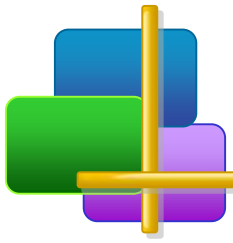
(I'm Fun) Digital Image Fundamentals



Week 6: Segmentation

Thuong Nguyen Canh

Institute for Datability Science, Osaka University

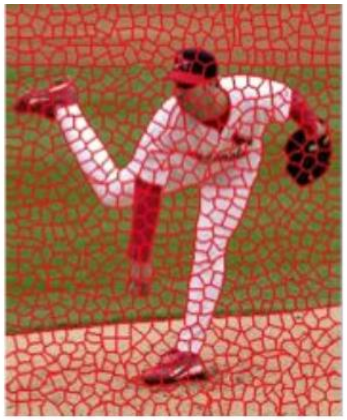
November 2019

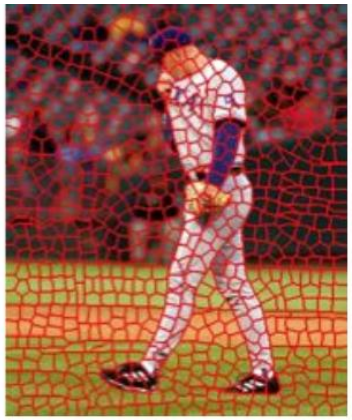
This material is based on the Digital Image Processing, Stanford University, Prof. Bernd Girod.

Image Segmentation?

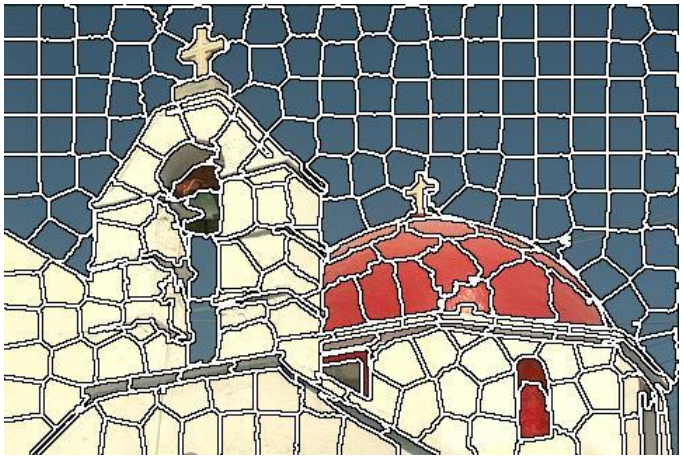
- Group pixels into “meaningful” regions that share some similar properties

“superpixels”





[X. Ren and J. Malik 2003, Mori et al. 2005]



- Segment images into meaning full objects

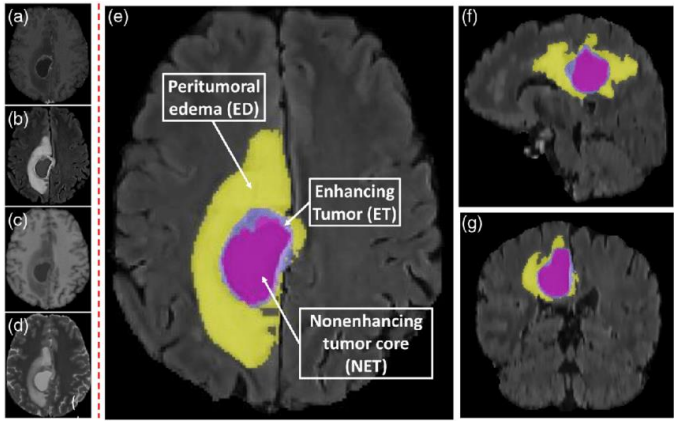


Image segmentation is the process of assigning a label to every pixels

Semantic Segmentation

- Label every pixel: recognize the class of every pixel
- Do not differentiate instances

Mottaghi et al, "The role of context for object detection and semantic segmentation in the wild", CVPR 2014

