Contour Detection and Hierarchical Image Segmentation – Some Experiments

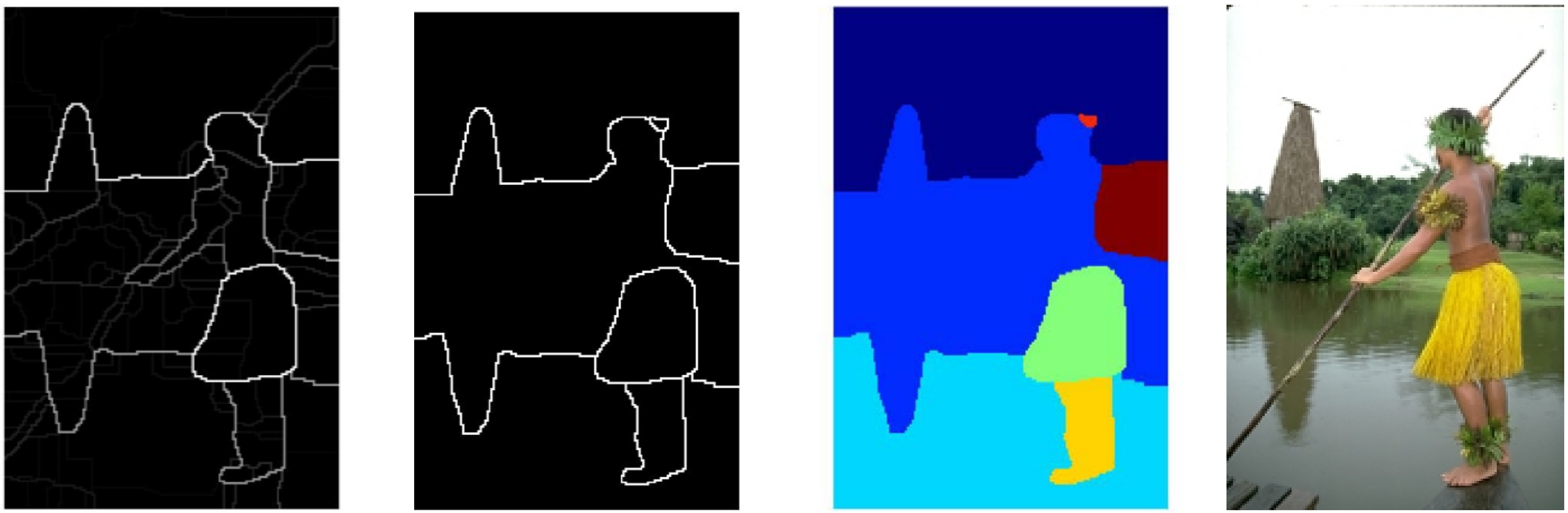
CS395T – Visual Recognition

Presented by Elad Liebman

Overview

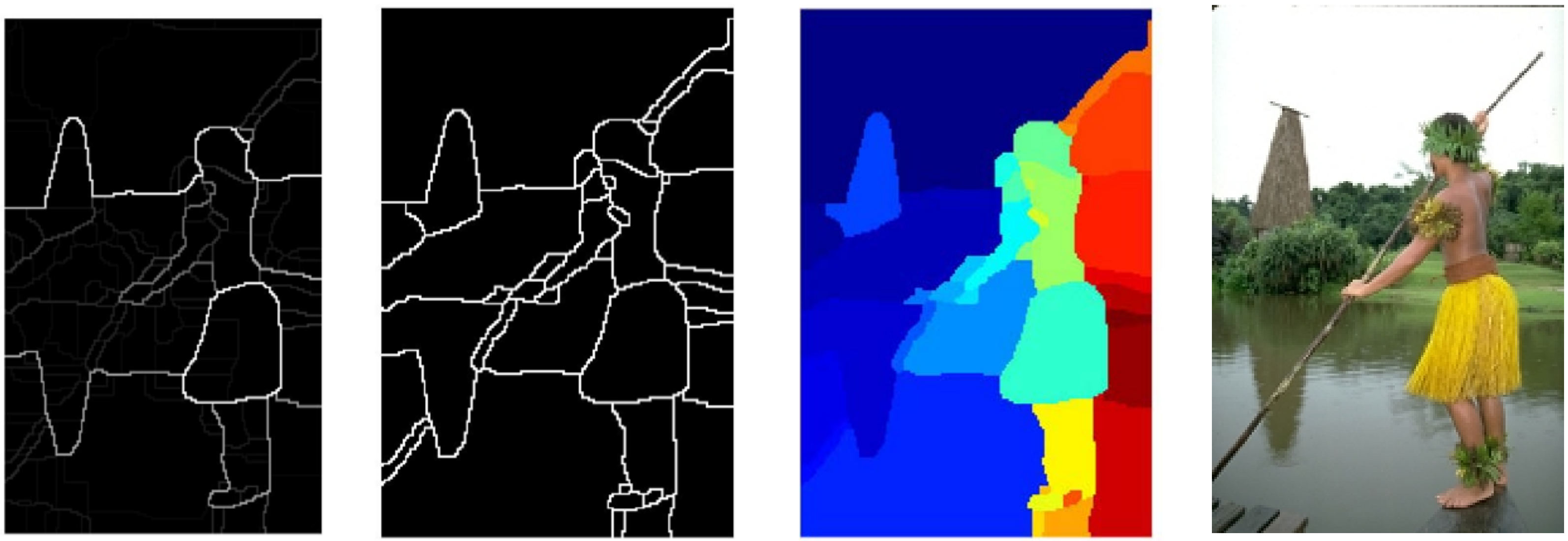
- Understanding the method better:
  - The importance of thresholding the ouput
  - Understanding the inputs better
  - Understanding the UCM
- Pushing for boundaries:
  - Types of difficult inputs
  - Difficult input examples

Warm-up:
choosing the right threshold



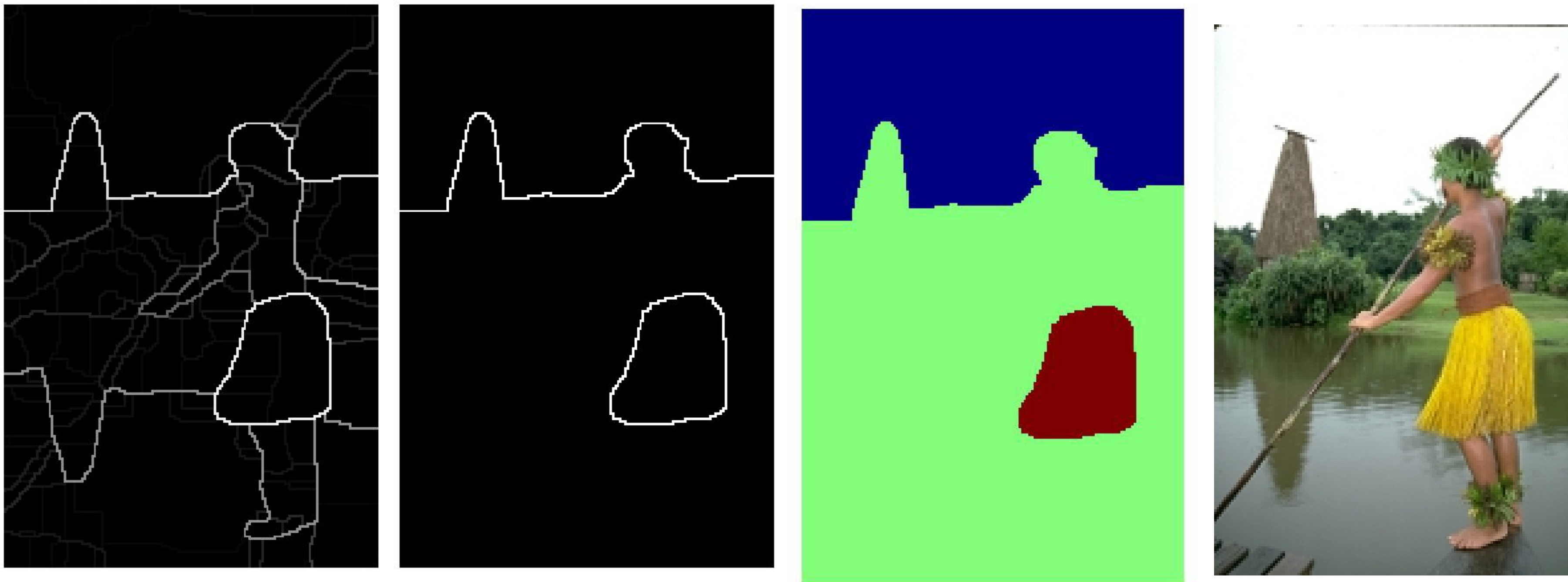
$$k = 0.4$$

Warm-up: choosing the right threshold



$$k = 0.1$$

Warm-up: choosing the right threshold



$$k = 0.8$$

So what goes into the OWT?

