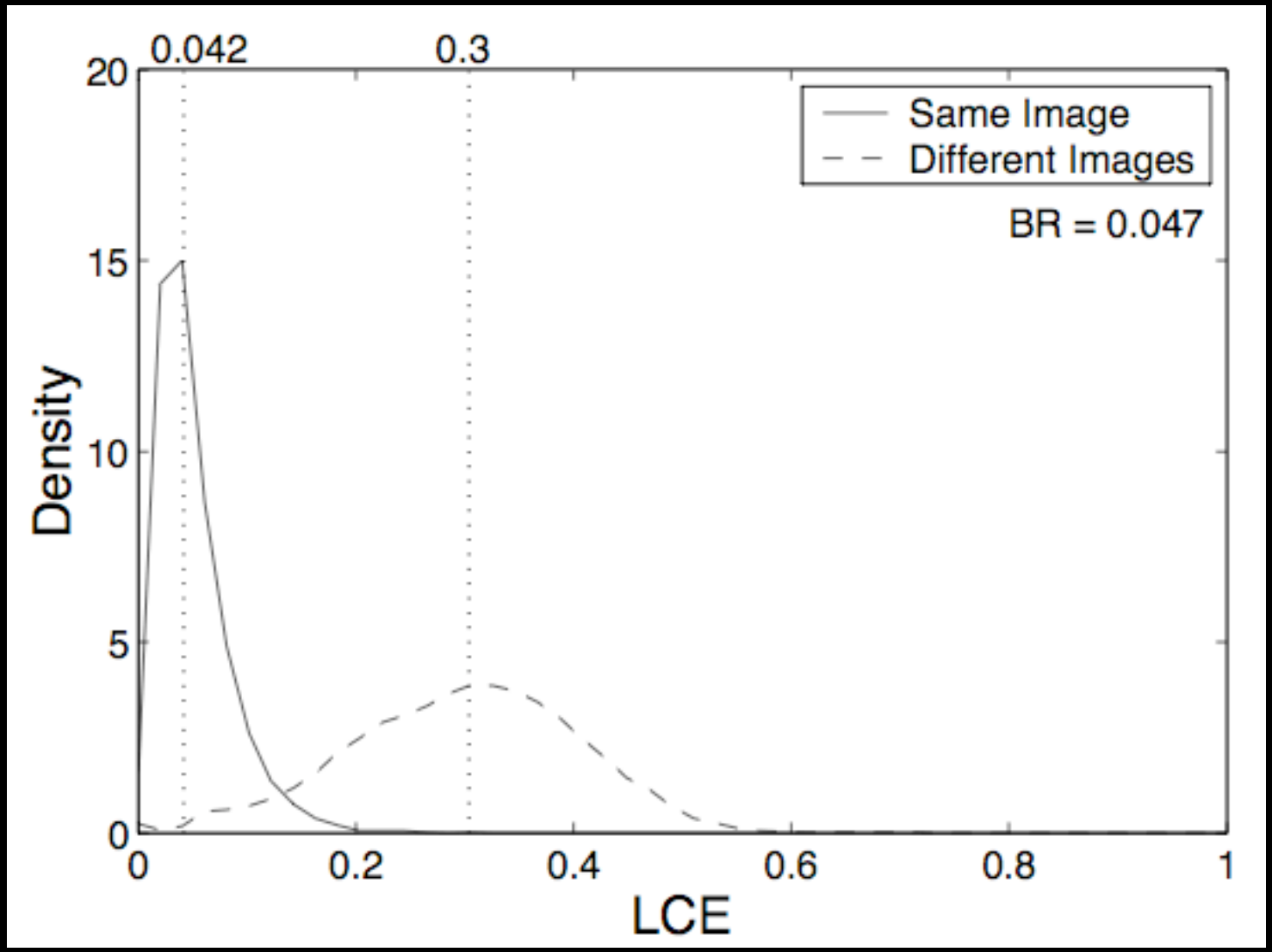
$E(S_1, S_2, p_i)$

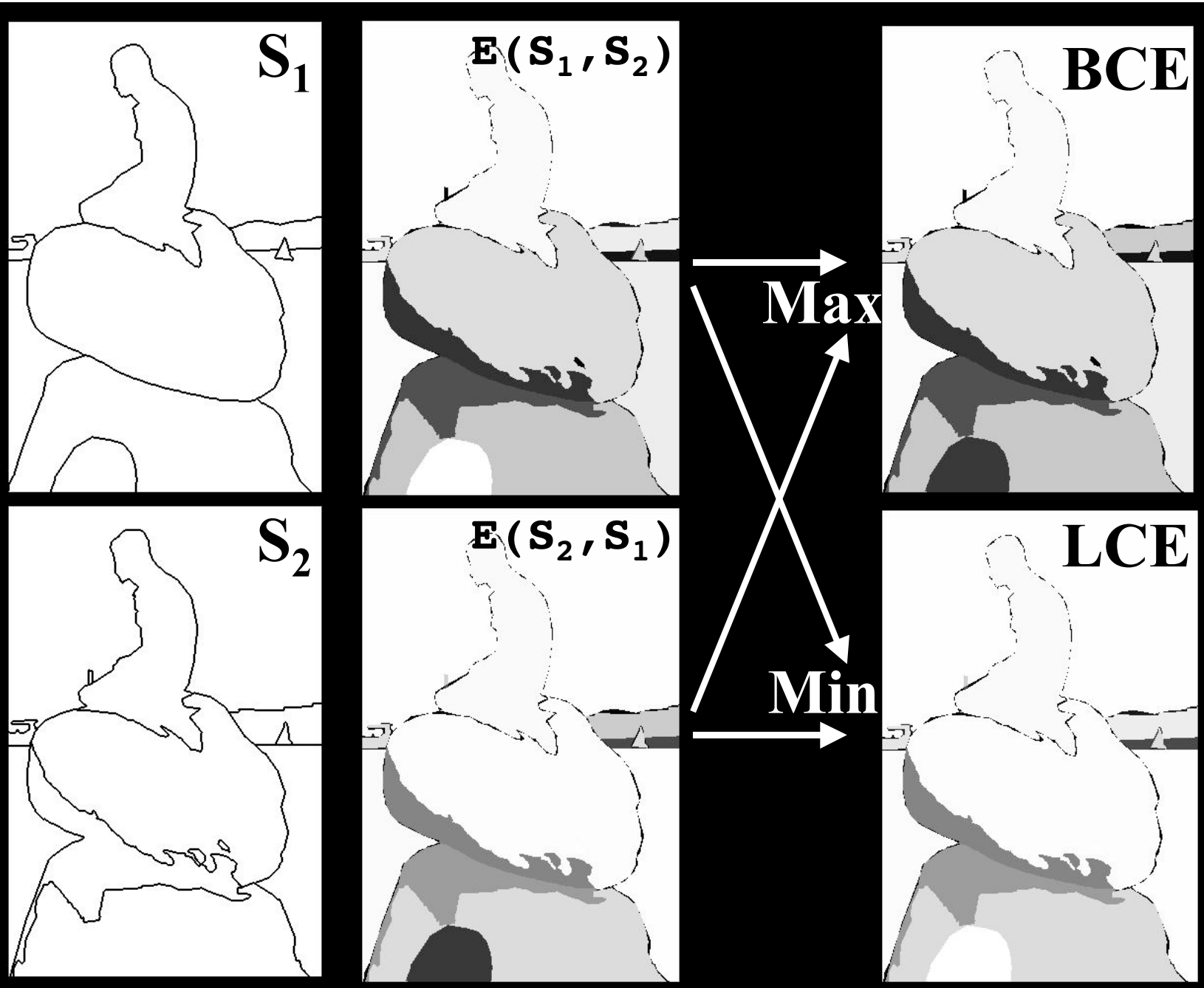
Local Refinement Error

$E(S_2, S_1, p_i)$


S_1
$E(S_1, S_2)$
S_2
$E(S_2, S_1)$
Min
LCE

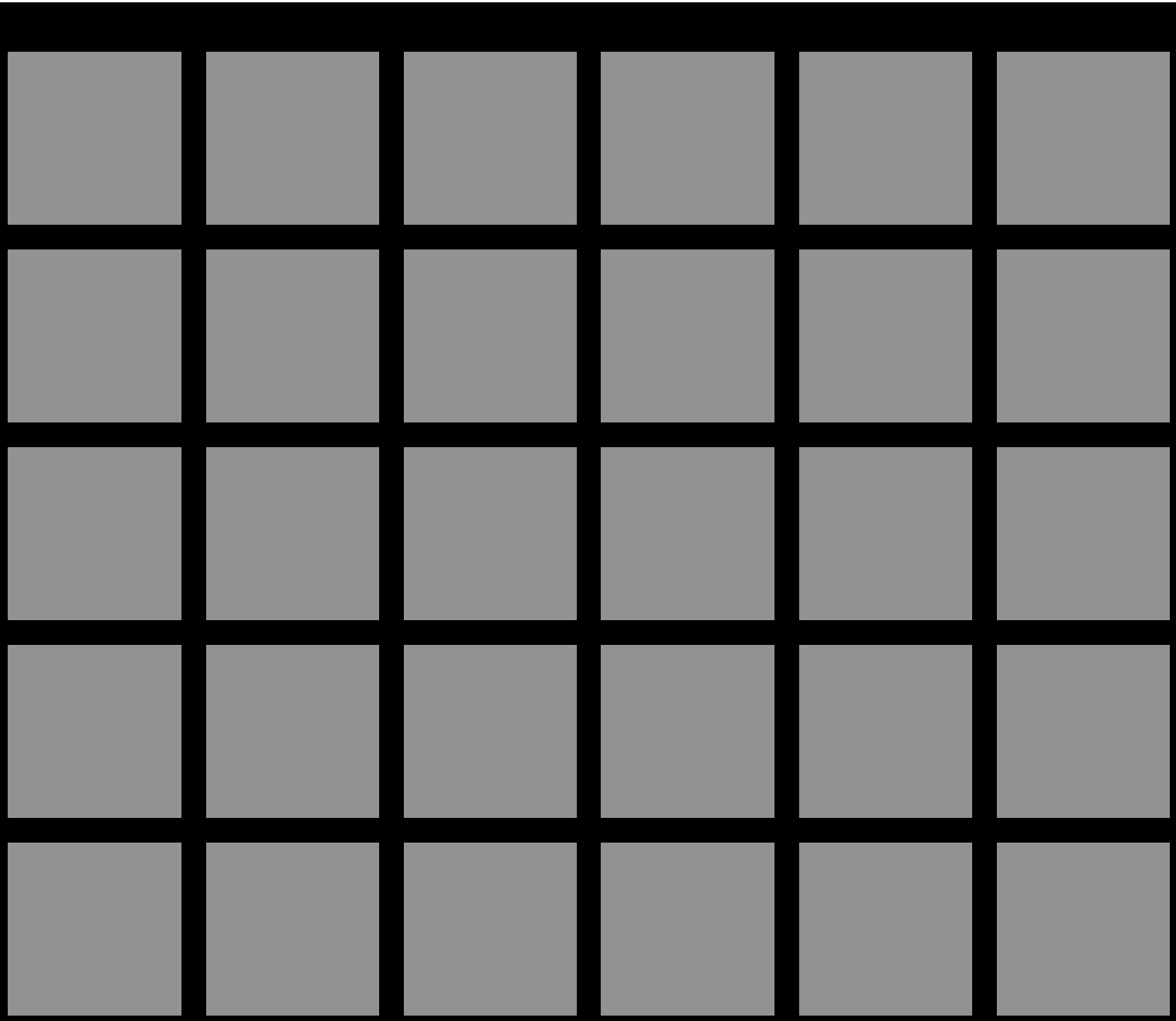
LCE Between Pairs of Human Segmentations