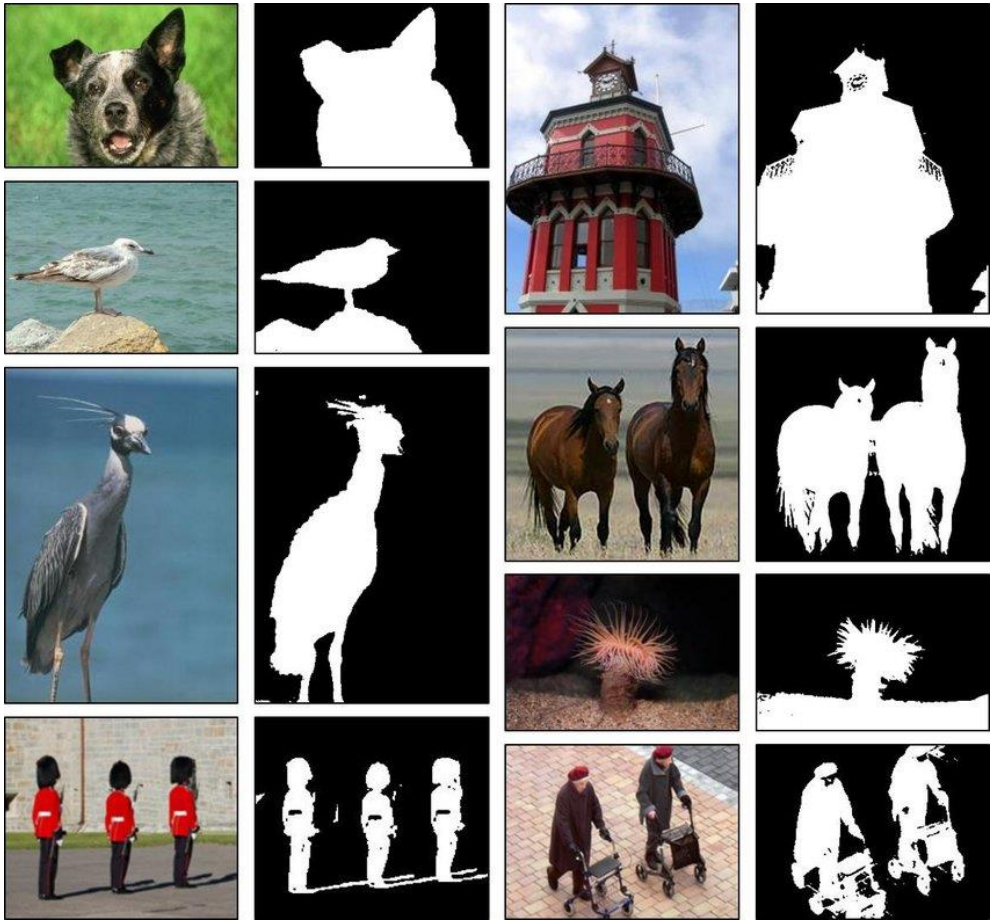
Instance Segmentation

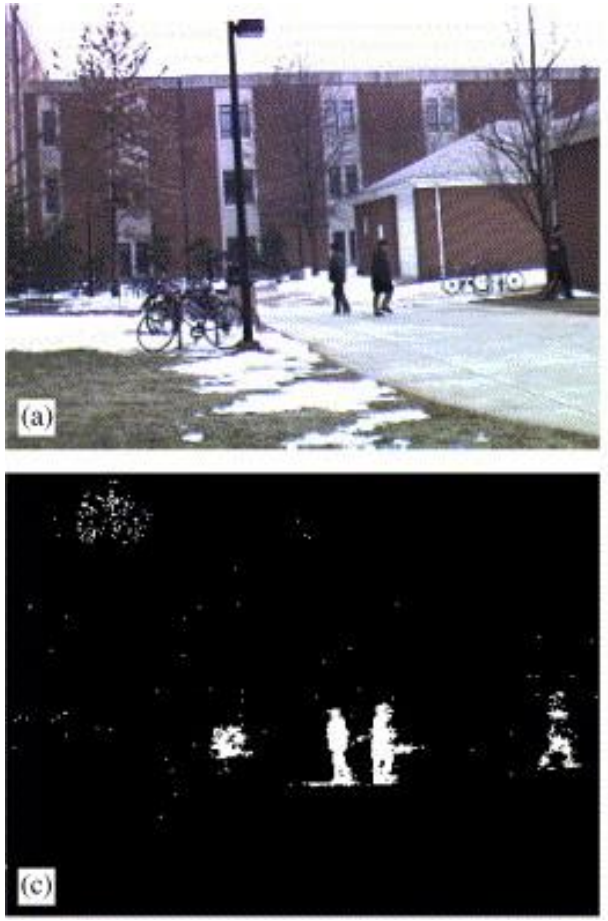
- Detect instances, categorize and label every pixel
- Labels are class-aware and instance-aware

Arnab,Torr "Pixelwise instance segmentation with a dynamically instantiated network", CVPR 2017