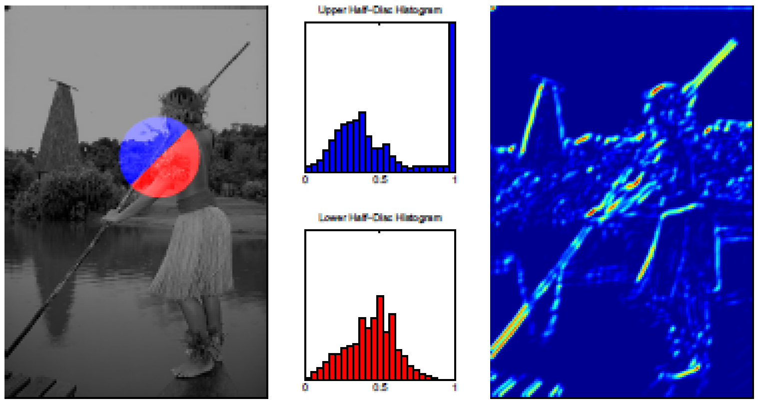
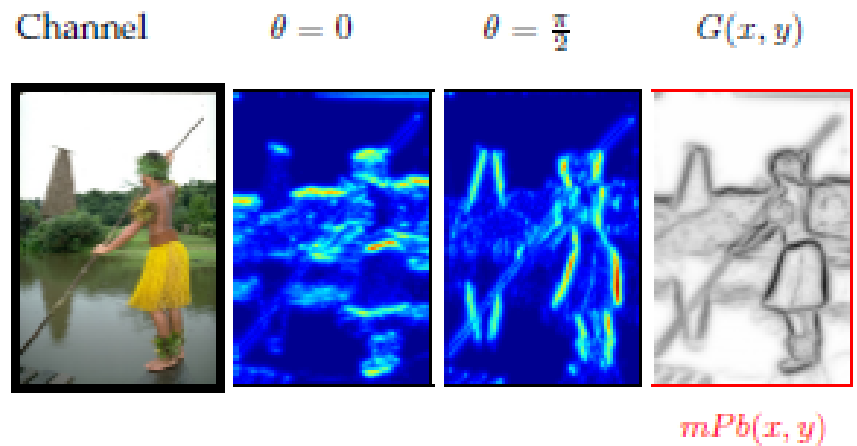


Fig. 4. **Oriented gradient of histograms.** Given an intensity image, consider a circular disc centered at each pixel and split by a diameter at angle θ. We compute histograms of intensity values in each half-disc and output the χ^2 distance between them as the gradient magnitude. The blue and red distributions shown in the middle panel are the histograms of the pixel brightness values in the blue and red regions, respectively, in the left image. The right panel shows an example result for a disc of radius 5 pixels at orientation $\theta = \frac{\pi}{4}$ after applying a second-order Savitzky-Golay smoothing filter to the raw histogram difference output. Note that the left panel displays a larger disc (radius 50 pixels) for illustrative purposes.



- From Contours to Regions: An Empirical Evaluation. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. CVPR 2009.
- Contour Detection and Hierarchical Image Segmentation. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. PAMI 2011

The Features

- Difference in feature channels on the two halves of a disc of radius σ and orientation θ.
- Feature channels in our case:
  - Color gradients
  - Brightness gradients
  - Texture gradients
- Comparison between the two disc halves using χ^2 distance.

The basic signals

- Color/brightness channels based on Lab color space.
- Repeatedly generated at different scales (different σ radii values - $\frac{\sigma}{2}, \sigma, 2\sigma$).
- All in all we get 13 different inputs.

Illustration n. 1: Color gradient a (red-green scale)

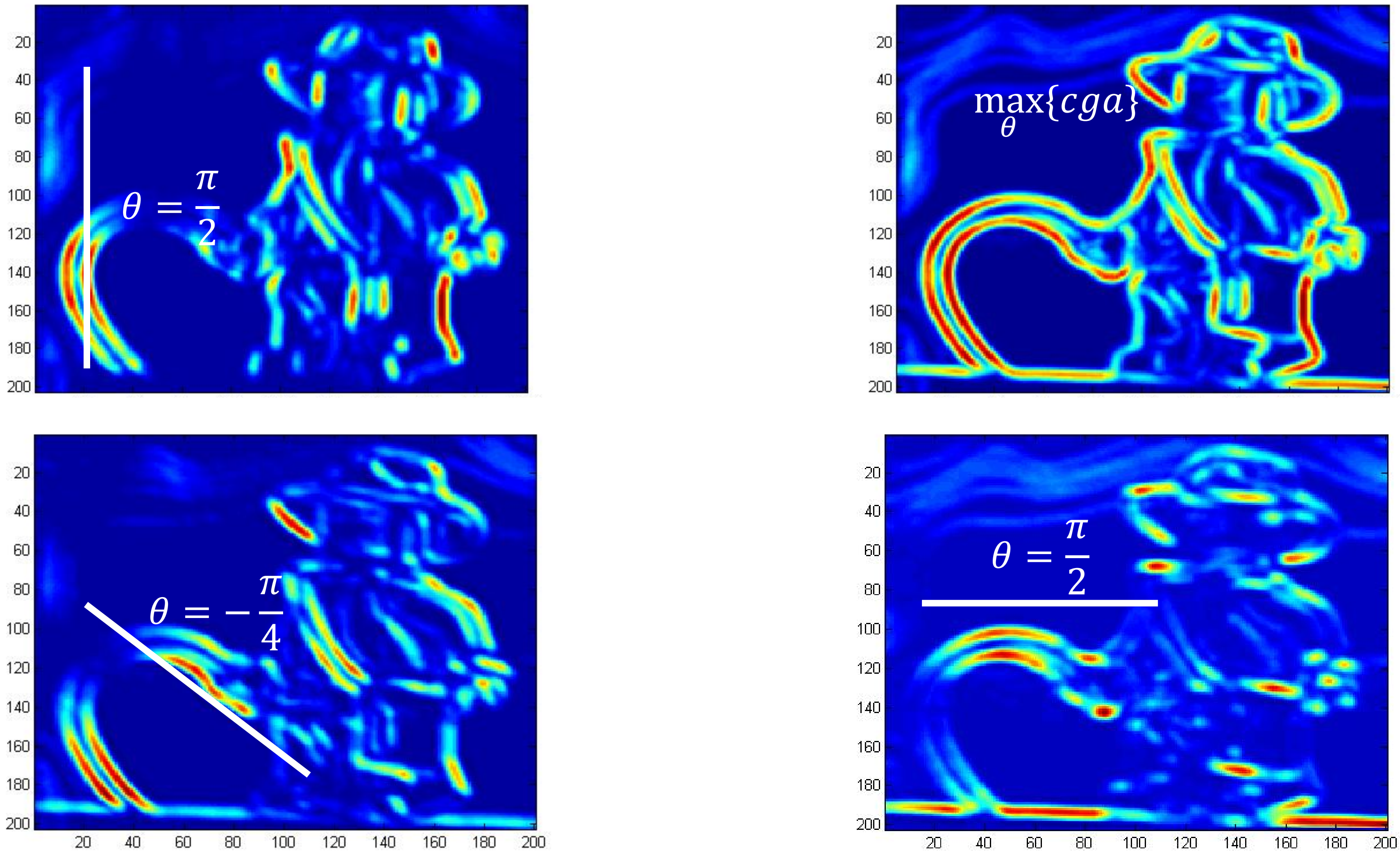


Illustration n. 2: Color gradient a, different σ value

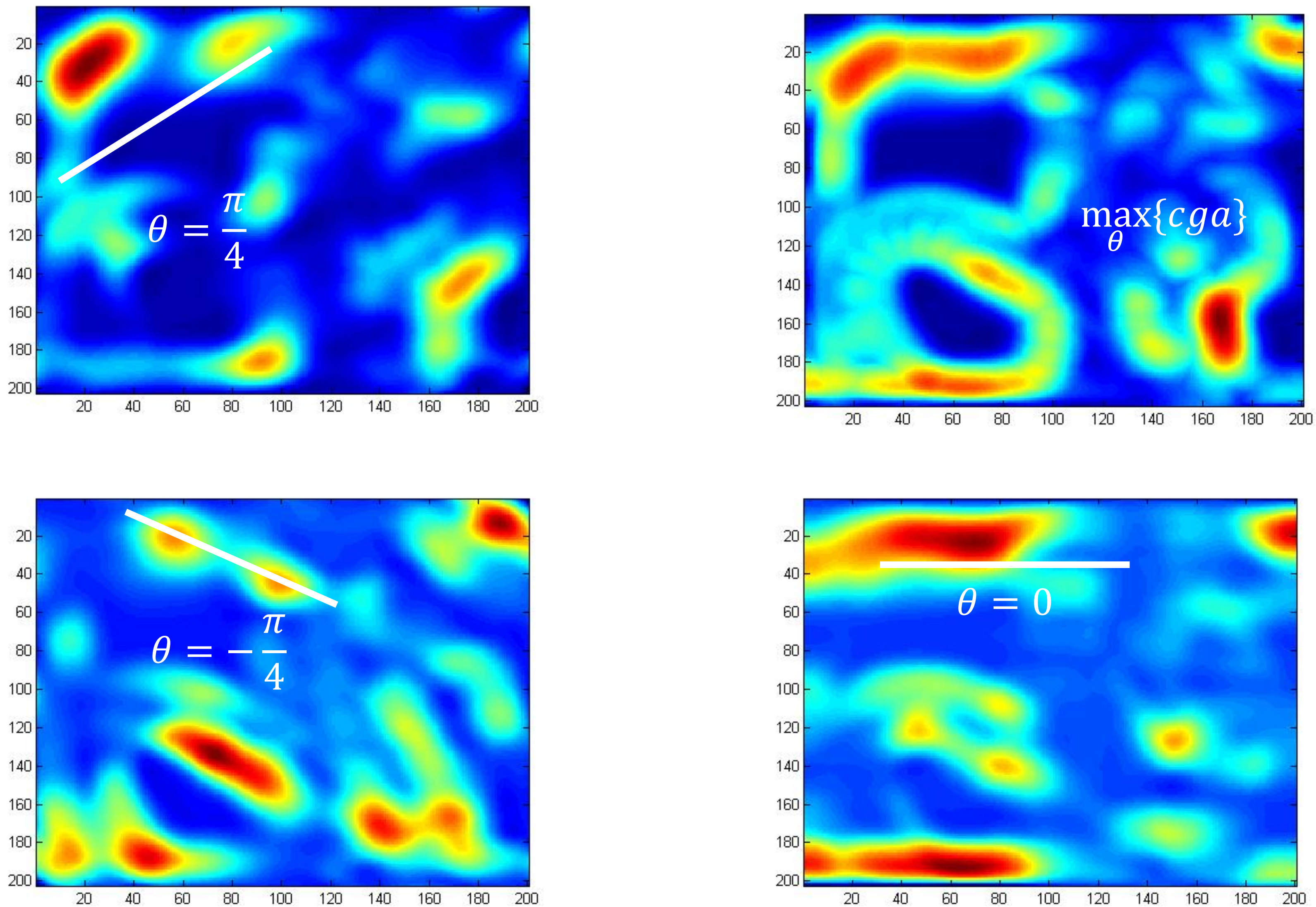
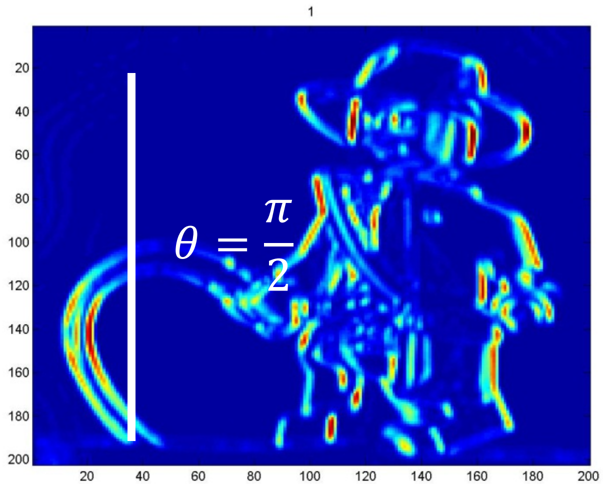
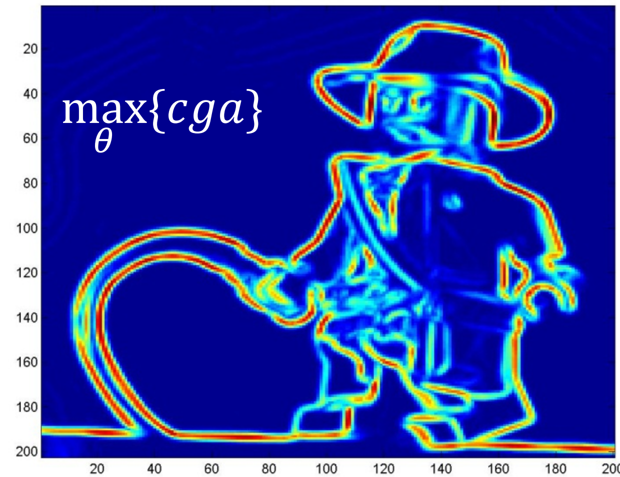
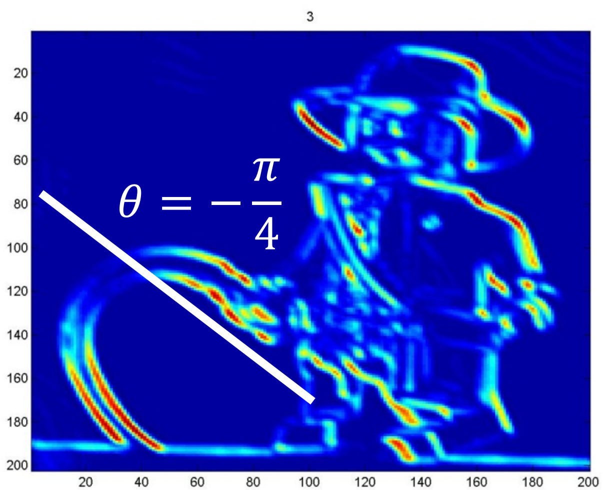