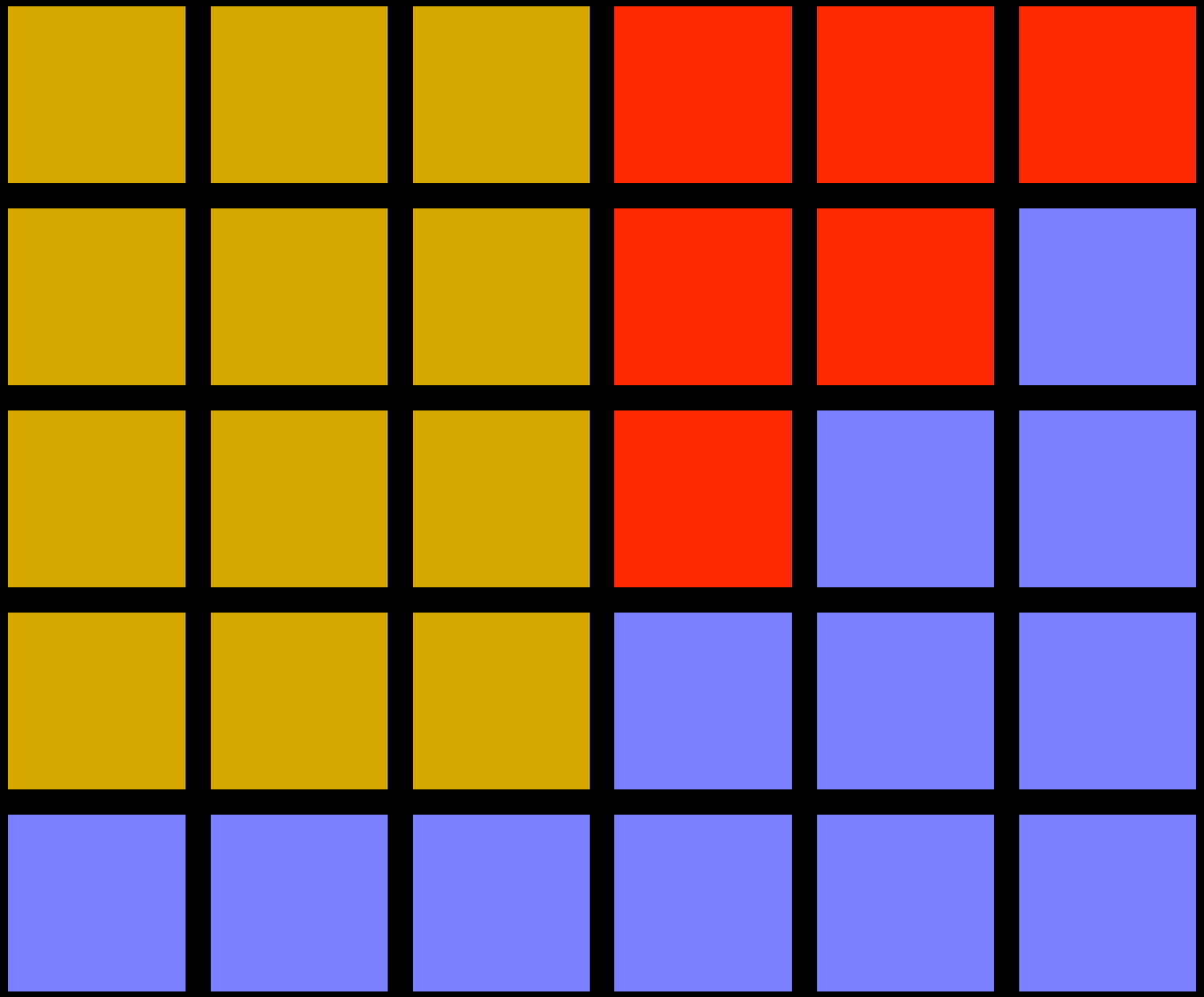


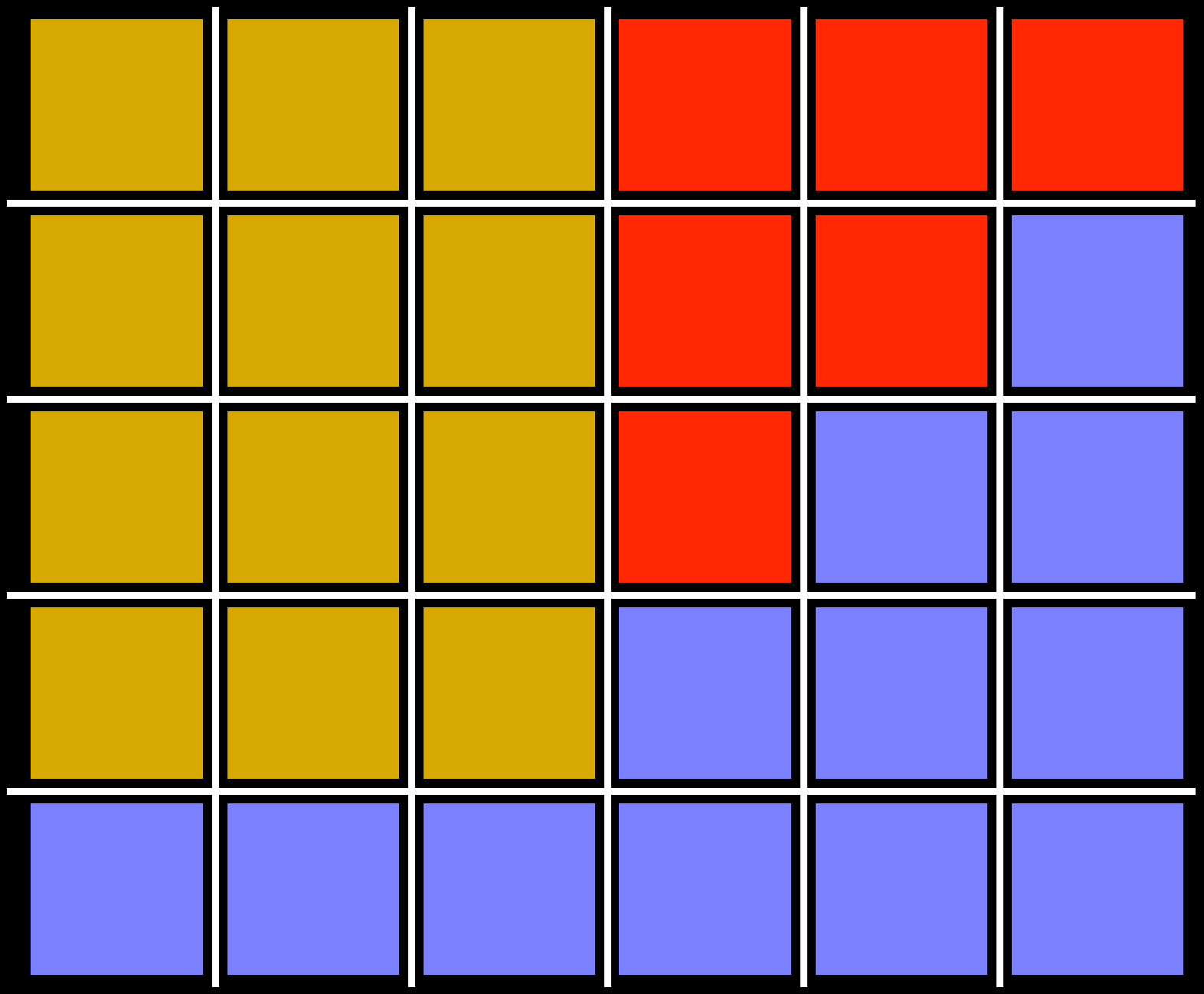

S_1
$E(S_1, S_2)$
BCE
Max
S_2
$E(S_2, S_1)$
LCE
Min

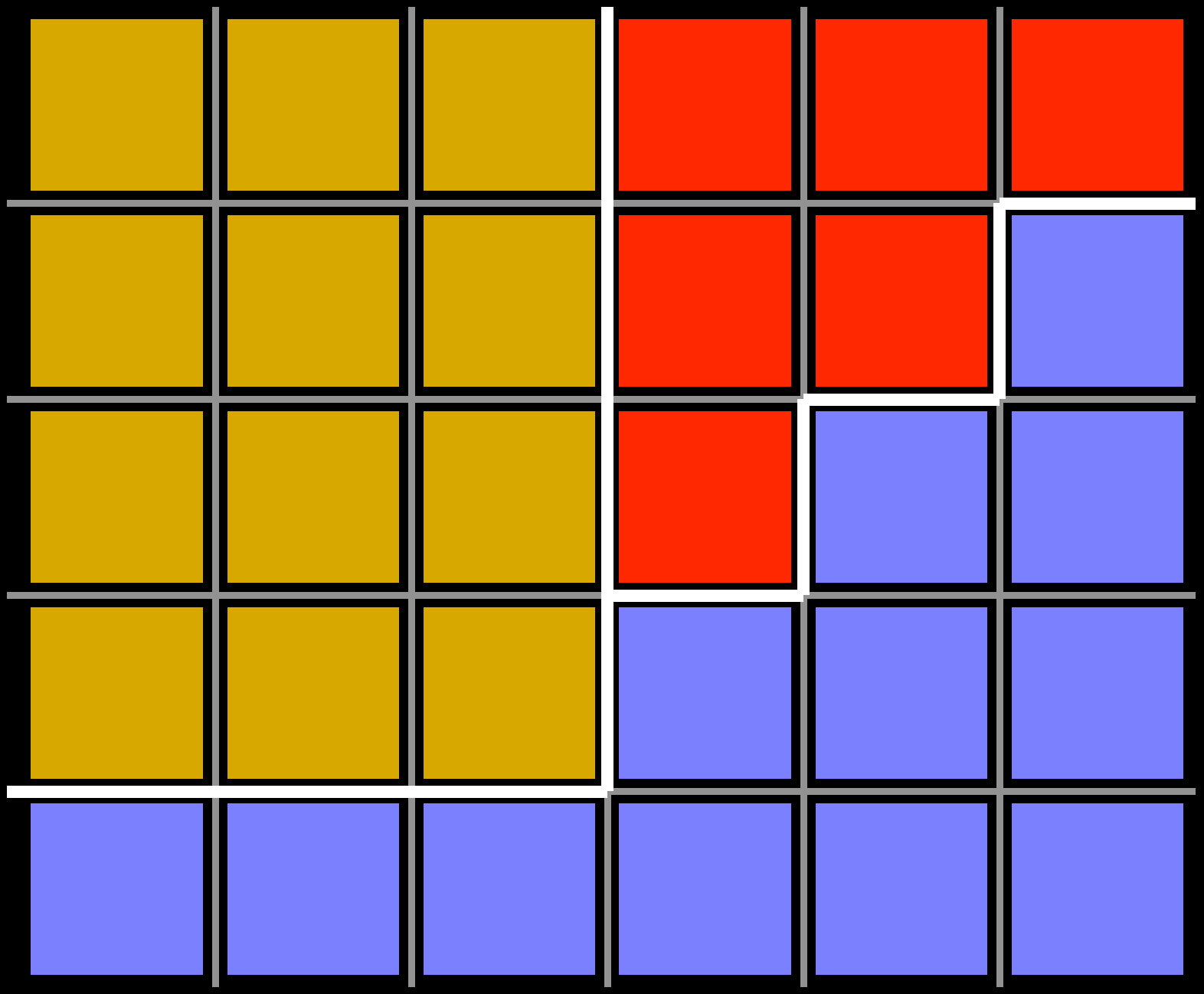