Figure-ground Segmentation

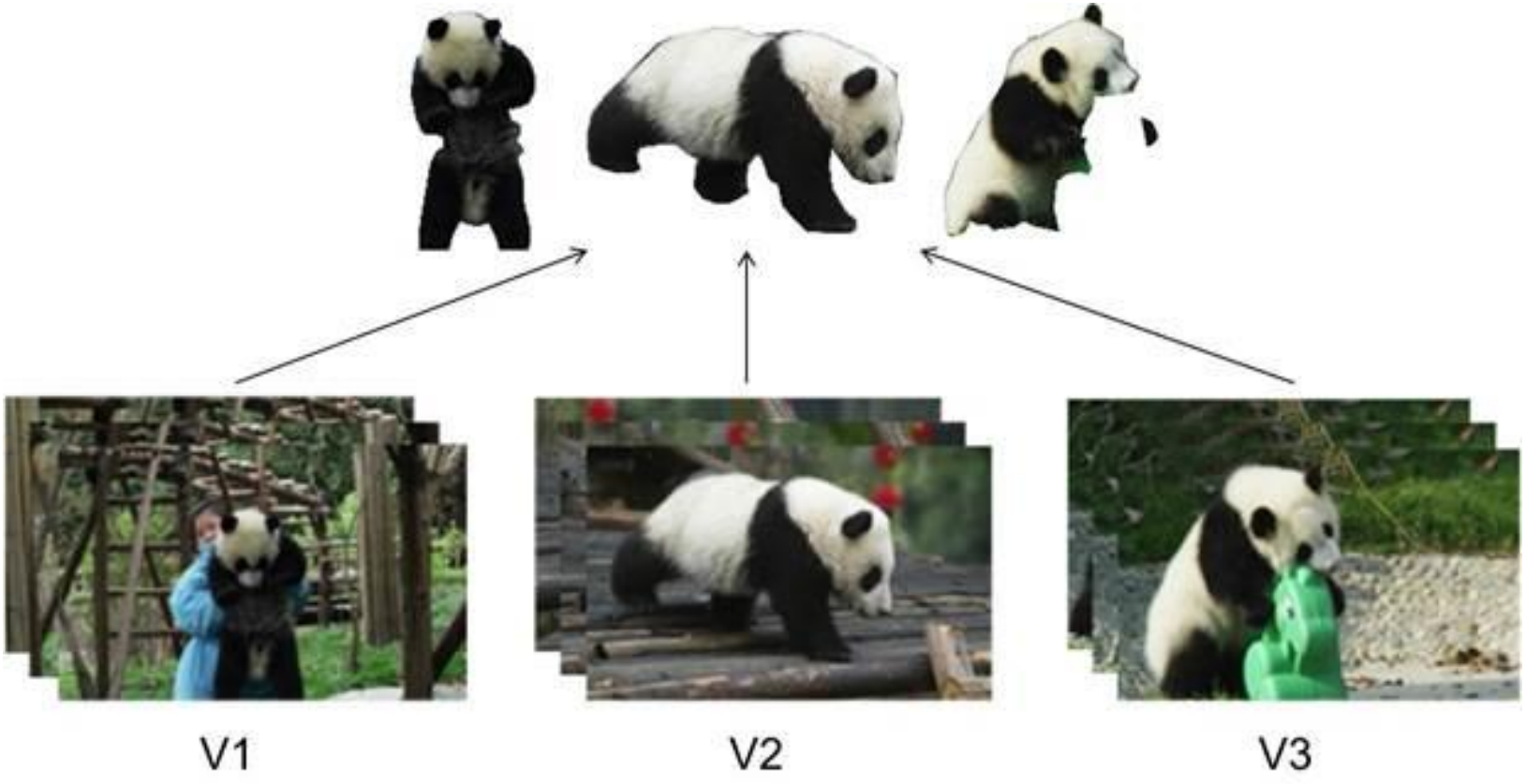
- An binary semantic segmentation





Co-segmentation

- Segment the common objects



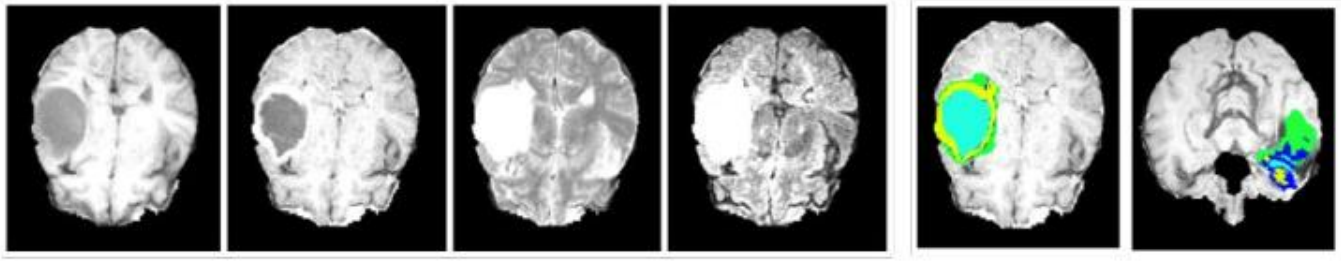
Applications

Image editing and composition (Xu, 2016)

Robotics

Autonomous driving
(cordts, 2016)

Medical image analysis
(Casamitjana, 2017)

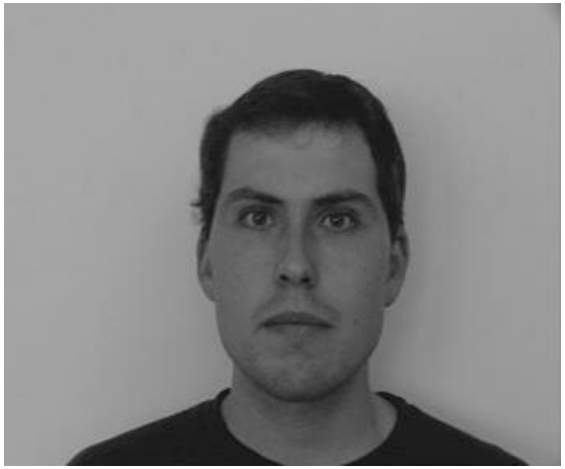


Segmentation Method

- Segmentation as “clustering”
  - Clustering: Group similar data points and represent them with a single token
  - Key challenges
    - What make two points/images/patches similar?
    - How to compute overall grouping?
- Thresholding based
- Edge based
- Region based
- Energy based
- Learning based