Illustration 3:
Brightness gradient (light-dark scale)







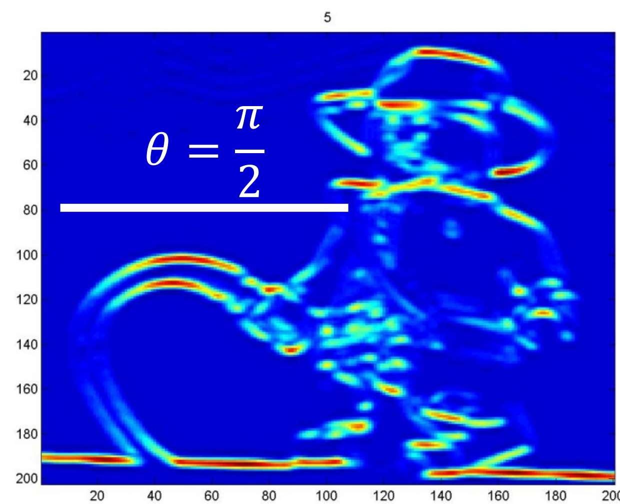
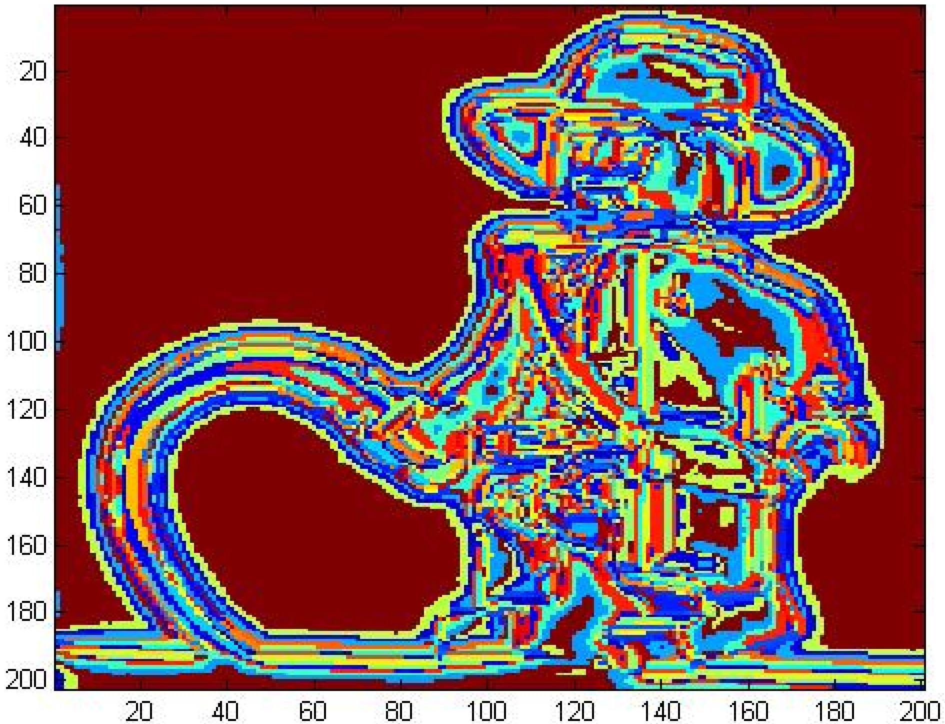


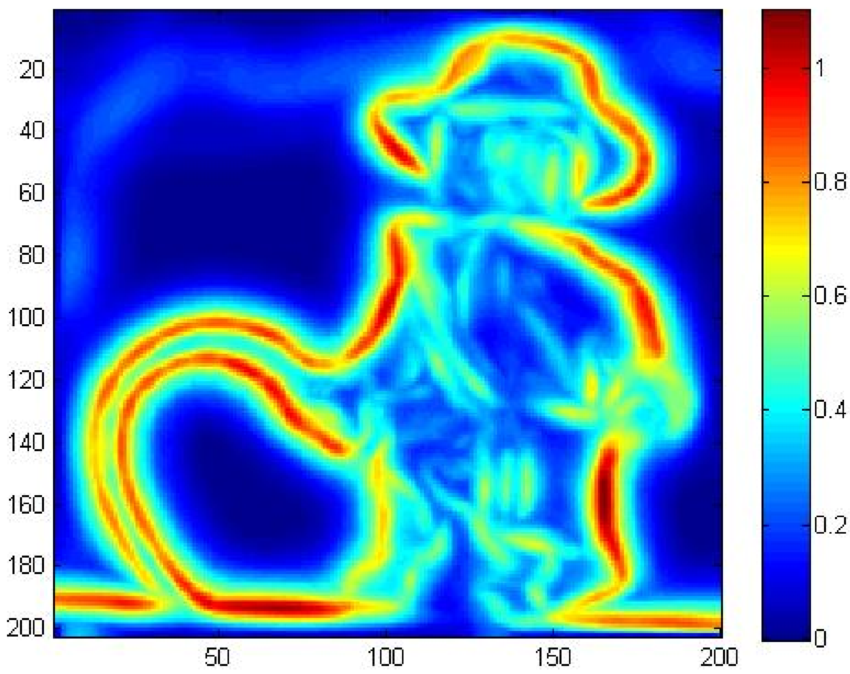
Illustration 4: Texton Map



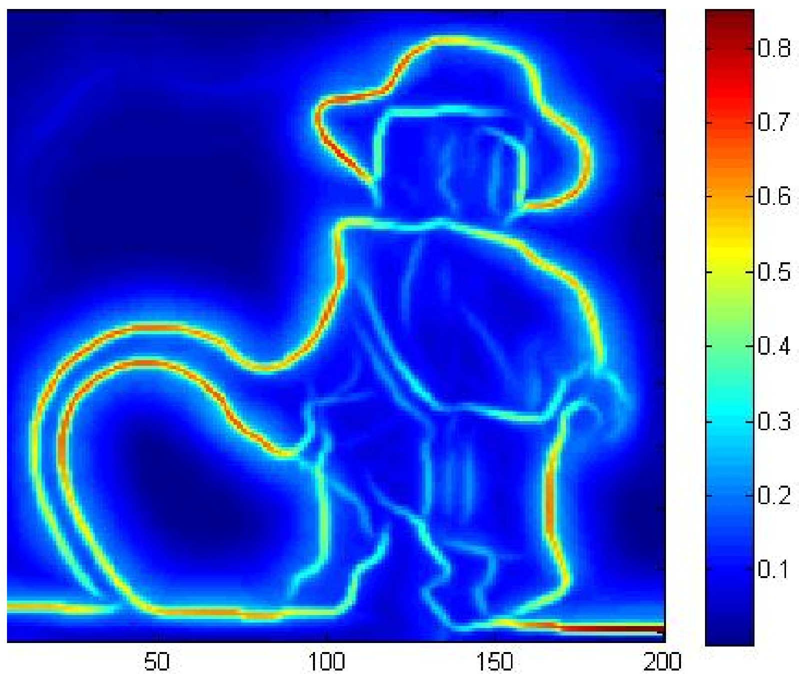
Multiscale Pb and Spectral/Global Pb

- The features are linearly combined to create a unified signal, the multiscale Pb input.
- They are then “globalized” spectrally.
- An affinity matrix w is constructed with W_{ij} being the maximal value of mPb along the line connecting i and j.
- Generalized eigenvectors are then extracted and combined (after processing) with the local Pb data to create the final input for the OWT.