Step #3 (again): Measuring Error

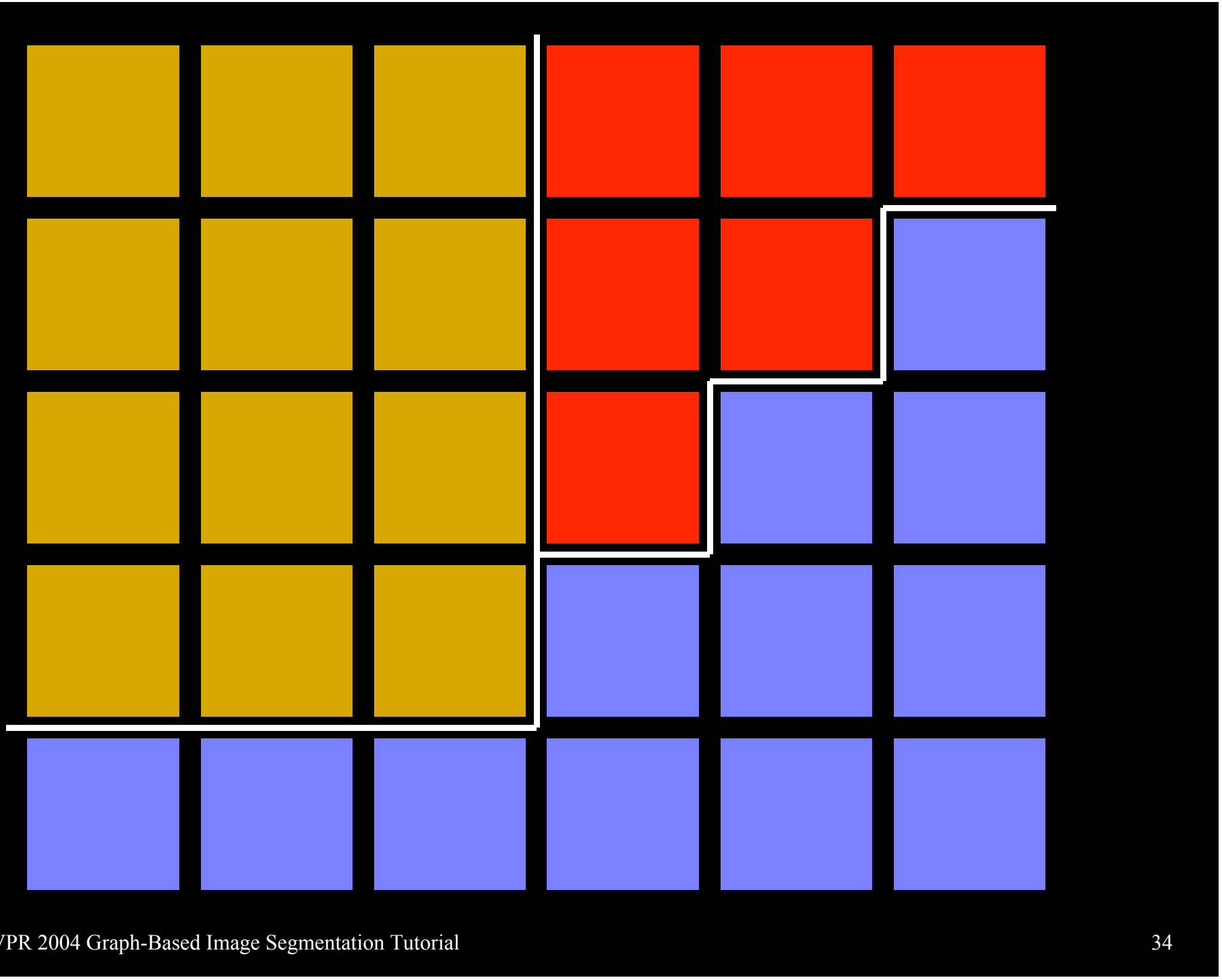
Approach: Edgel matching

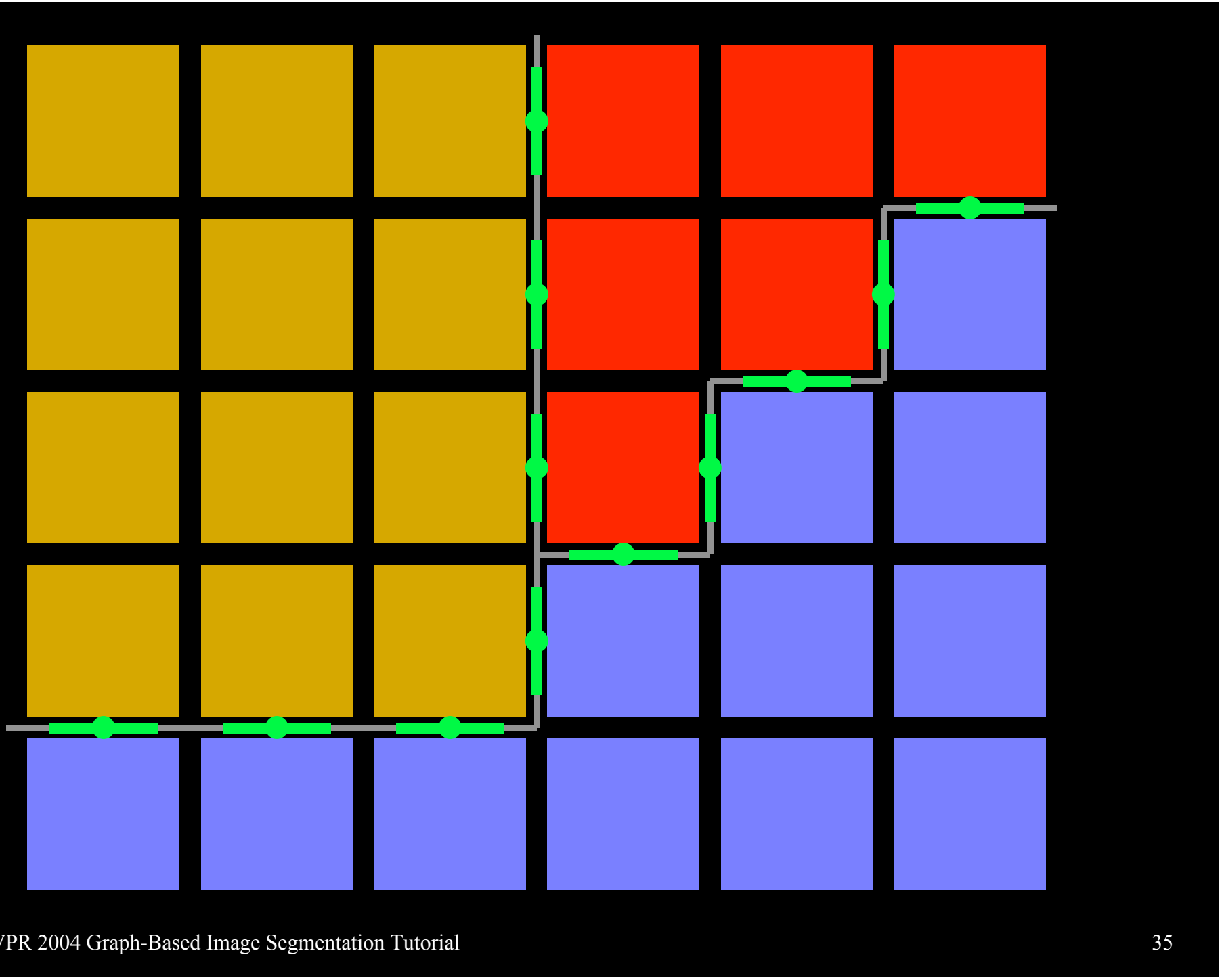


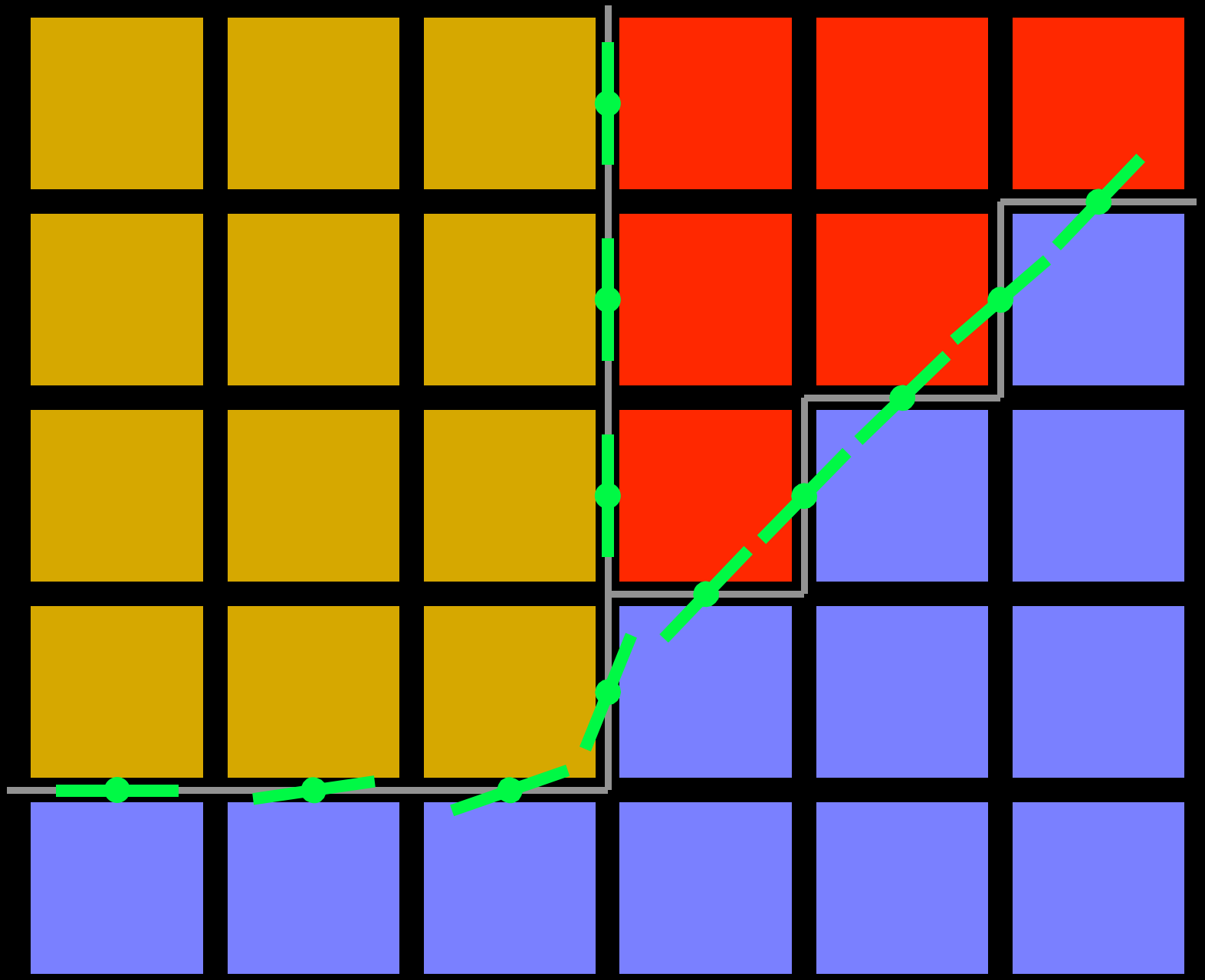