Gray-Level Thresholding

Original



$f[x, y] \in \mathbb{N}^{286 \times 334}$

Range 0 ~ 255

Binary Thresholded

$m[x, y] \in \mathbb{B}^{286 \times 334}$

Fixed Threshold 105

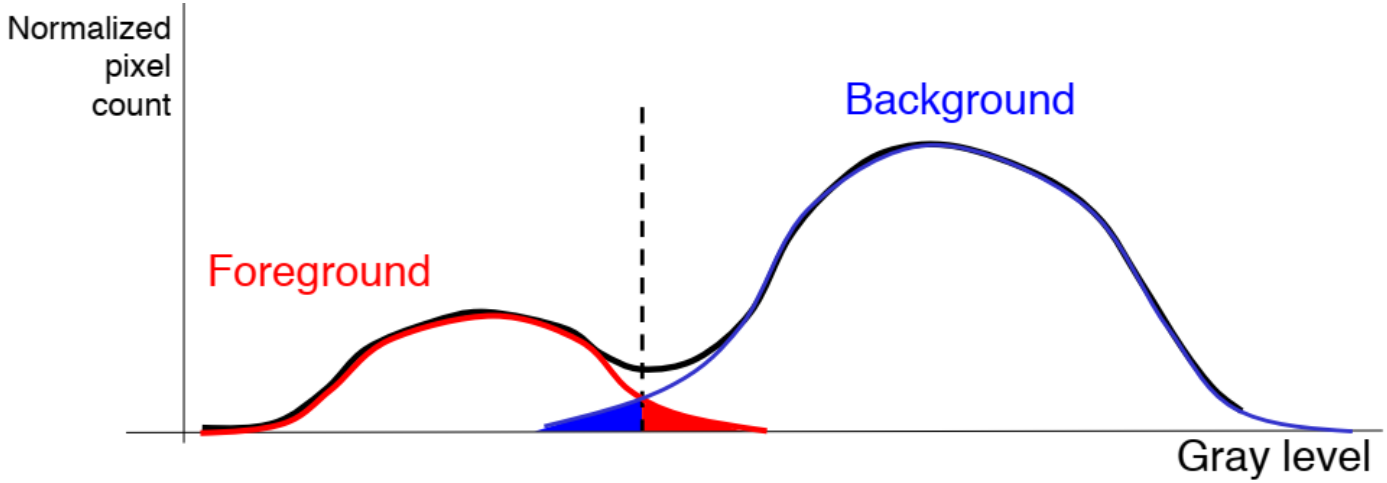
Masked Images

$f[x, y] \cdot m[x, y]$

Pixel-wise multiplication

How can holes be filled?

How to choose the threshold?



- Unsupervised thresholding
  - Find threshold T that minimizes within-class variance

$$\sigma^2_{within}(T) = \frac{N_{Fgrnd}(T)}{N}\sigma^2_{Fgrnd}(T) + \frac{N_{Bgrnd}(T)}{N}\sigma^2_{Bgrnd}(T)$$

Maximize between-class variance

$$\sigma^2_{between}(T) = \sigma^2 - \sigma^2_{within}(T)$$

$$= \left(\frac{1}{N}\sum_{x,y} f^2[x,y] - \mu^2\right) - \frac{N_{Fgrd}}{N}\left(\frac{1}{N_{Fgrd}}\sum_{x,y\in Fgrnd} f^2[x,y] - \mu^2_{Fgrnd}\right) - \frac{N_{Bgrnd}}{N}\left(\frac{1}{N_{Bgrnd}}\sum_{x,y\in Bgrnd} f^2[x,y] - \mu^2_{Bgrnd}\right)$$

$$= -\mu^2 + \frac{N_{Fgrnd}}{N}\mu^2_{Fgrnd} + \frac{N_{Bgrnd}}{N}\mu^2_{Bgrnd} = \frac{N_{Fgrnd}}{N}\left(\mu_{Fgrnd} - \mu\right)^2 + \frac{N_{Bgrnd}}{N}\left(\mu_{Bgrnd} - \mu\right)^2$$

$$= \frac{N_{Fgrnd}(T)\cdot N_{Bgrnd}(T)}{N^2}\left(\mu_{Fgrnd}(T) - \mu_{Bgrnd}(T)\right)^2$$

[Otsu, 1979]