mPb vs. gPb



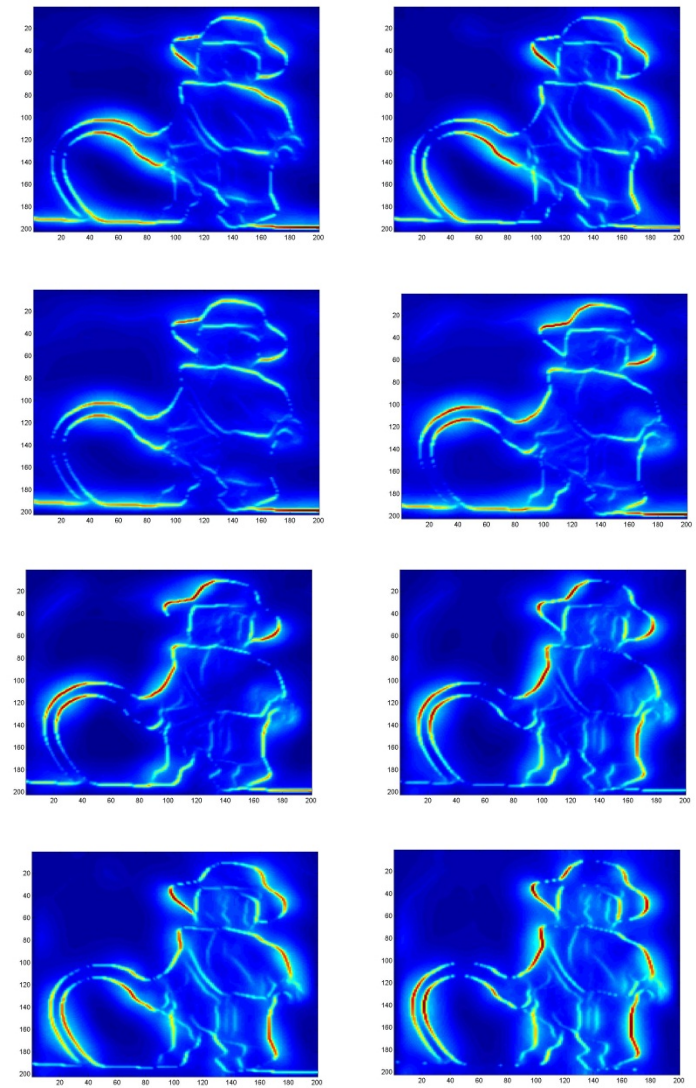
20
40
60
80
100
120
140
160
180
200
50
100
150
200
1
0.8
0.6
0.4
0.2
0



50
100
150
200
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

(In practice we sample 8 orientations…)

$$gPb = \max_{\theta}\{$$

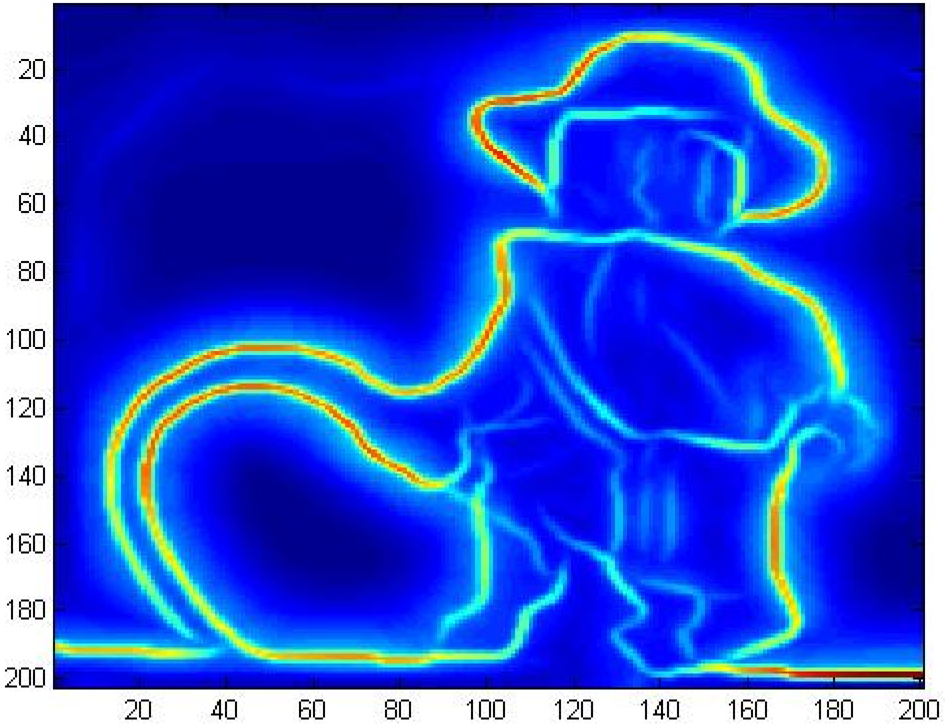


$$\}$$

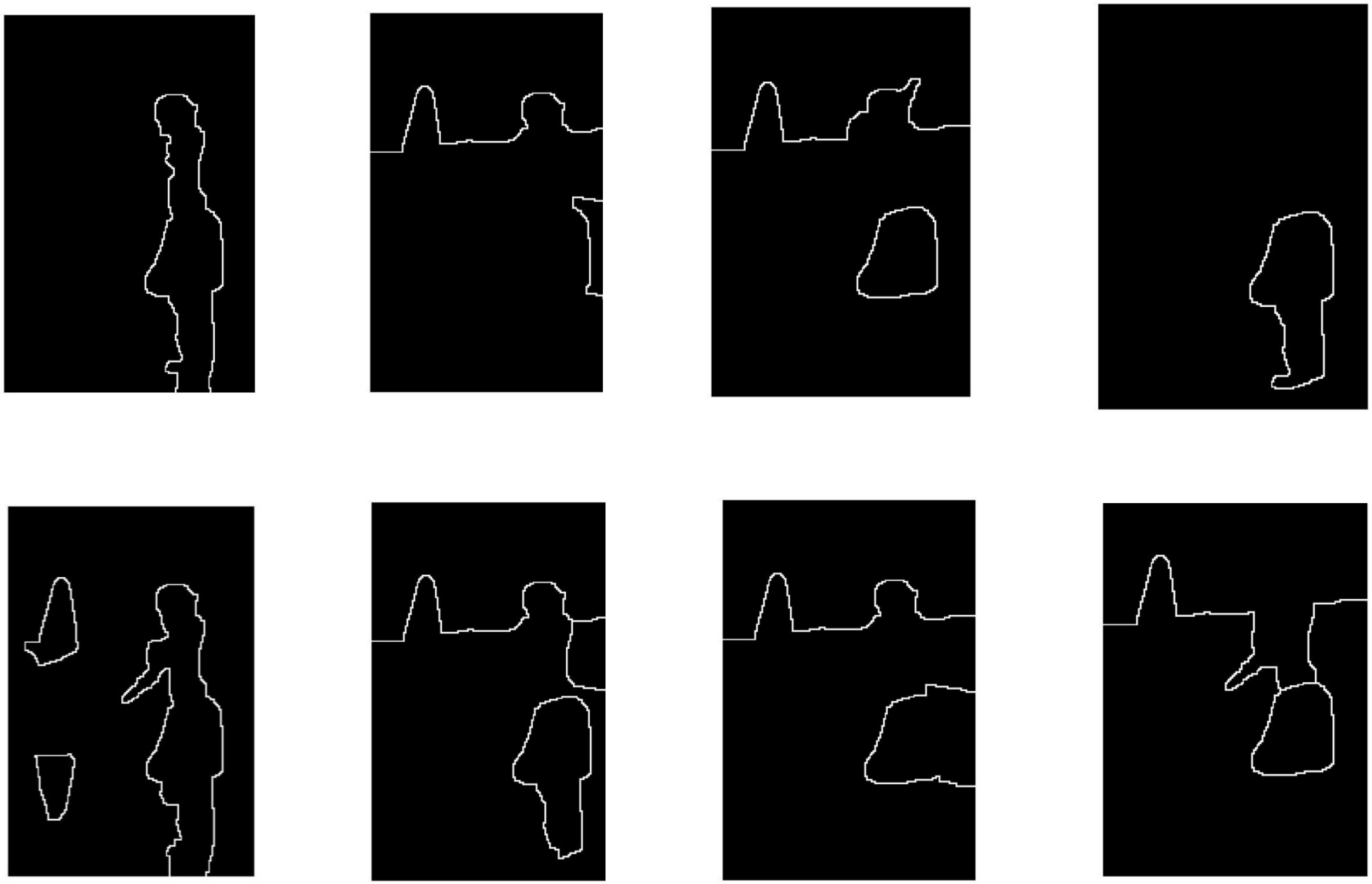
$$\theta = [0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \ldots, \frac{7\pi}{8}]$$

(In practice we sample 8 orientations…)