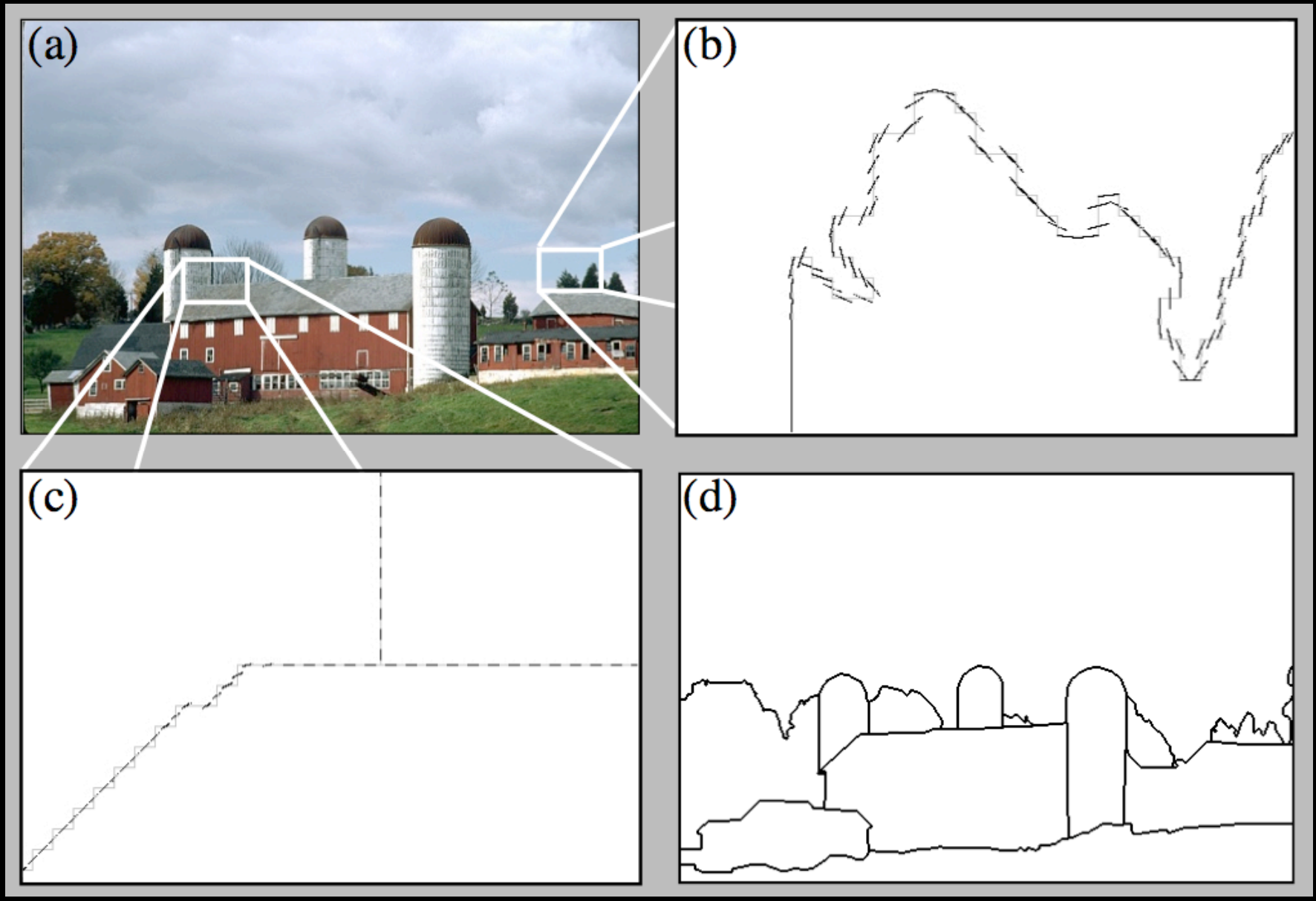


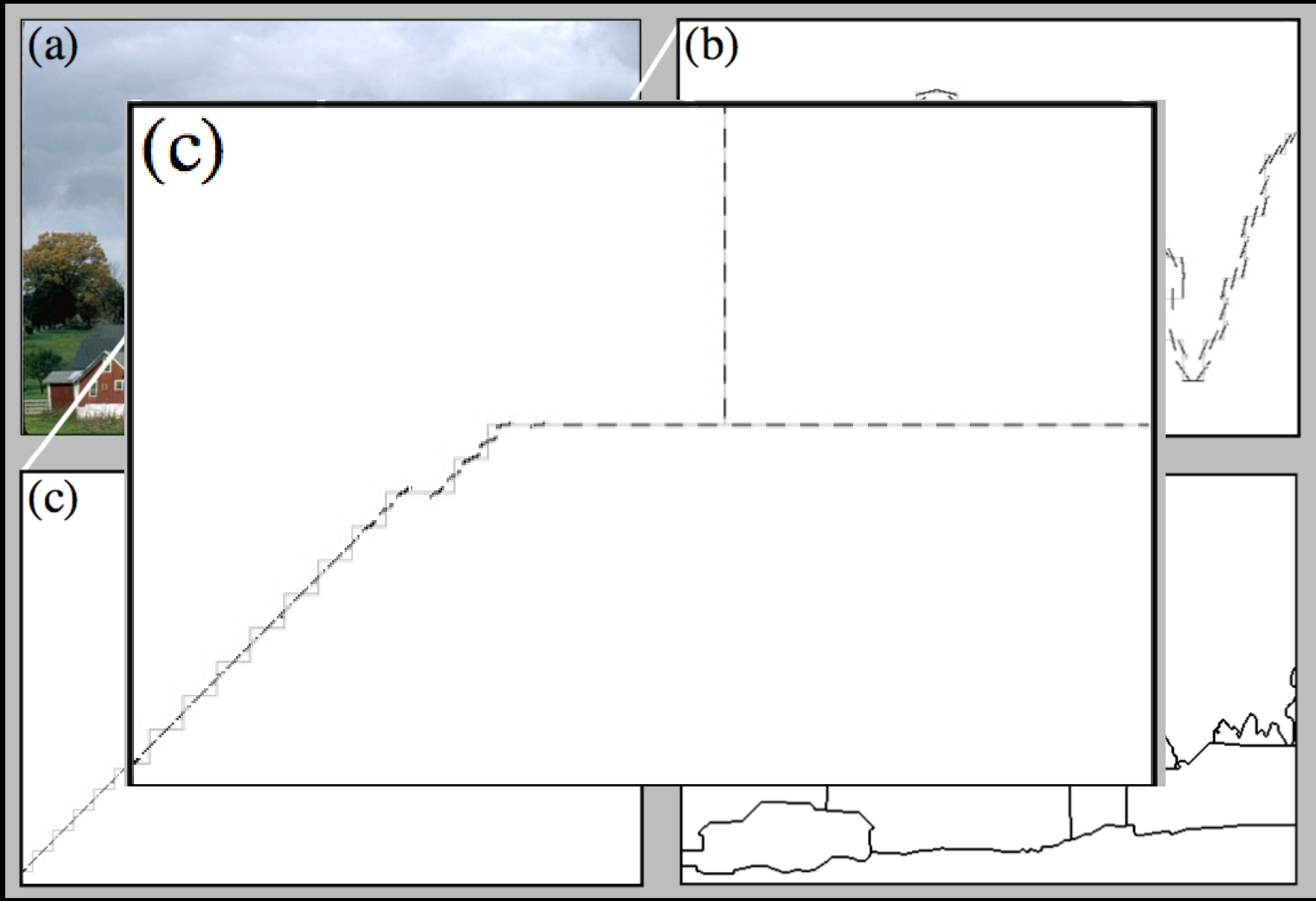


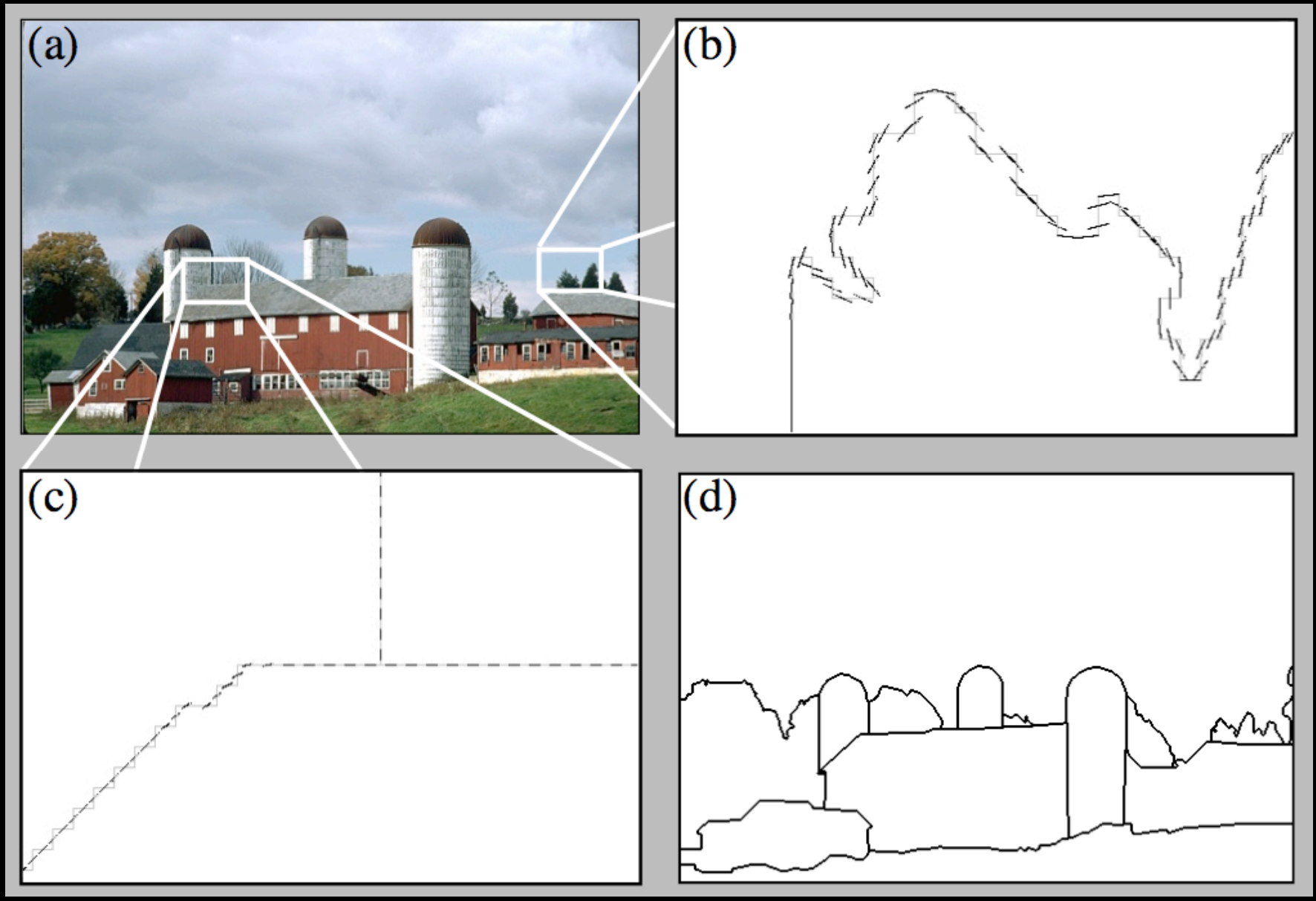


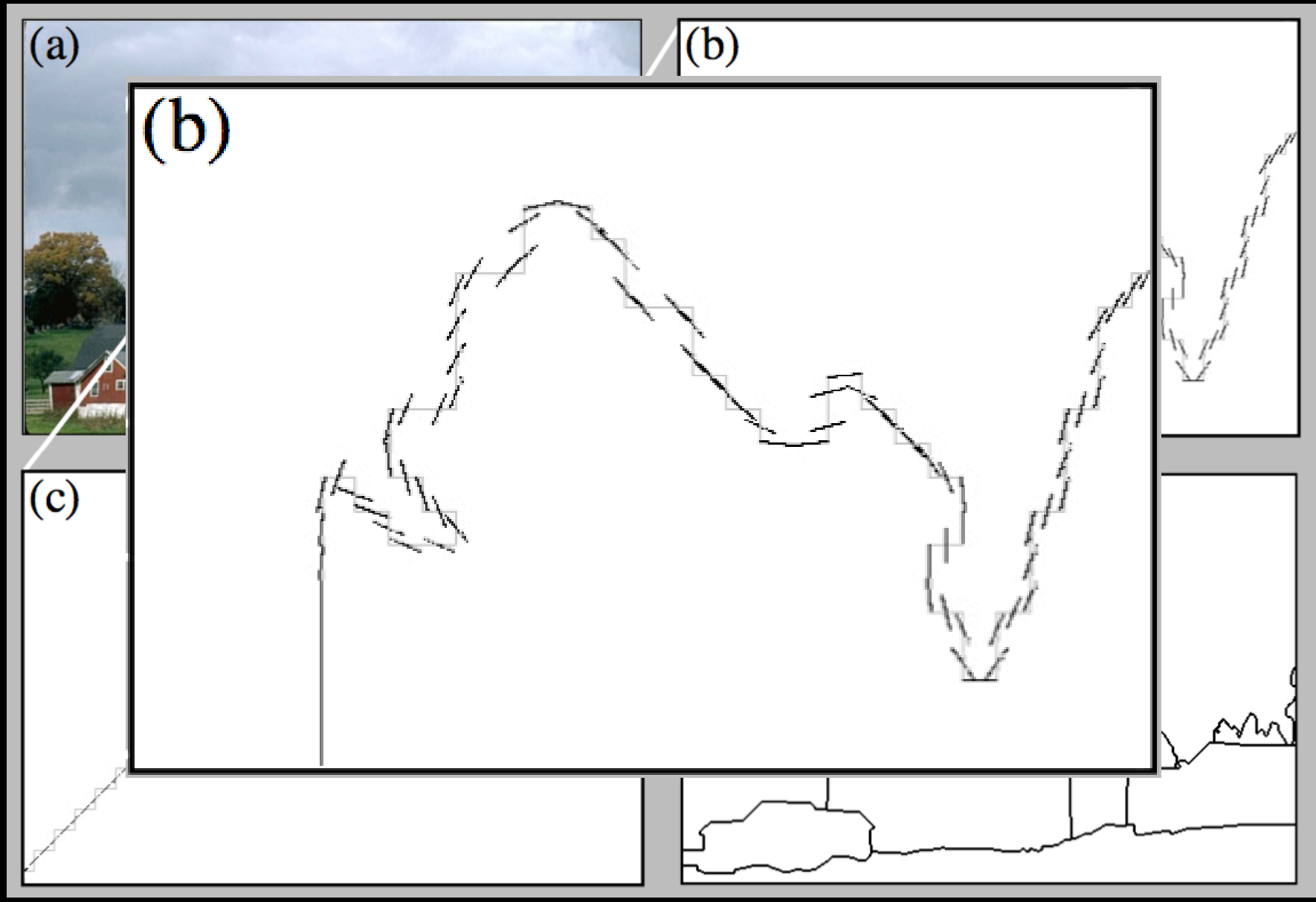





(a)
(b)
(c)
(d)