Unsupervised thresholding

- Algorithm: Search for threshold T to maximize

$$\sigma^2_{between}(T) = \frac{N_{Fgrnd}(T) \cdot N_{Bgrnd}(T)}{N^2} \left(\mu_{Fgrnd}(T) - \mu_{Bgrnd}(T) \right)^2$$

- Useful recursion for sweeping T across histogram:

$$N_{Fgrnd}(T+1) = N_{Fgrnd}(T) + n_T$$

$$N_{Bgrnd}(T+1) = N_{Bgrnd}(T) - n_T$$

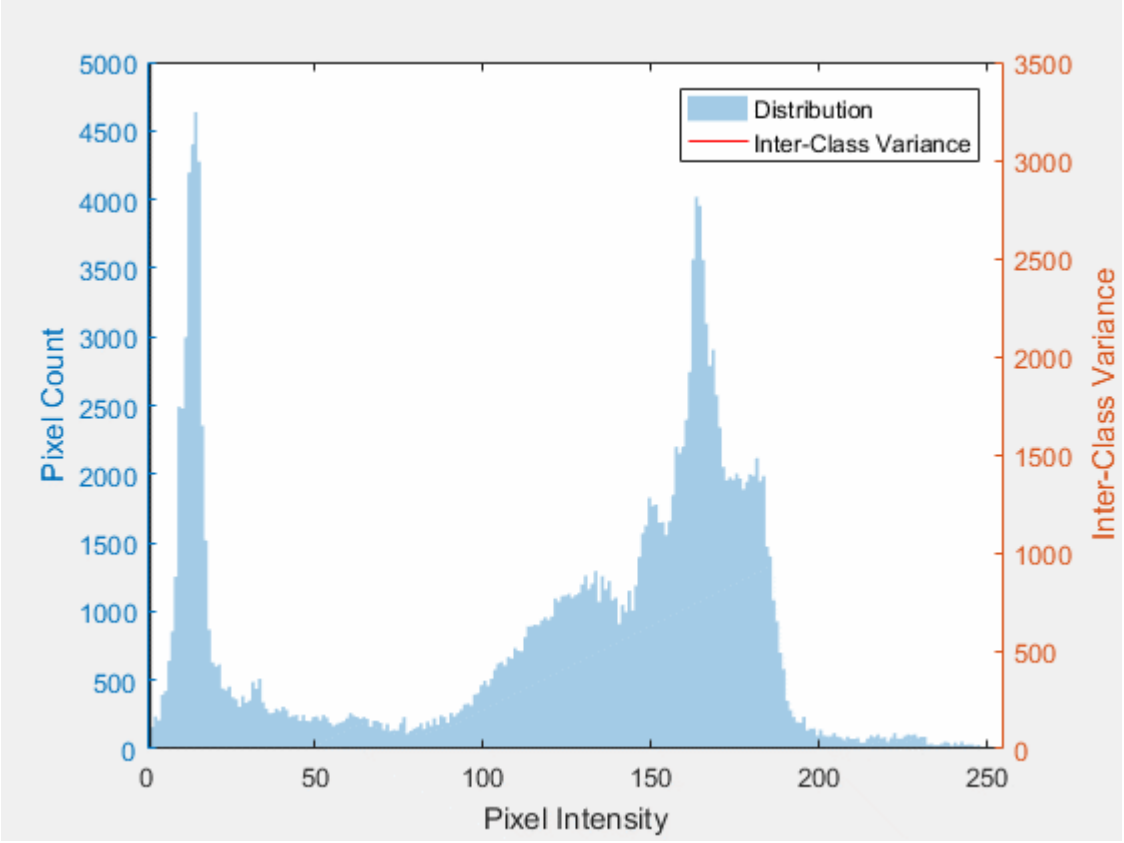
$$\mu_{Fgrnd}(T+1) = \frac{\mu_{Fgrnd}(T) N_{Fgrnd}(T) + n_T T}{N_{Fgrnd}(T+1)}$$

$$\mu_{Bgrnd}(T+1) = \frac{\mu_{Bgrnd}(T) N_{Bgrnd}(T) - n_T T}{N_{Fgrnd}(T+1)}$$

[Otsu, 1979]