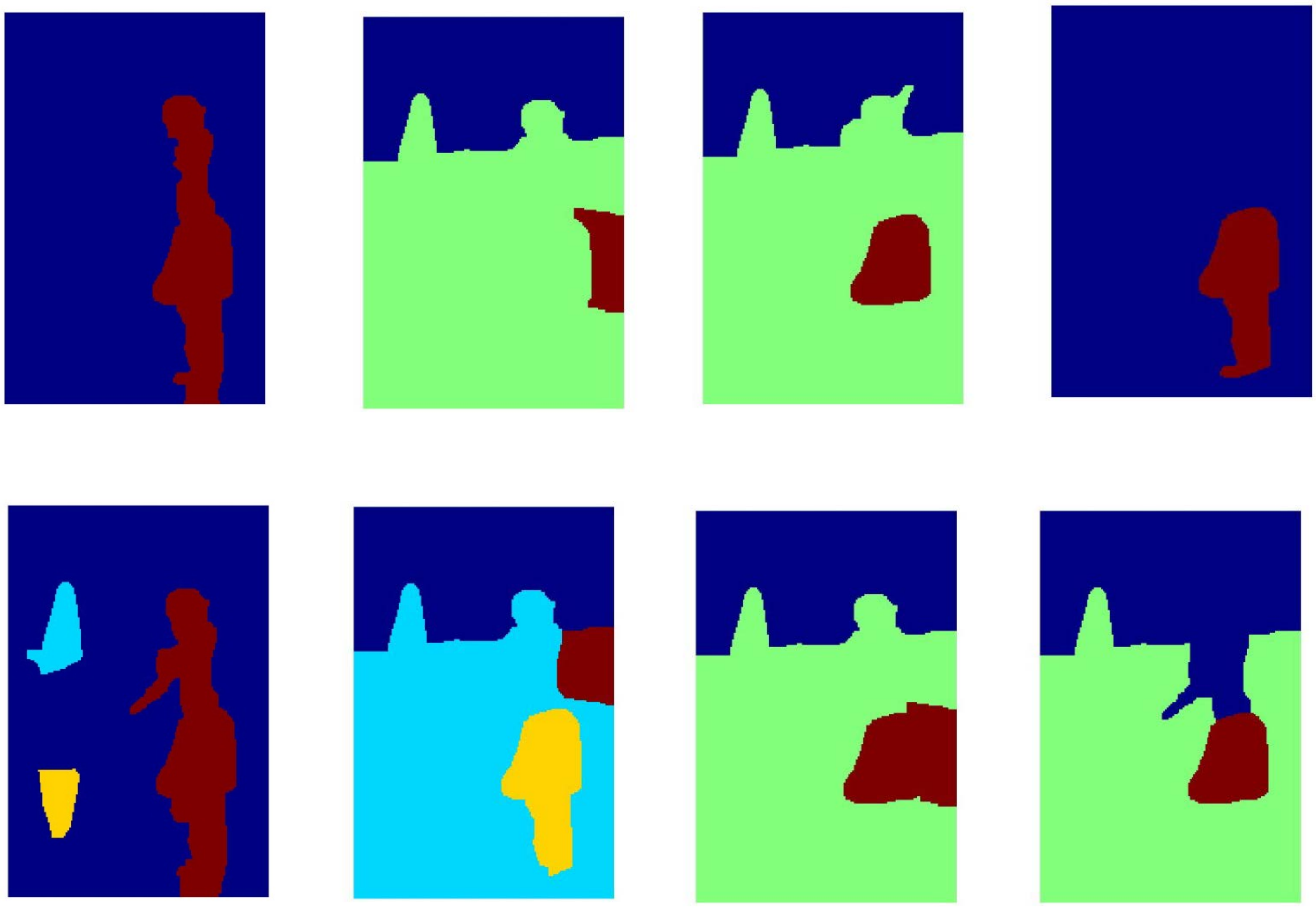
$gPb =$



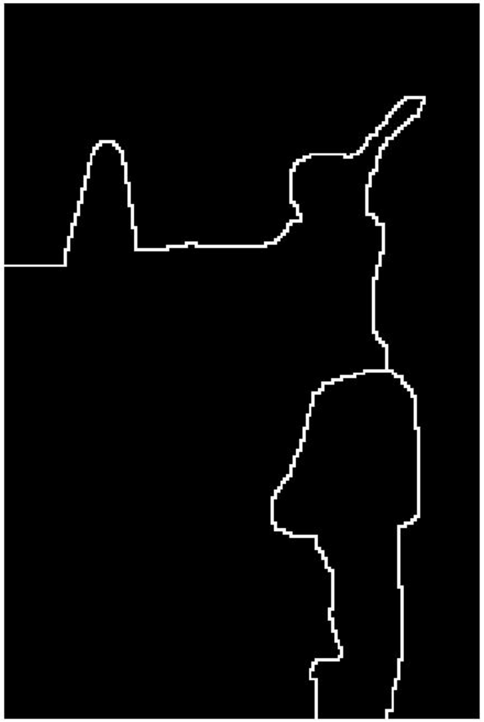
Playing with the orientations a bit…

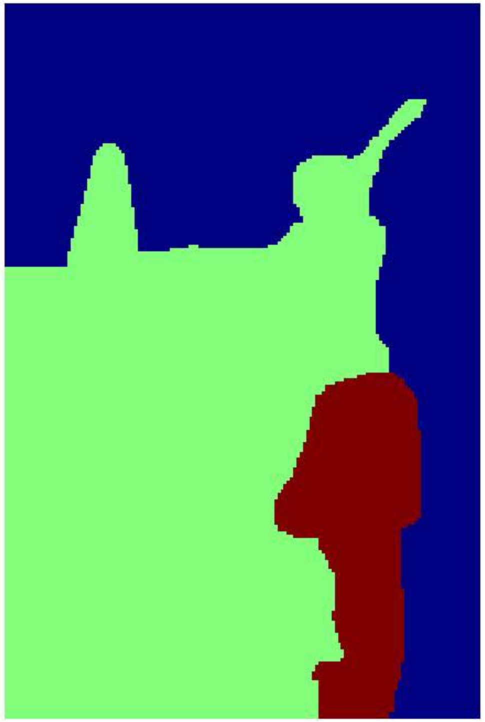


Playing with the orientations a bit…



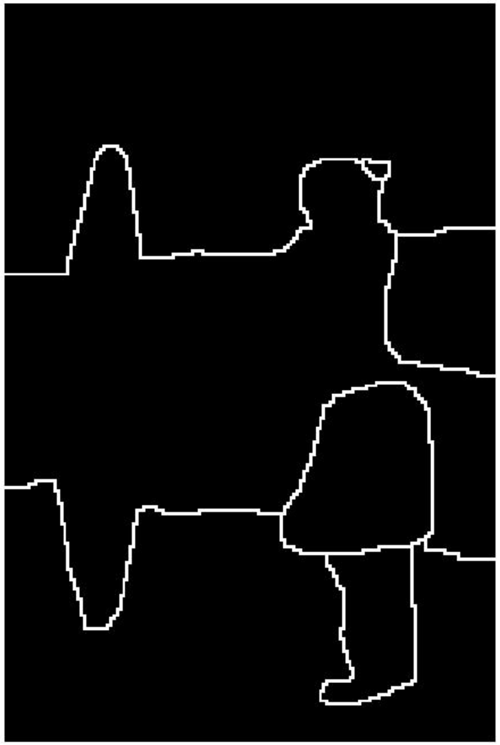
Using less than 8 orientations

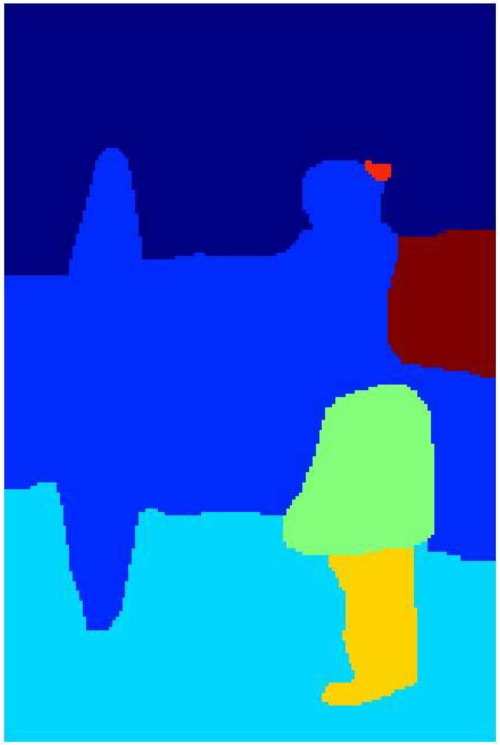




2 orientations

Using less than 8 orientations



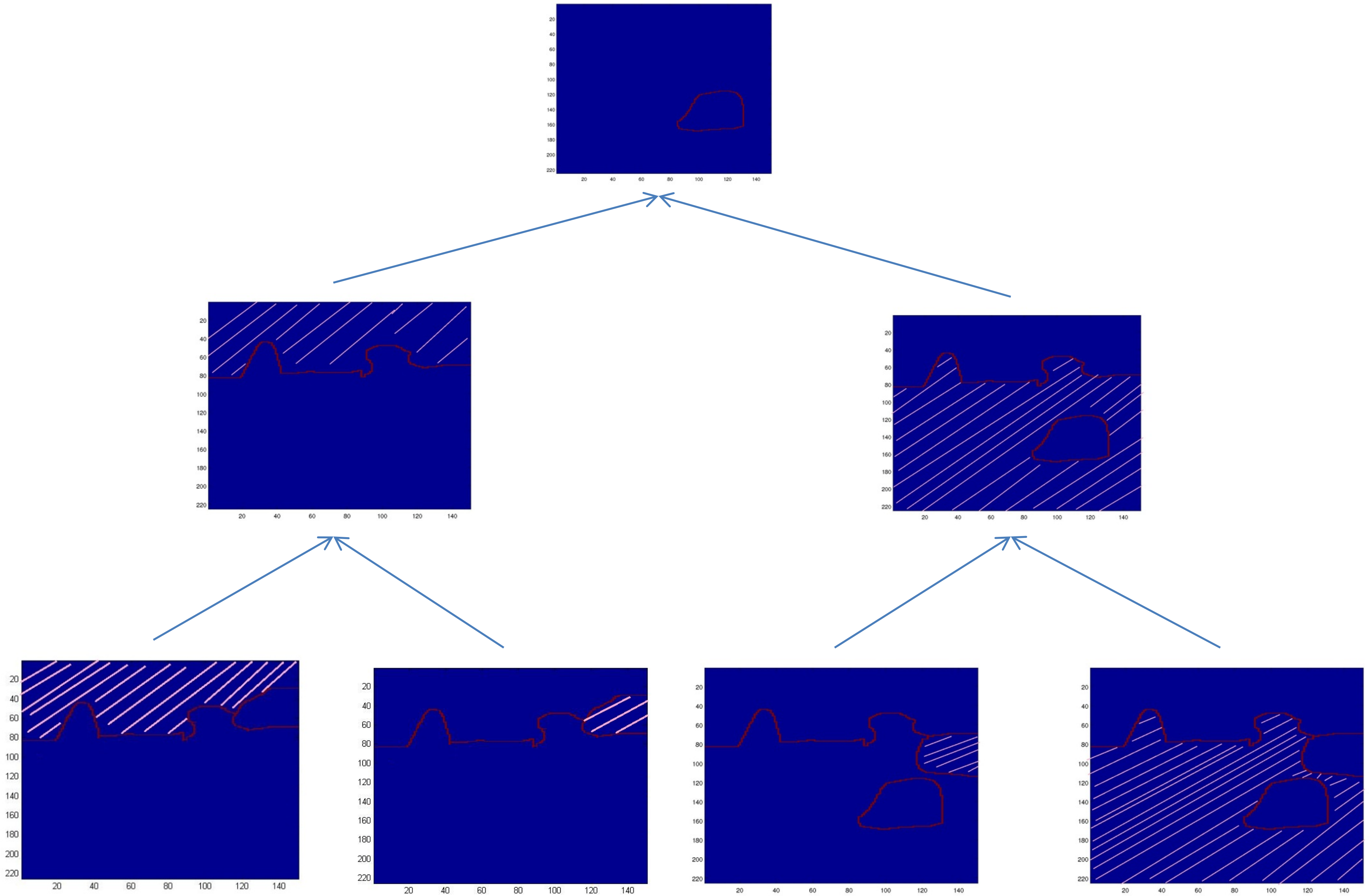


4 orientations (in this case it's enough)