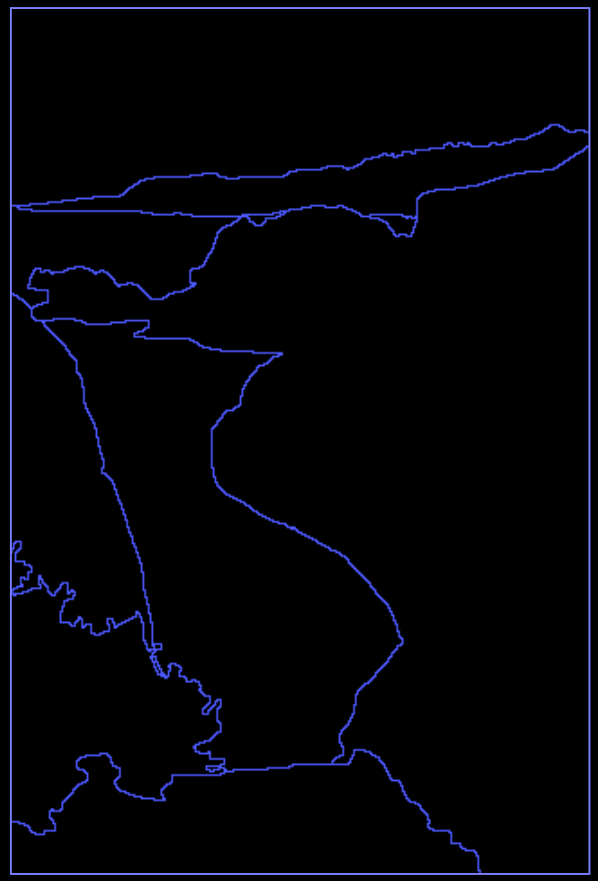

(a)
(b)
(c)
(c)


(a)
(b)
(c)
(d)

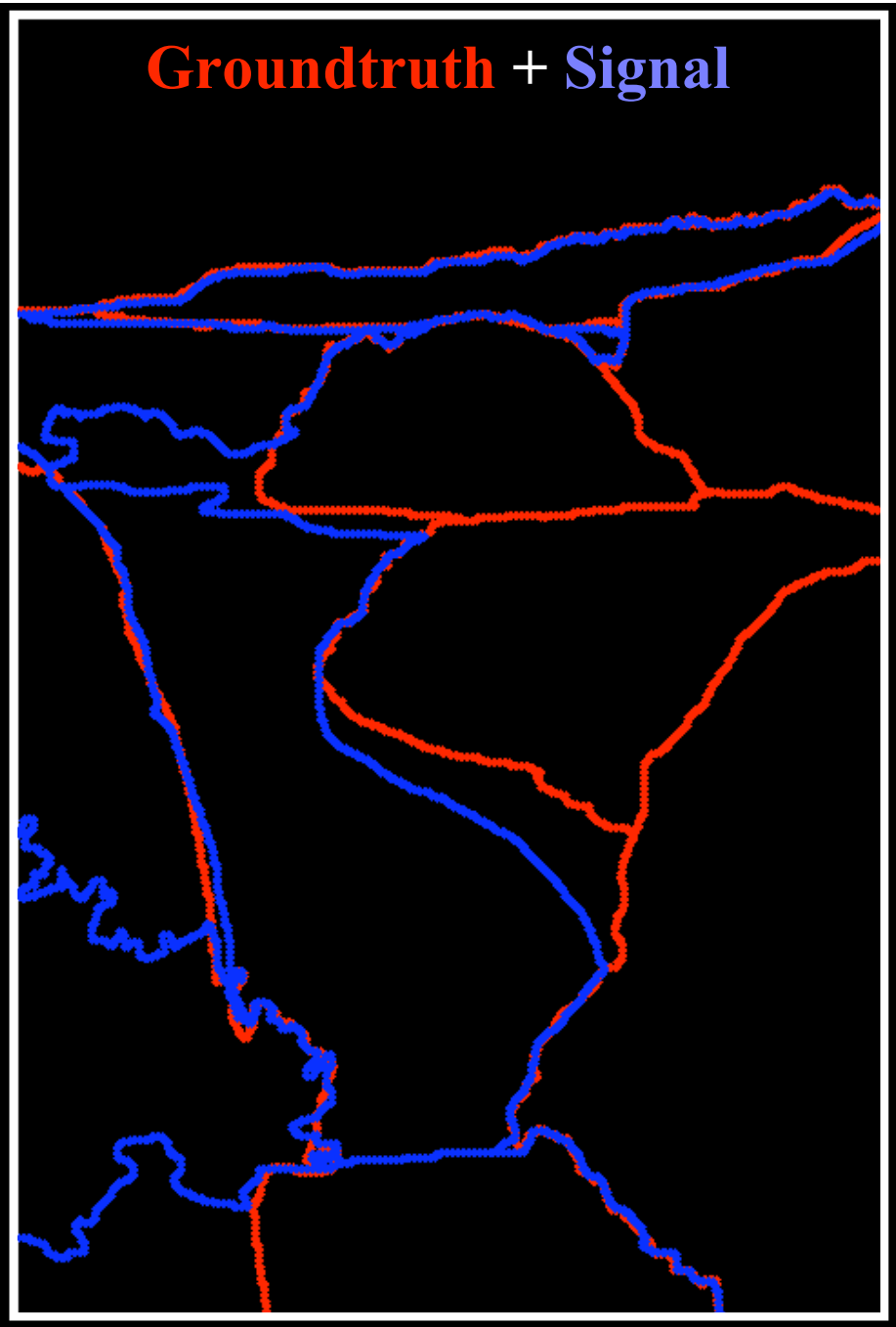

(a)
(b)
(b)
(c)

Groundtruth

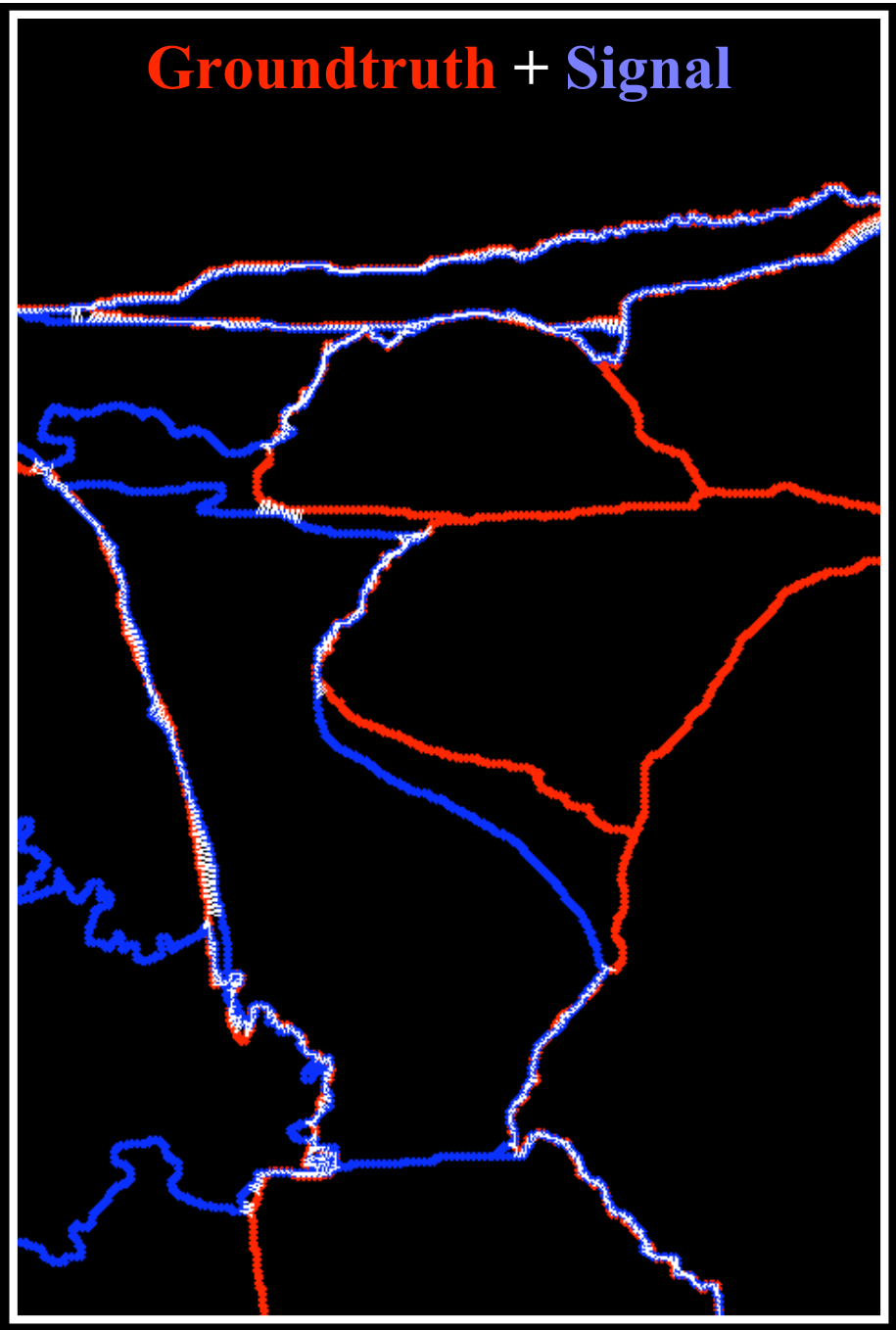
Signal



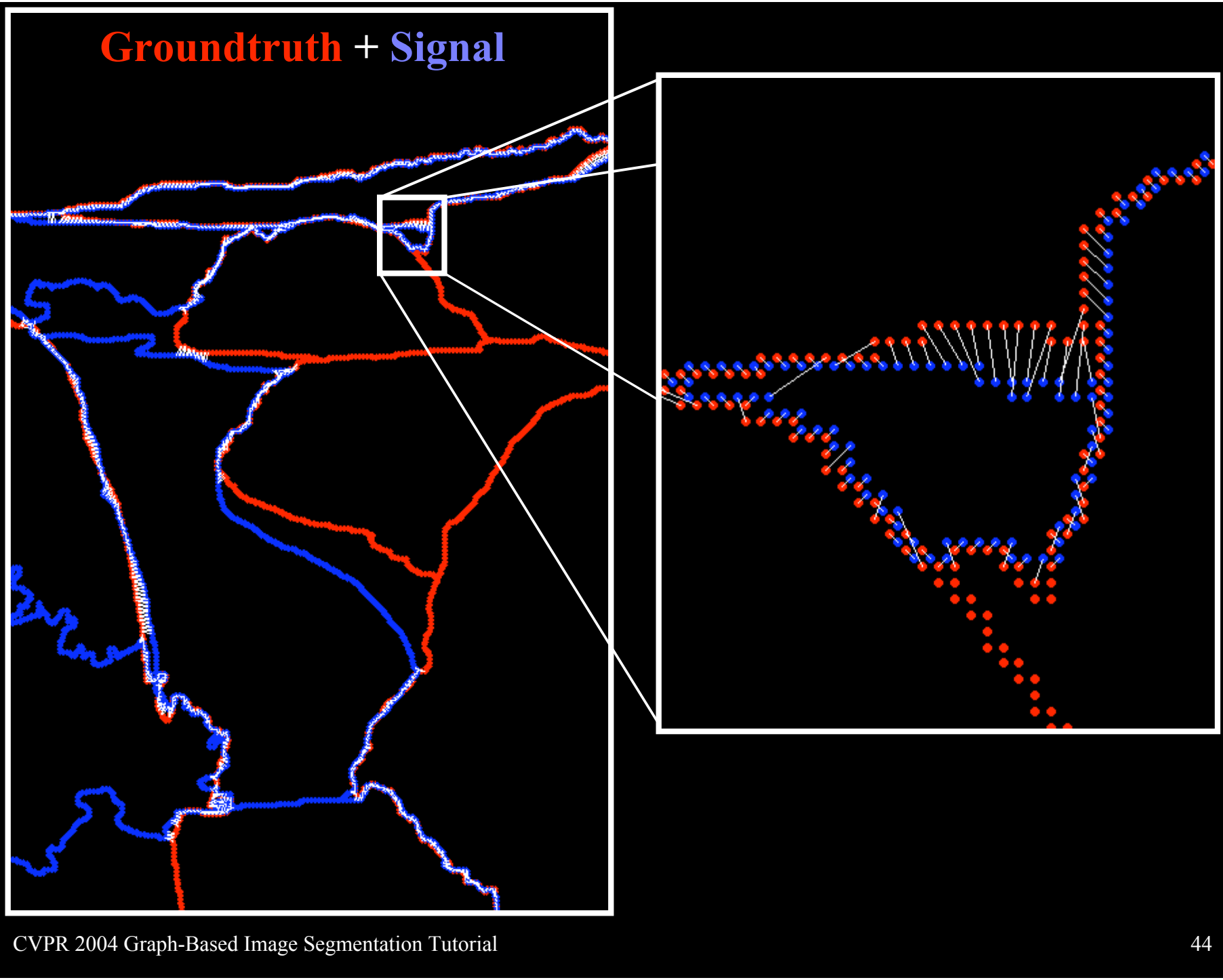
How do we correspond boundary elements?