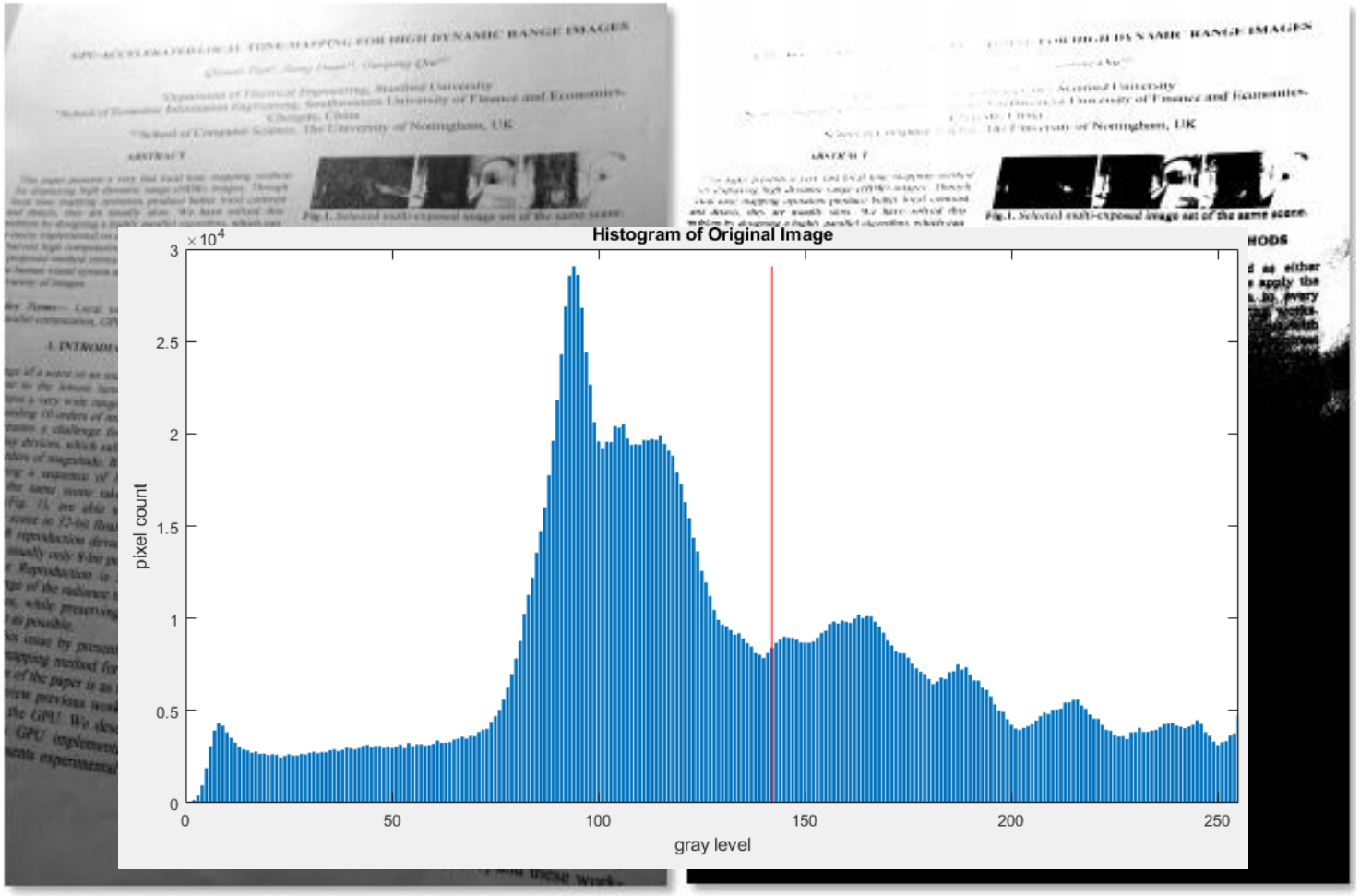
HW: Implement Otsu method

Unsupervised thresholding



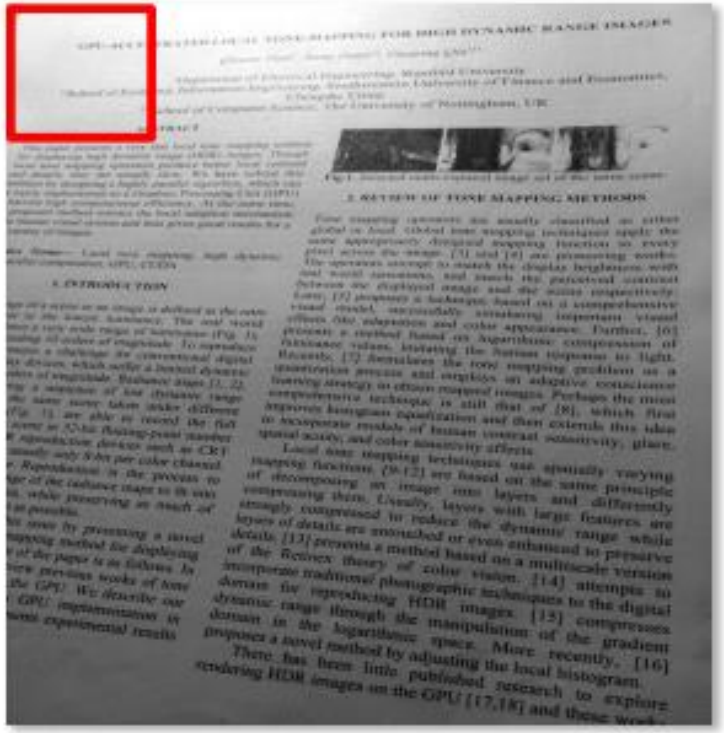
Locally Adaptive Thresholding

- A single thresholding value might not work

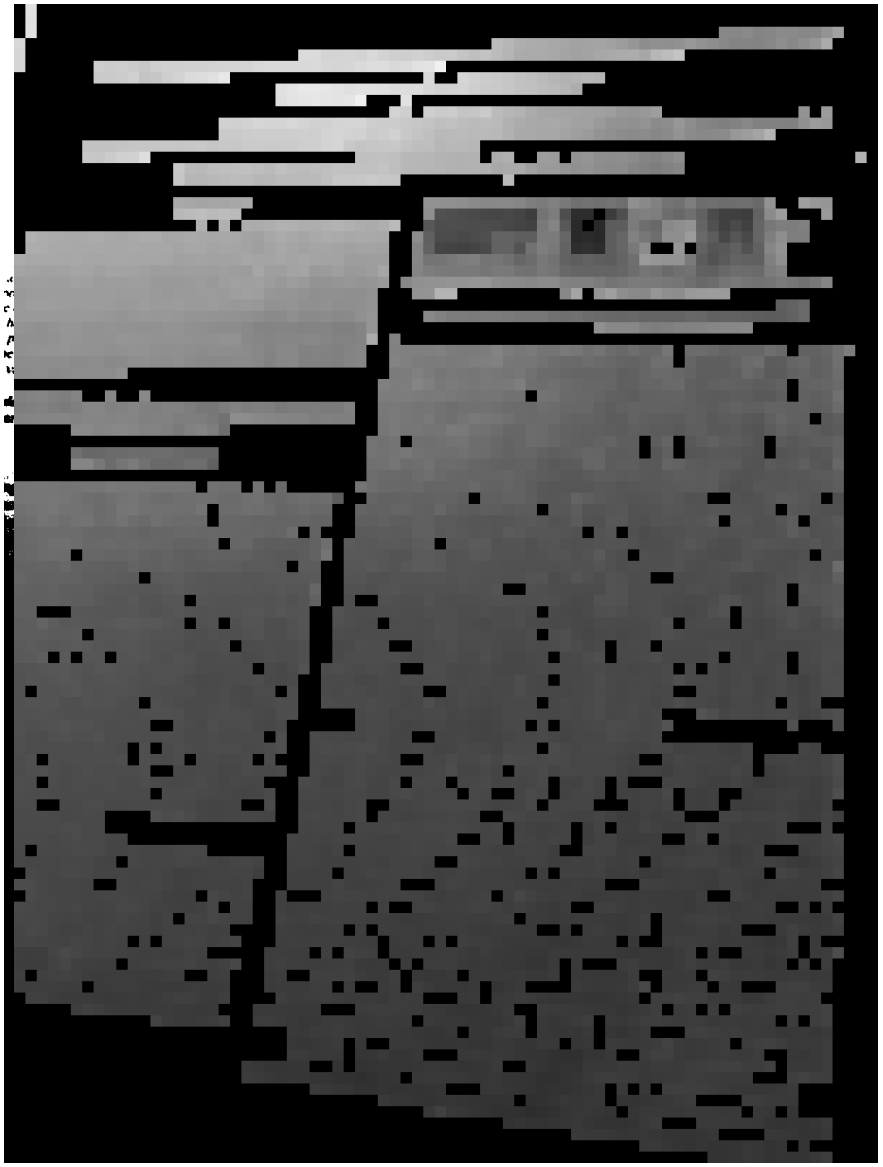


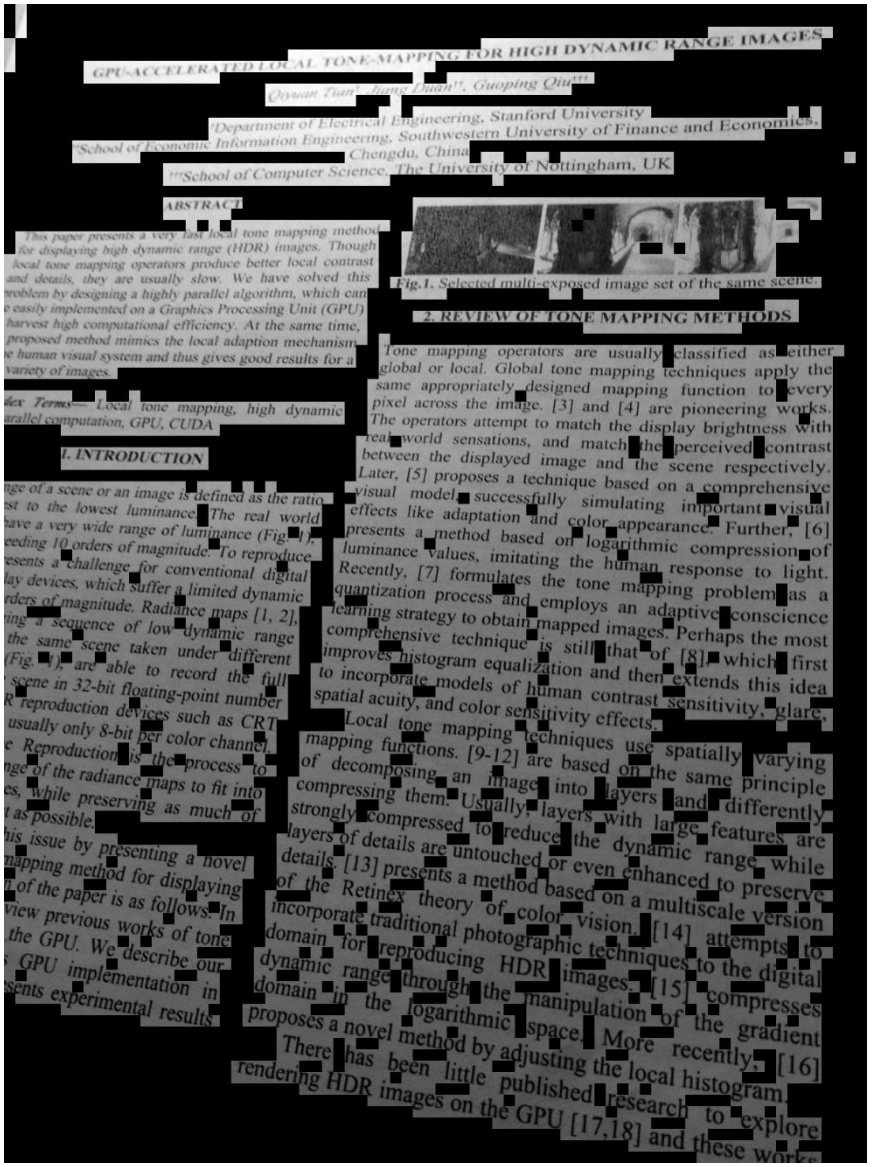
Locally Adaptive Thresholding

- Slide a window over the image
- For each window position, decide whether to perform thresholding
  - Thresholding should not be performed in uniform areas
  - Use variance or other suitable criterion
- Non-uniform areas: apply Otsu's method (based on local histogram)
- Uniform areas: classify the entire area as foreground or background based on mean value



Locally Adaptive Thresholding





GPU-ACCELERATED LOCAL TONE-MAPPING FOR HIGH DYNAMIC RANGE IMAGES

Qiyuan Tian[1], Jiang Duan[1], Guoping Qiu[1,2]

[1]Department of Electrical Engineering, Stanford University

[2]School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, China

[1,2]School of Computer Science, The University of Nottingham, UK

ABSTRACT

This paper presents a very fast local tone mapping method for displaying high dynamic range (HDR) images. Though local tone mapping operators produce better local contrast and details, they are usually slow. We have solved this problem by designing a highly parallel algorithm, which can be easily implemented on a Graphics Processing Unit (GPU) to harvest high computational efficiency. At the same time, the proposed method mimics the local adaption mechanism of the human visual system and thus gives good results for a variety of images.