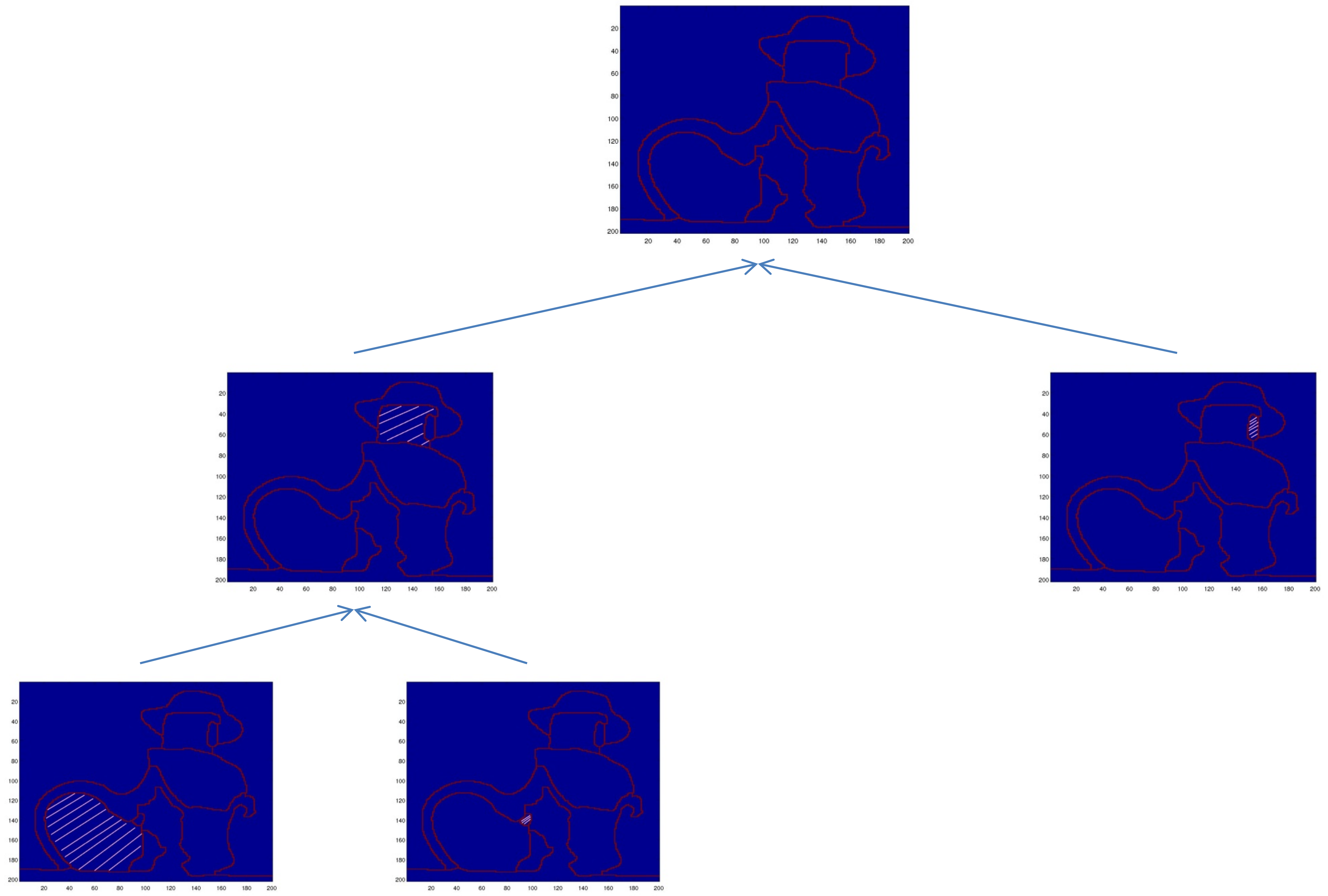
The Ultrametric Contour Map

- The basic idea is to organize and merge sub-regions hierarchically and iteratively.
- At each point we merge two regions with the weakest boundary between them.
- The result is a dendrogram of nested regions.

Example 1



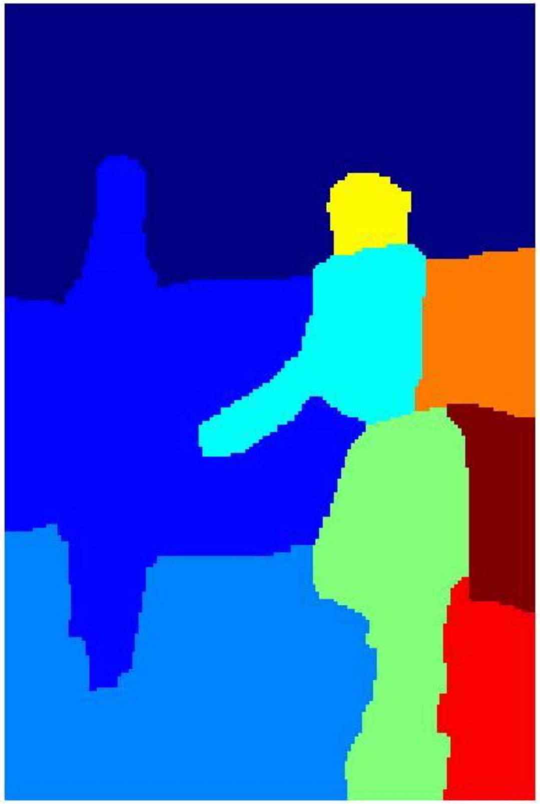
Example 2

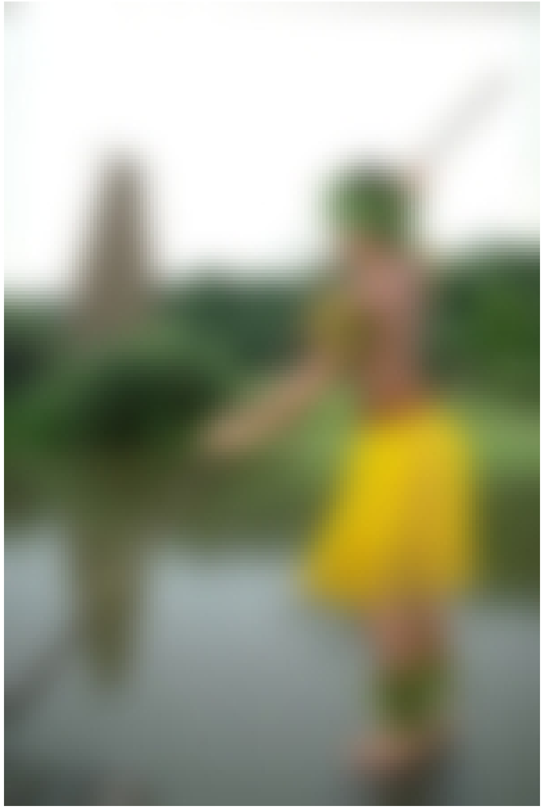


Potentially Difficult Input

- We would like to test the OWT-UCM method against problematic input.
- See where it might break.

warming up - blurring



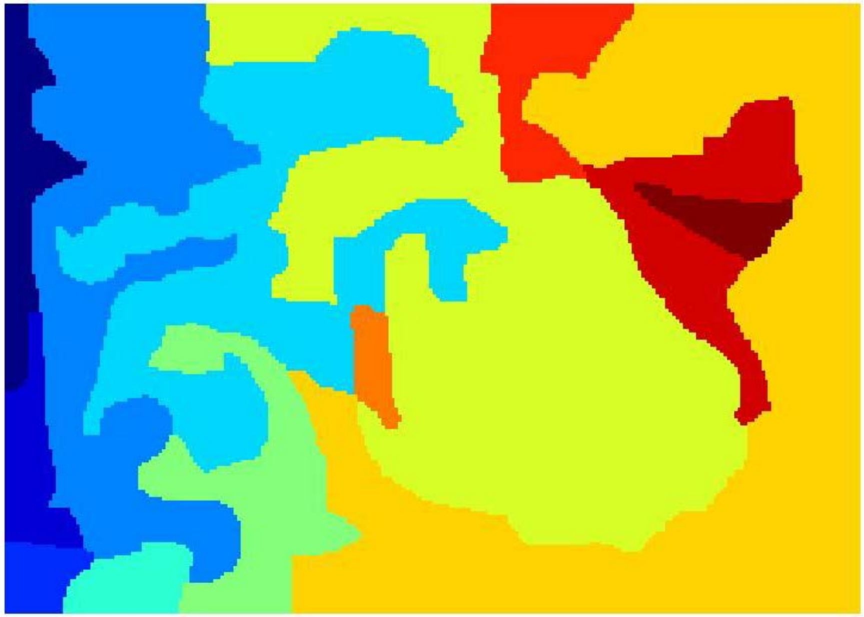


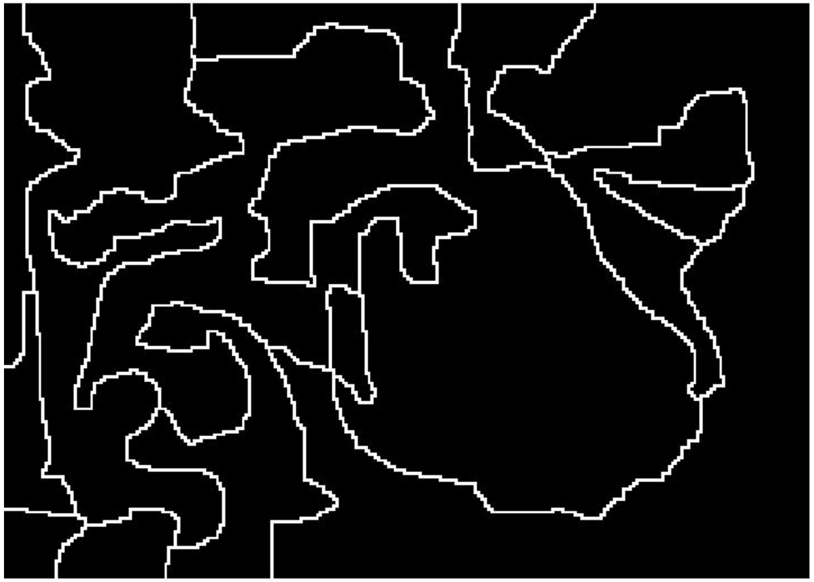
Potentially Difficult Input II

- Blurring isn’t very interesting – doesn’t reflect the kind of challenges the algorithm is likely to face.
- What realistic problems exist?
  - Difficult patches
  - Complex, contour-rich images
  - Not enough color/texture information