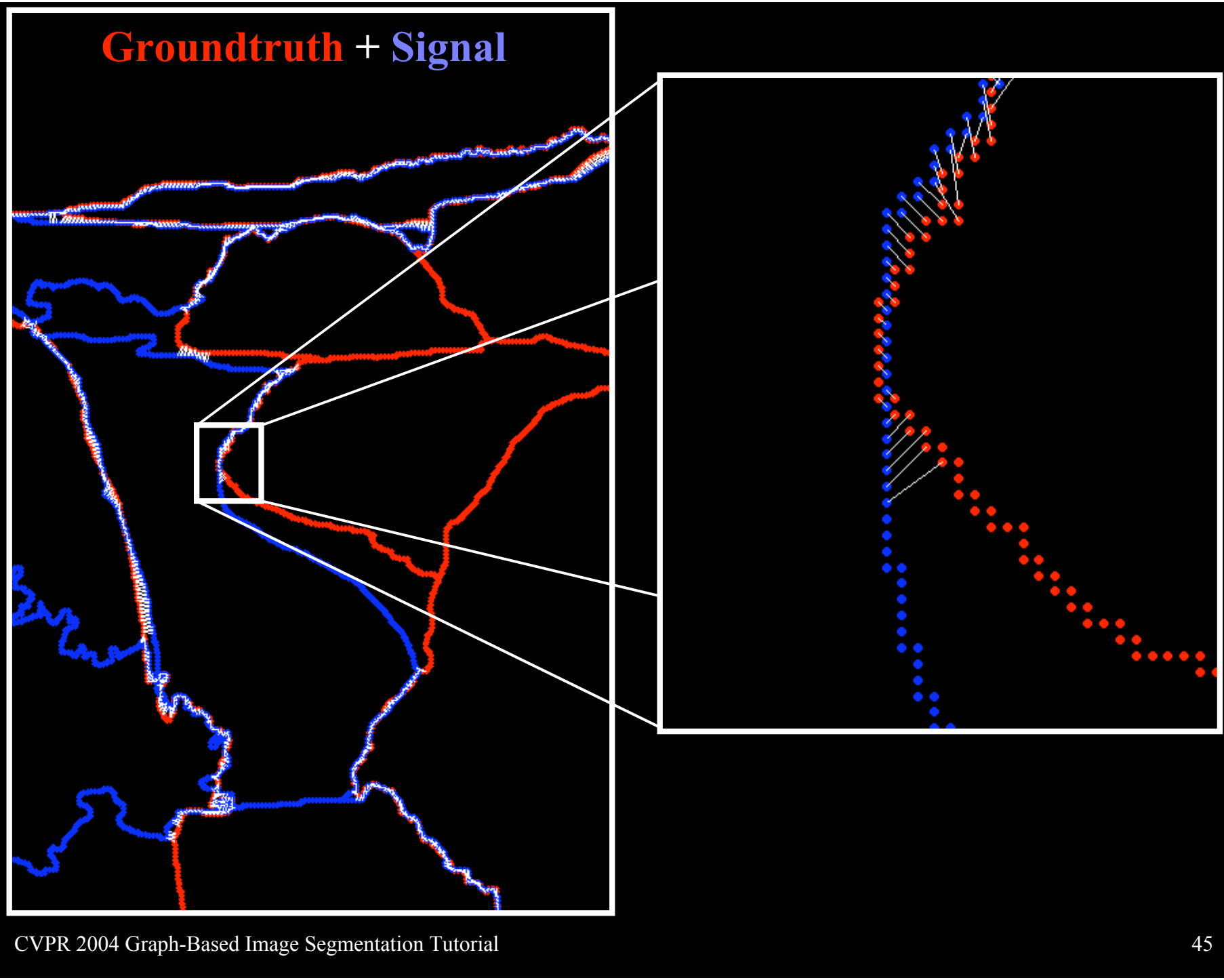


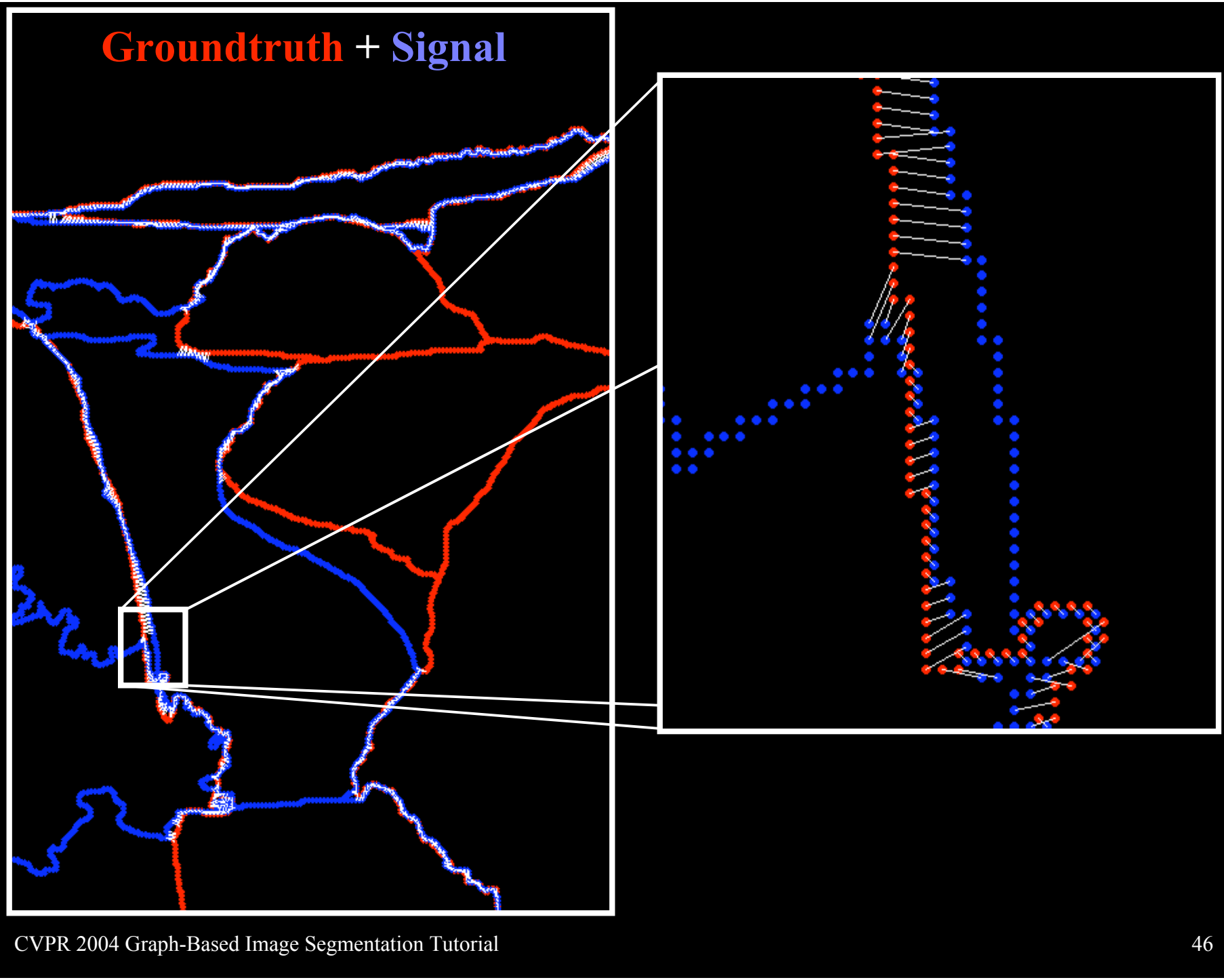
Simply overlaying does not work…



Compute correspondence via min-cost assignment on bipartite graph.


Groundtruth + Signal


Groundtruth + Signal


Groundtruth + Signal

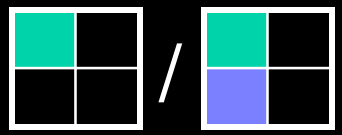
ROC vs. Precision/Recall

$$F = 2/(R^{-1}+P^{-1}) = 2PR/(P+R)$$

Signal \ Truth	P	N
P	TP	FP
N	FN	TN

PR Curve

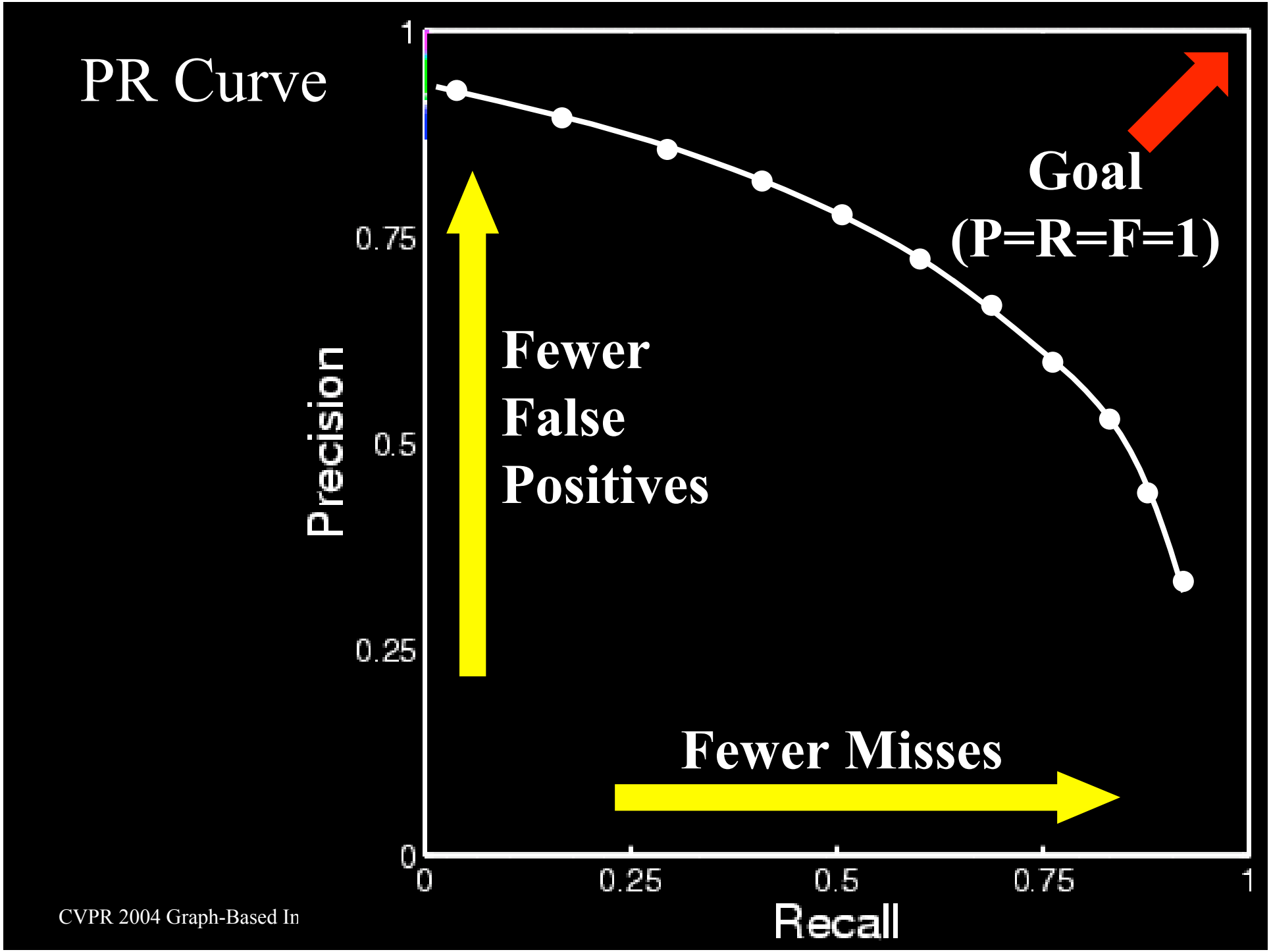
Precision = TP / (TP+FP) =  **= Specificity**

Recall = TP / (TP+FN) =  **= Sensitivity**

ROC Curve

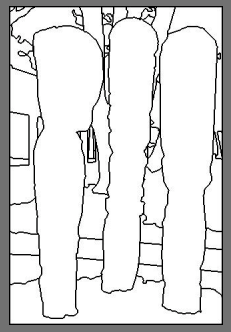
Hit Rate = TP / (TP+FN) = 

False Alarm Rate = FP / (FP+TN) = 

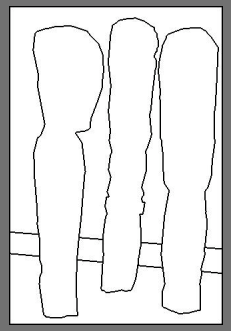

PR Curve
Goal
(P=R=F=1)
Fewer
False
Positives
Fewer Misses
Precision
Recall
1
0.75
0.5
0.25
0
0
0.25
0.5
0.75
1

But how do we compute hits, misses, and false positives?

S_1

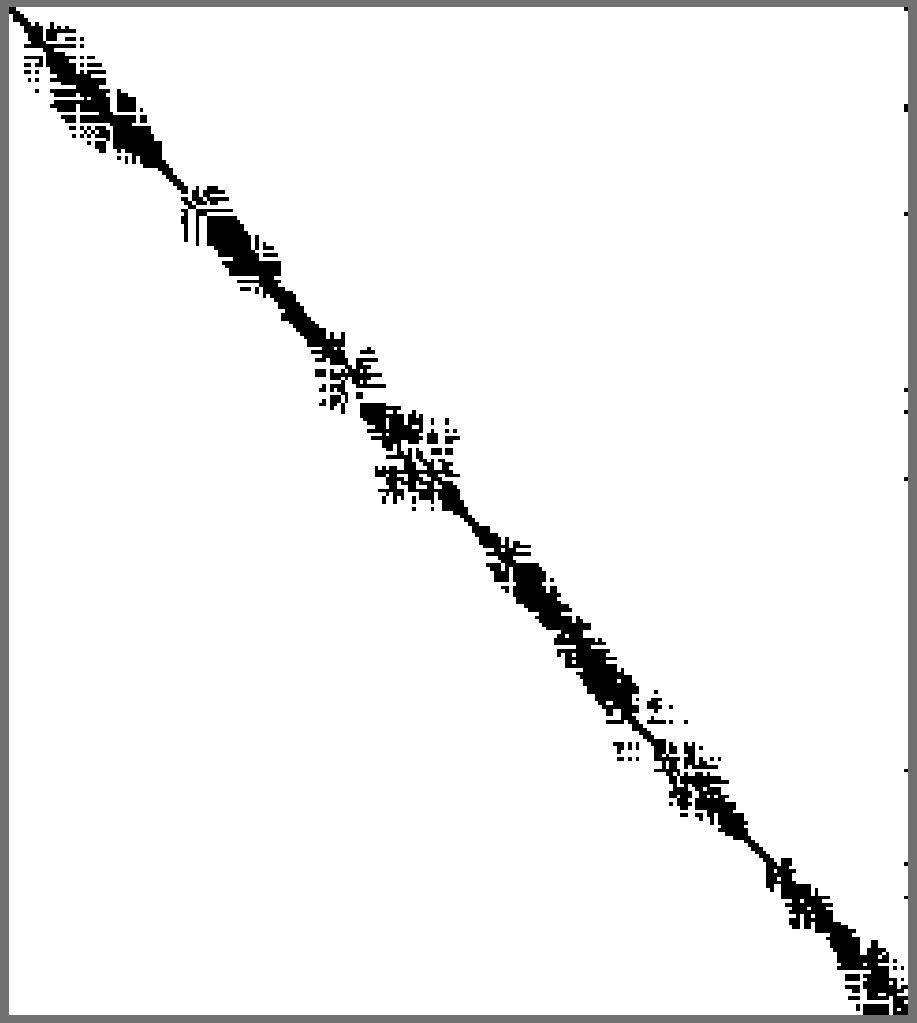


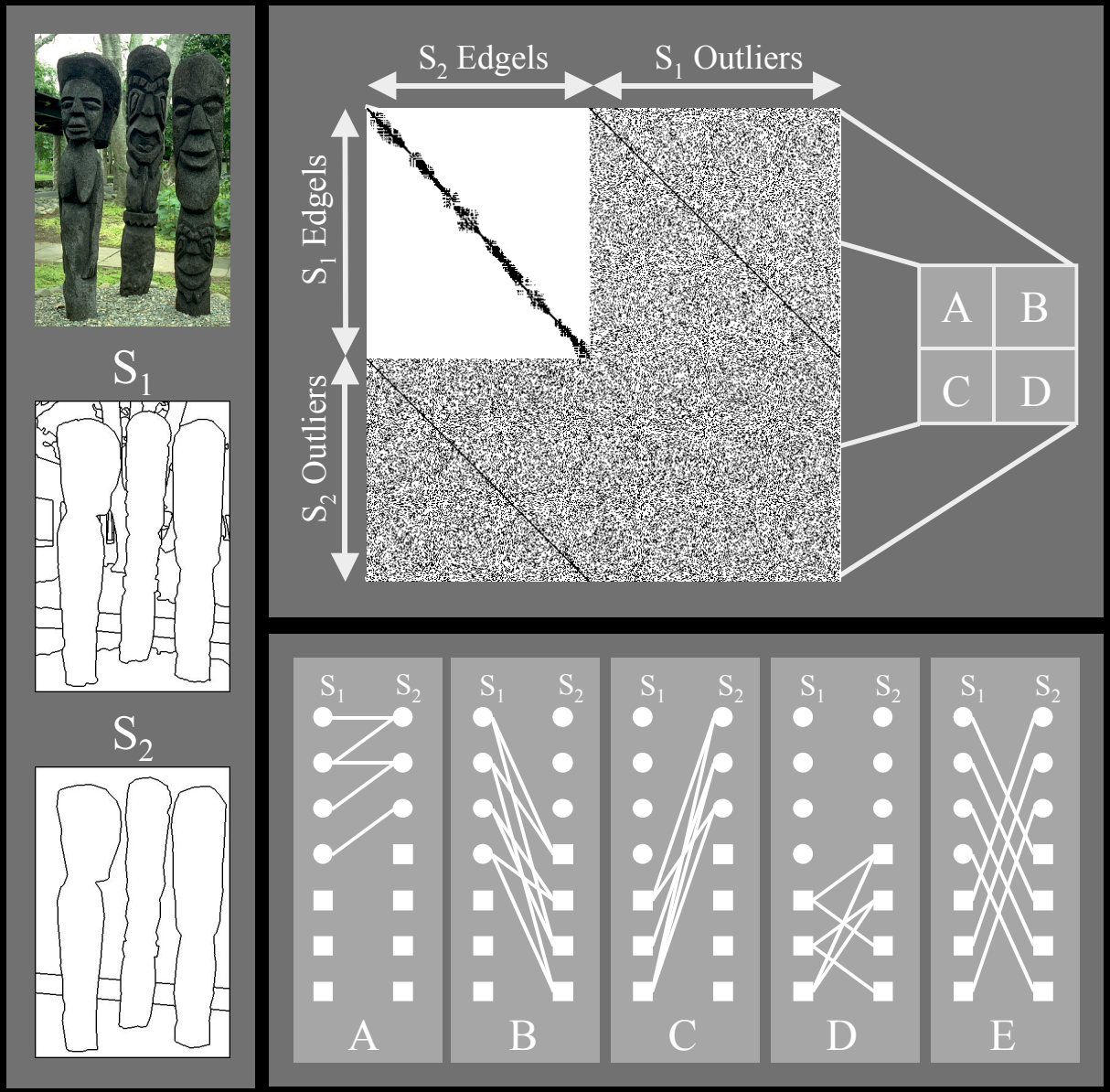
S_2



S_2 Edgels

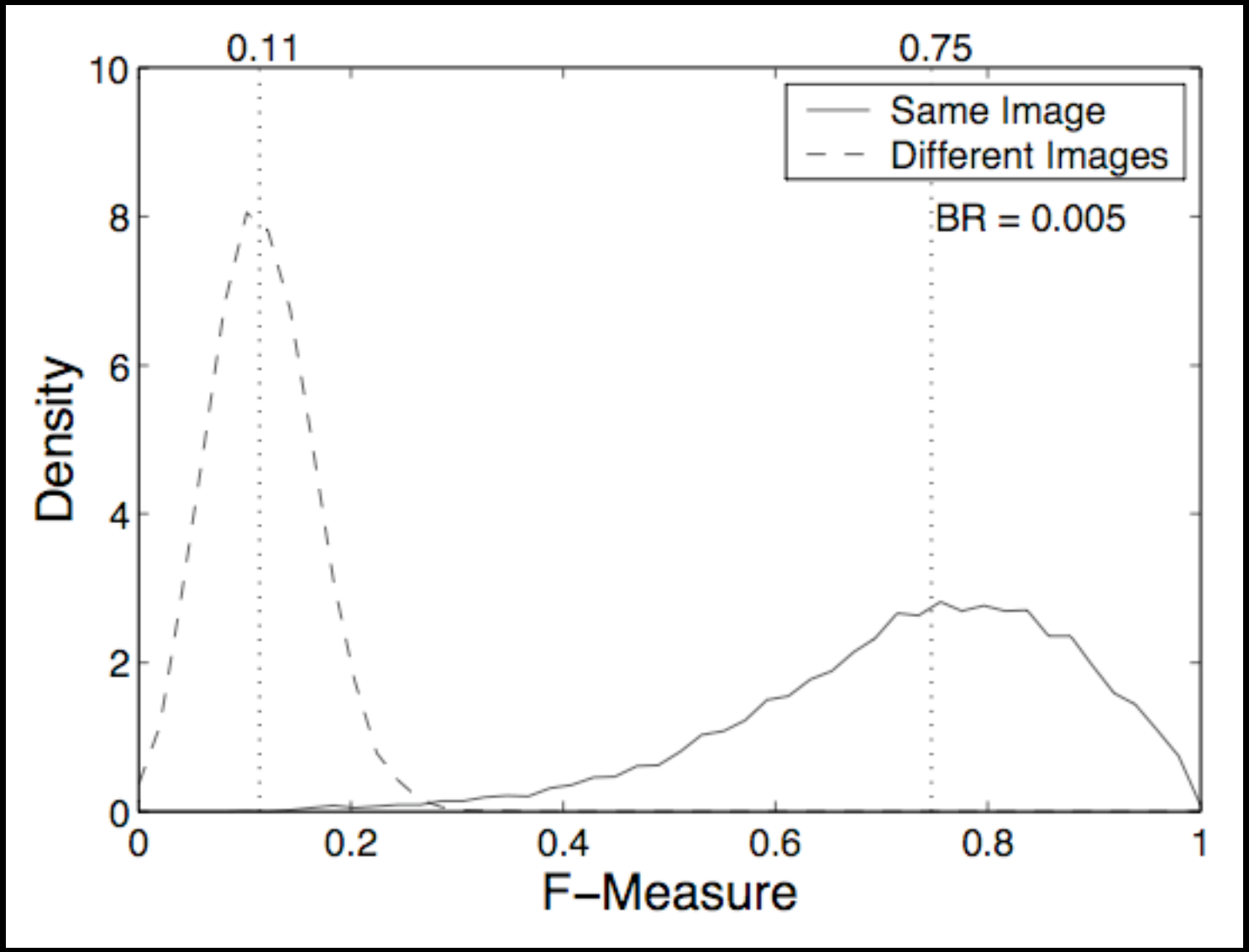
S_1 Edgels




S1
S2
S2 Edgels
S1 Outliers
S1 Edgels
S2 Outliers
A
B
C
D
S1
S2
A
B
C
D
E

Fast min-cost assignment for sparse graphs?