Difficult Input – abstract art

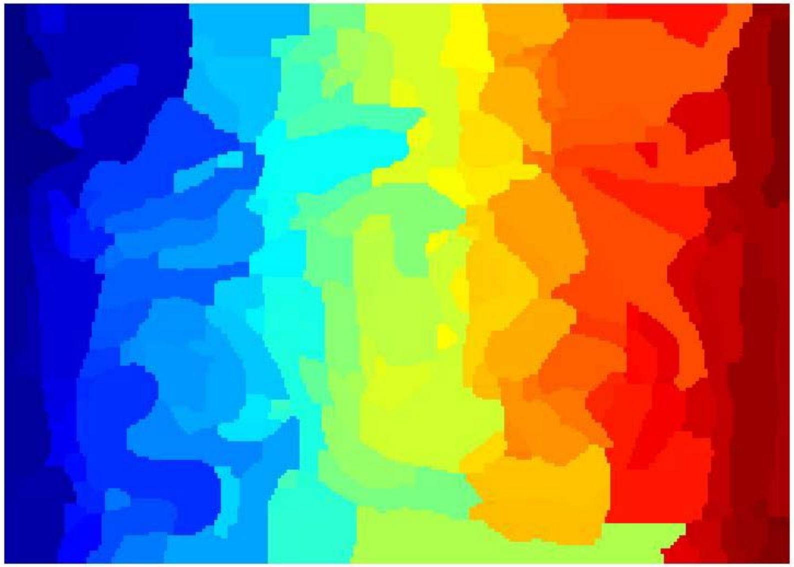
$$k = 0.4$$

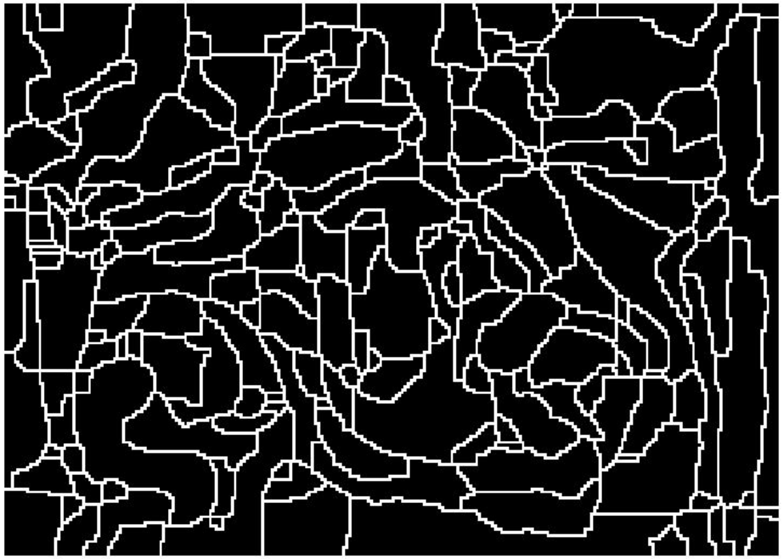




Difficult Input – abstract art

$$k = 0.1$$



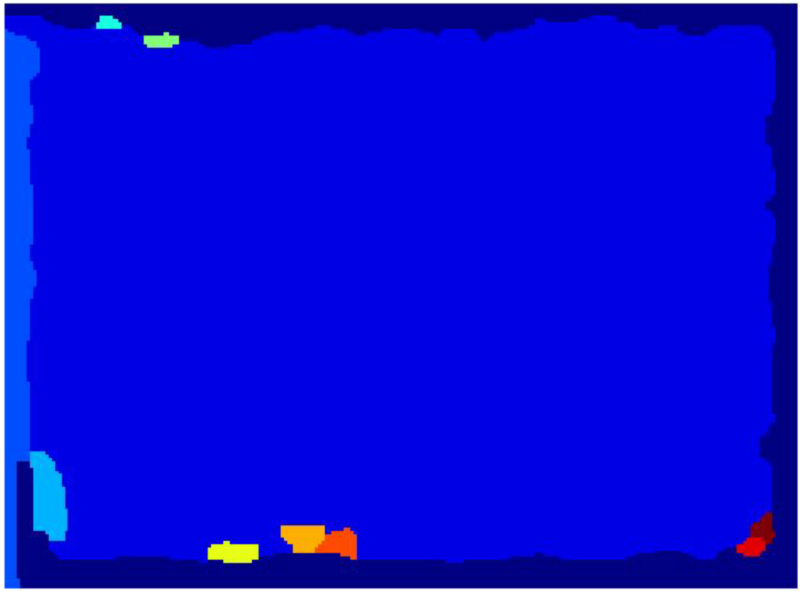


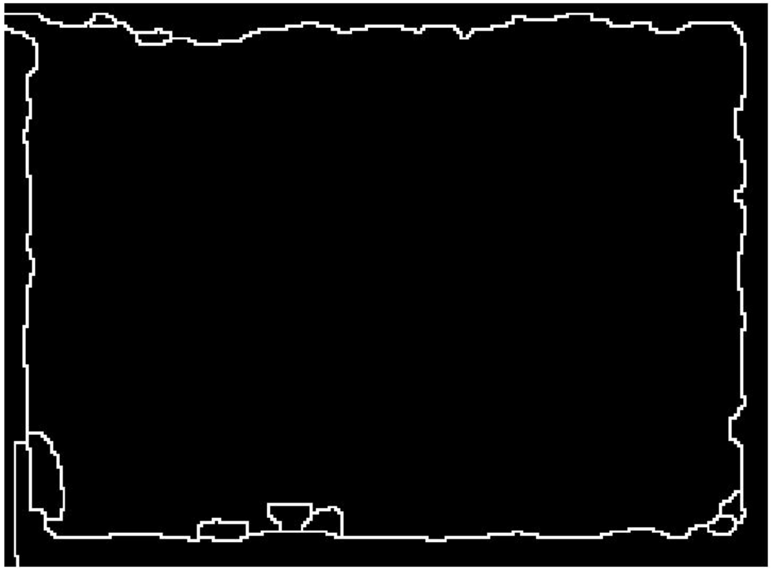
Difficult(er) Input – abstract(er?) art

$$k = 0.4$$

Difficult(er) Input – abstract(er?) art

$$k = 0.1$$

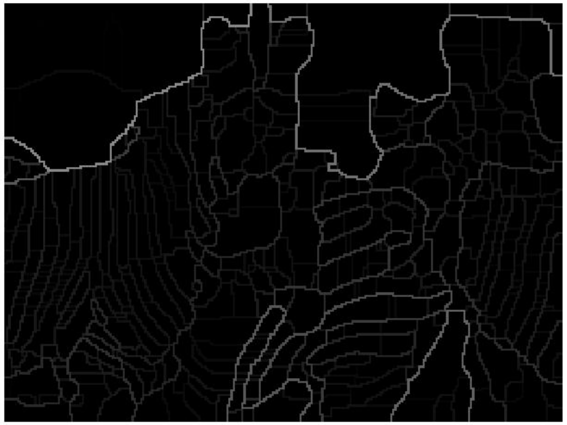




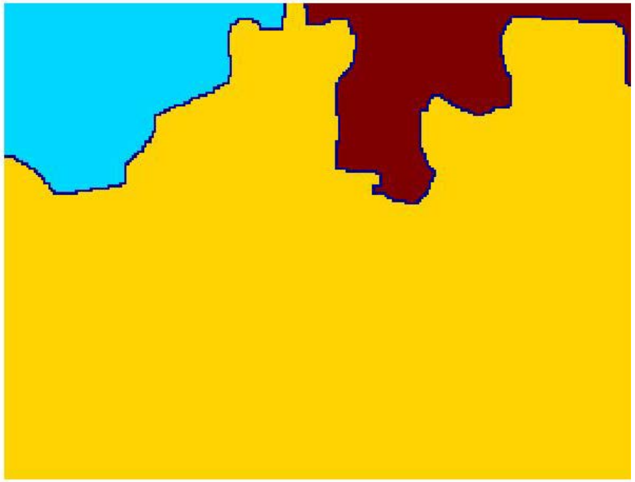
Another Difficult Example

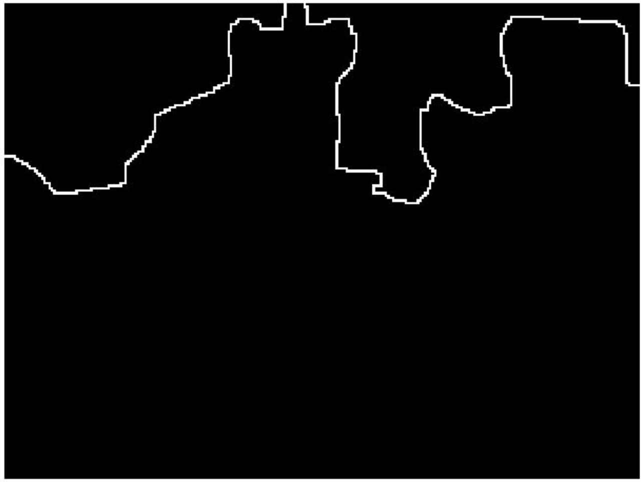
- Occlusion
- Similarity of objects in image:
  - Similar textures
  - Similar colors
  - Contour lines blend

Another Difficult Example

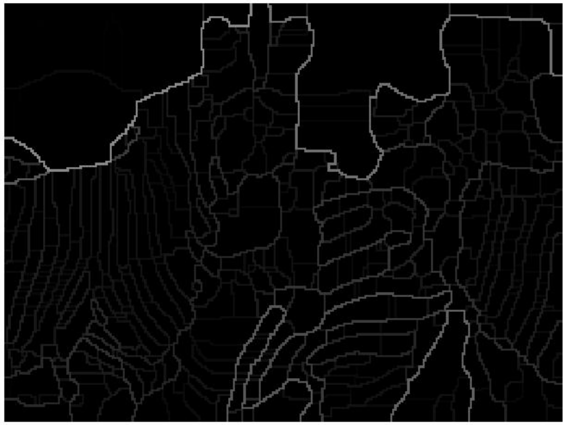


$$k = 0.4$$

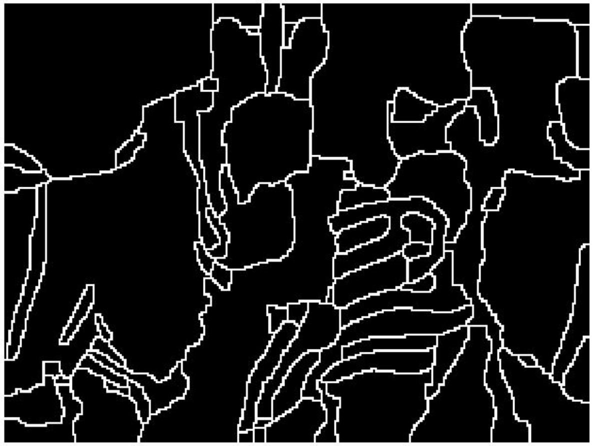


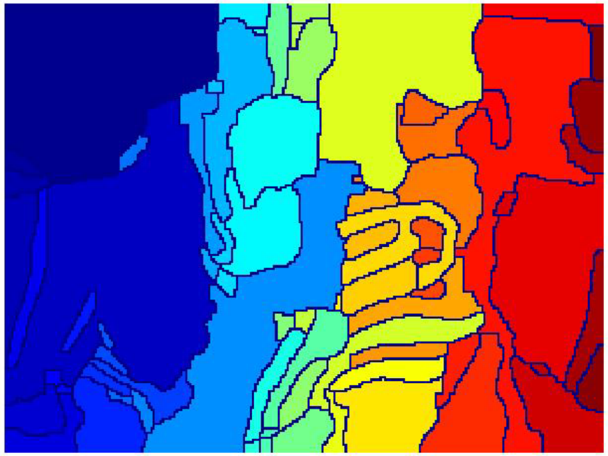


Another Difficult Example

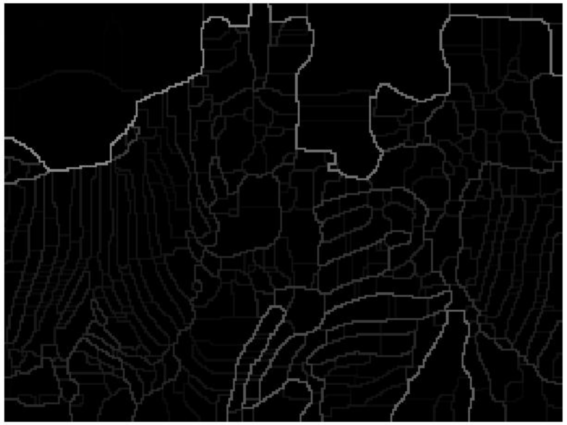


$$k = 0.1$$

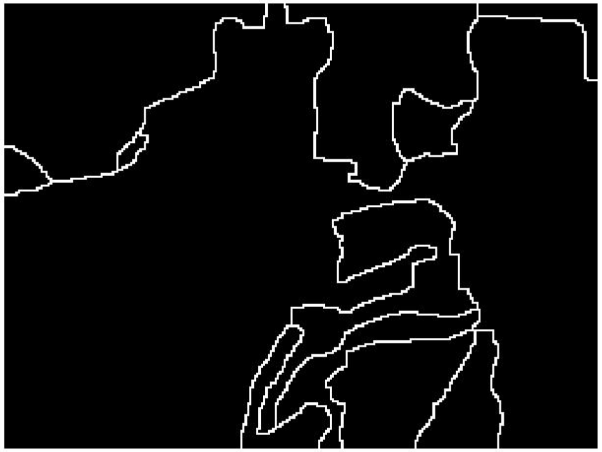


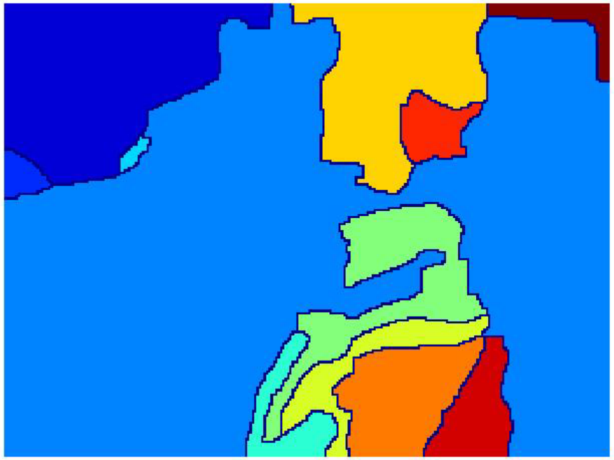


Another Difficult Example

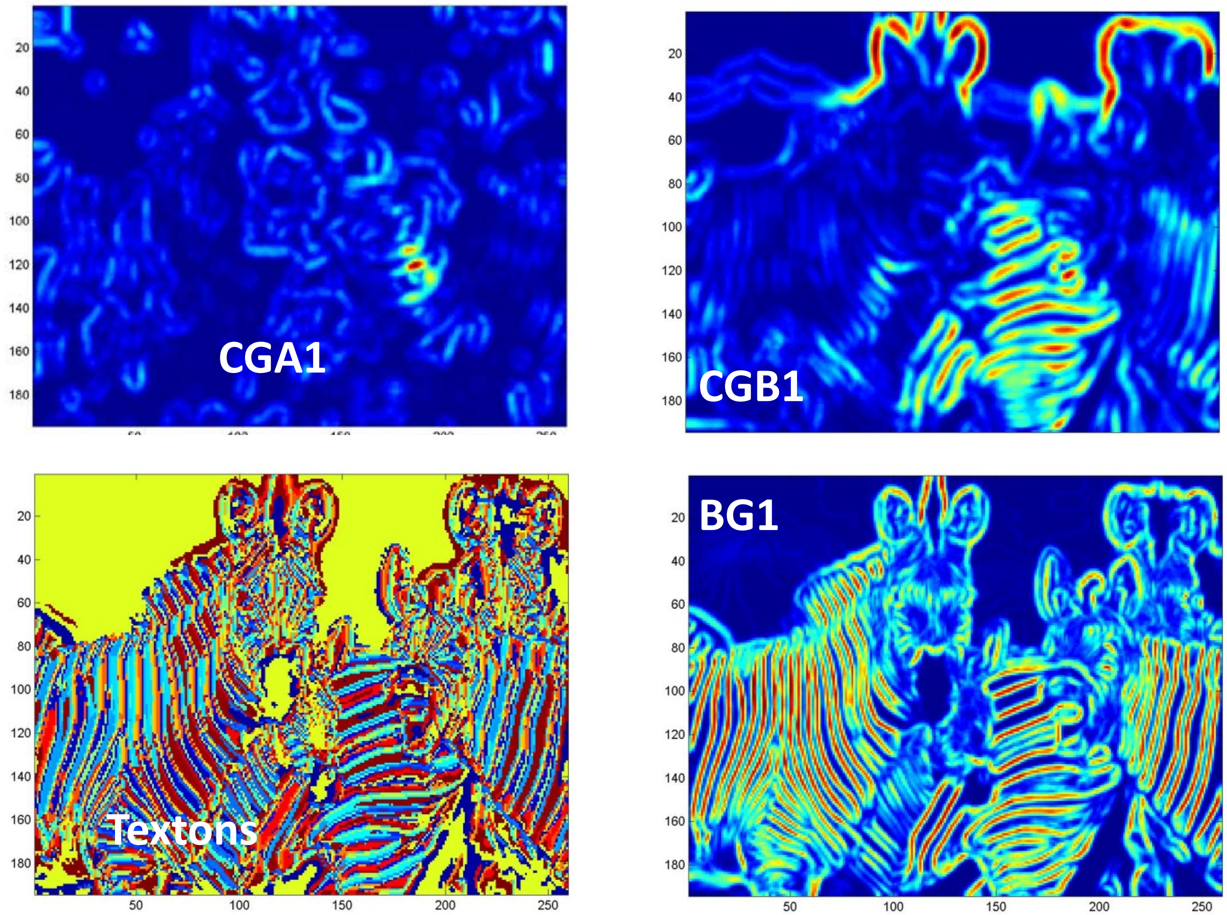


$$k = 0.2$$





Sneaking a look under the hood



- Possibly we need to weight elements differently in certain cases…

Summary

- Studied some of the technical aspects of the method:
  - Threshold selection
  - Understanding the input data
  - Illustration of UCM in action
- Tested the method against difficult input:
  - Problematic contours
  - Complex images
  - Cases where features aren’t informative enough
  - Many similar items occluding one another

References

- From Contours to Regions: An Empirical Evaluation. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. CVPR 2009.
- Contour Detection and Hierarchical Image Segmentation. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. PAMI 2011
- Using Contours to Detect and Localize Junctions in Natural Images. M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. CVPR 2008.
- What are Textons? S. Zhu, C. Guo, Y. Want Z. Xu, International Journal of Computer Vision 2005.
- Class notes for **16-721: Learning-Based Methods in Vision**, taught by Prof. Alexei Efros, CMU.
- Class notes for **CS 143: Introduction to Computer Vision**, taught by Prof. James Hayes, Brown.