- Andrew Goldberg's CSA package for max-flow / min-cut graph problems
- Sparse graphs
- Linear time (!)
- C code, but I have C++ / MATLAB wrappers

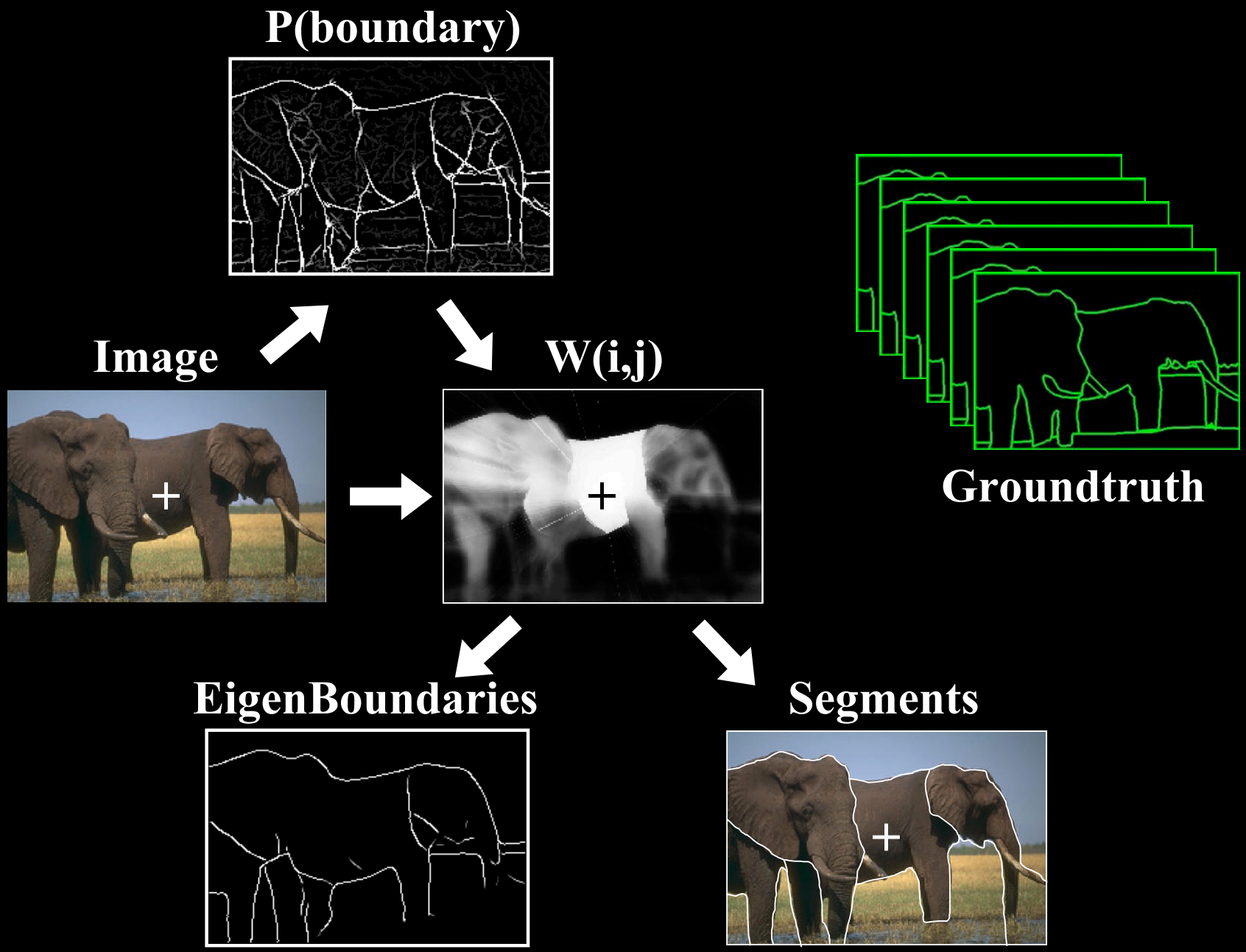

0.11
0.75
10
8
6
4
2
0
Density
Same Image
Different Images
BR = 0.005
0
0.2
0.4
0.6
0.8
1
F–Measure

Advantages of edgel matching over region overlap

- Far more discriminative
  - More sensitive to boundary complexity
  - Does not match offset interiors
- Intuitive measures and units
  - Detection-oriented framework
- Both a low-level and mid-level benchmark
  - Sensible for evaluating both edges and regions
  - **Leverage data collection / groundtruthing cost**
  - **We can quantify improvement between levels**

One last error measure…

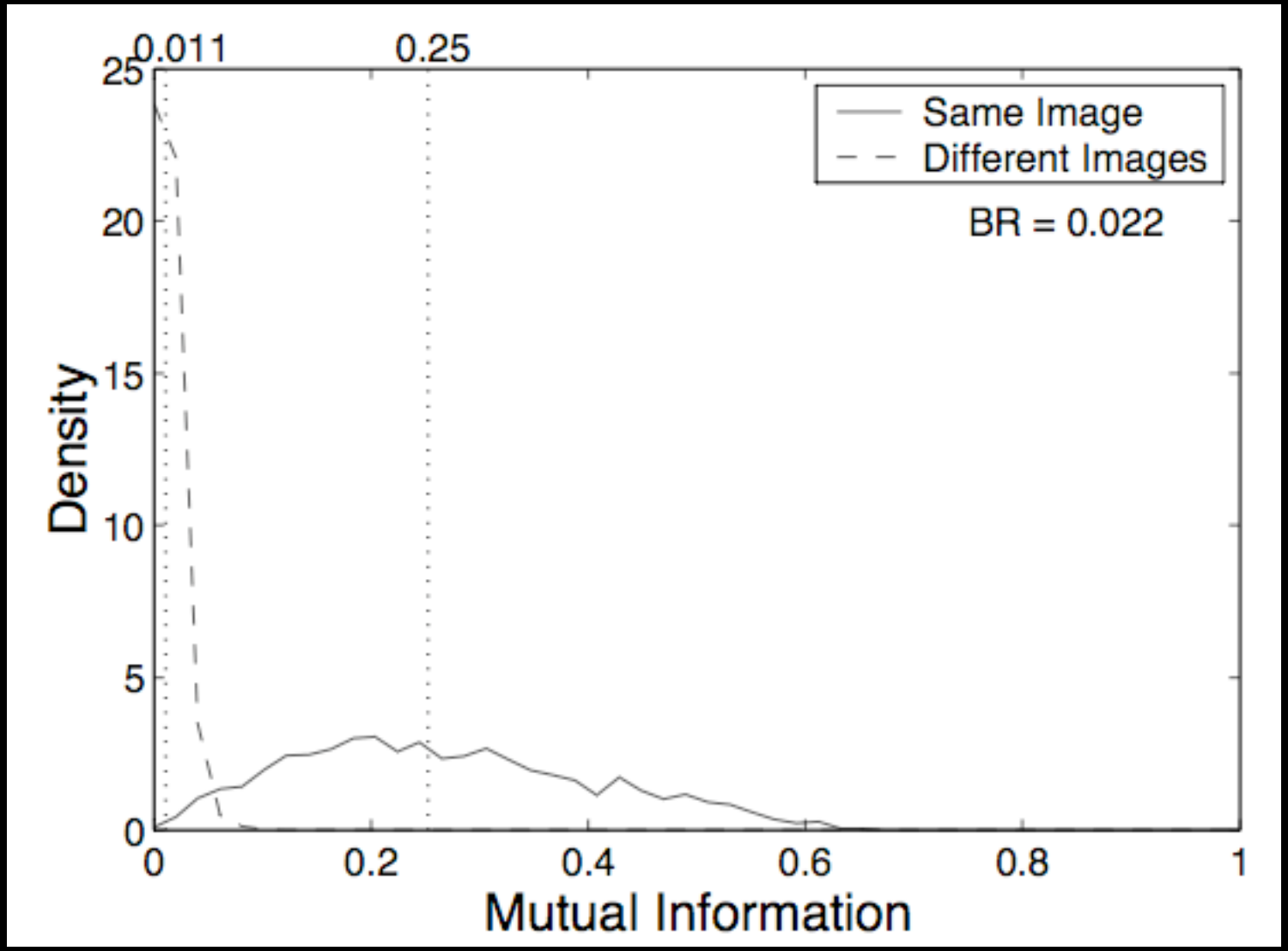
A micro-benchmark for W(i,j), the internal representation used in spectral clustering algorithms.


P(boundary)
Image
W(i,j)
Groundtruth
EigenBoundaries
Segments

Task: Evaluate W(i,j) given S(i,j)

W(i,j) is real-valued, S(i,j) is binary-valued

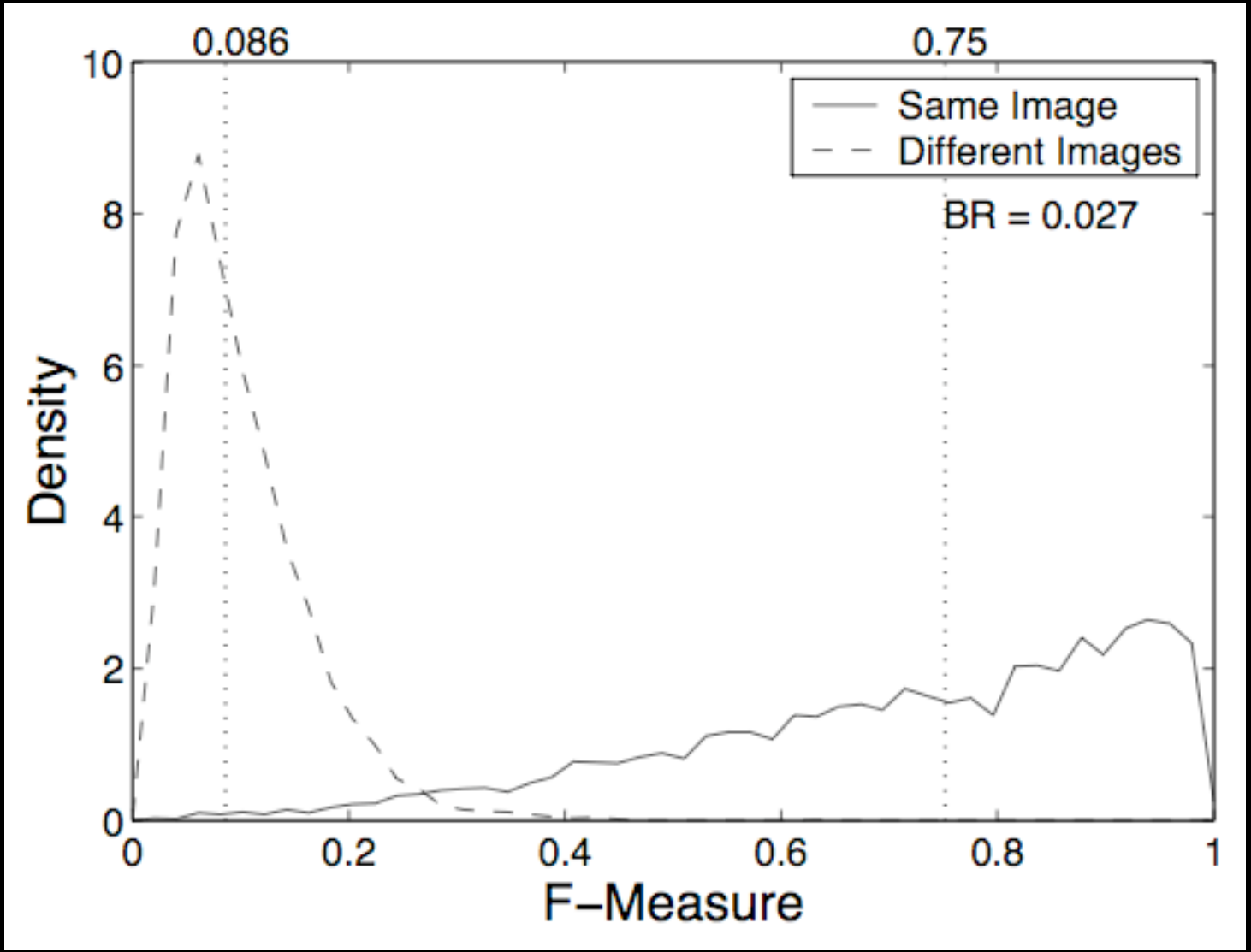
1. Compute mutual information between W and S


0.011
0.25
Same Image
Different Images
BR = 0.022
Density
Mutual Information
0
5
10
15
20
25
0
0.2
0.4
0.6
0.8
1

Task: Evaluate W(i,j) given S(i,j)

W(i,j) is real-valued, S(i,j) is binary-valued

1. Compute mutual information between W and S
2. Treat W as a detector for S, and compute PR curves as before.


0.086
0.75
Same Image
Different Images
BR = 0.027
Density
F–Measure
0
2
4
6
8
10
0
0.2
0.4
0.6
0.8
1

Task: Evaluate W(i,j) given S(i,j)

W(i,j) is real-valued, S(i,j) is binary-valued

- Compute mutual information between W and S
- Treat W as a detector for S, and compute PR curves as before

- Equally discriminative
- Equally arbitrary
- PR curve more informative

Summary

- Benchmarks are worth the effort
  - Borrow techniques from other areas
- Leverage dataset/groundtruth effort by formulating benchmarks at many levels
  - Application-level / Mid-Level / Micro
- Edgel matching is our measure of choice
  - Intuitive, flexible, useful
- Code & Data:
  - http://www.cs.berkeley.edu/projects/vision/grouping