Index Terms— Local tone mapping, high dynamic range, parallel computation, GPU, CUDA

1. INTRODUCTION

Fig.1. Selected multi-exposed image set of the same scene.

2. REVIEW OF TONE MAPPING METHODS

Tone mapping operators are usually classified as either global or local. Global tone mapping techniques apply the same appropriately designed mapping function to every pixel across the image. [3] and [4] are pioneering works. The operators attempt to match the display brightness with real-world sensations, and match the perceived contrast between the displayed image and the scene respectively. Later, [5] proposes a technique based on a comprehensive visual model, successfully simulating important visual effects like adaptation and color appearance. Further, [6] presents a method based on logarithmic compression of luminance values, imitating the human response to light. Recently, [7] formulates the tone mapping problem as a quantization process and employs an adaptive conscience learning strategy to obtain mapped images. Perhaps the most comprehensive technique is still that of [8] which first improves histogram equalization and then extends this idea to incorporate models of human contrast sensitivity, glare, spatial acuity, and color sensitivity effects.

Local tone mapping techniques use spatially varying mapping functions. [9-12] are based on the same principle of decomposing an image into layers and differently compressing them. Usually, layers with large features are strongly compressed to reduce the dynamic range while layers of details are untouched or even enhanced to preserve details. [13] presents a method based on a multiscale version of the Retinex theory of color vision. [14] attempts to incorporate traditional photographic techniques to the digital domain for reproducing HDR images. [15] compresses dynamic range through the manipulation of the gradient domain in the logarithmic space. More recently, [16] proposes a novel method by adjusting the local histogram. There has been little published research to explore rendering HDR images on the GPU [17,18] and these works

Maximally stable extremal regions

- Extremal region: any connected region in an image with all pixel values above (or below) a threshold

  Region that differ in properties, such as brightness or color, compared to surrounding regions. Almost uniform intensity

- Observations:
  - Nested extremal regions result when the threshold is successively raised
    - or lowered.
  - The nested extremal regions form a “component tree.”

- Key idea: choose thresholds θ such that the resulting bright (or dark) extremal regions are nearly constant when these thresholds are perturbed by +/-Δ

“maximally stable” extremal regions (MSER)

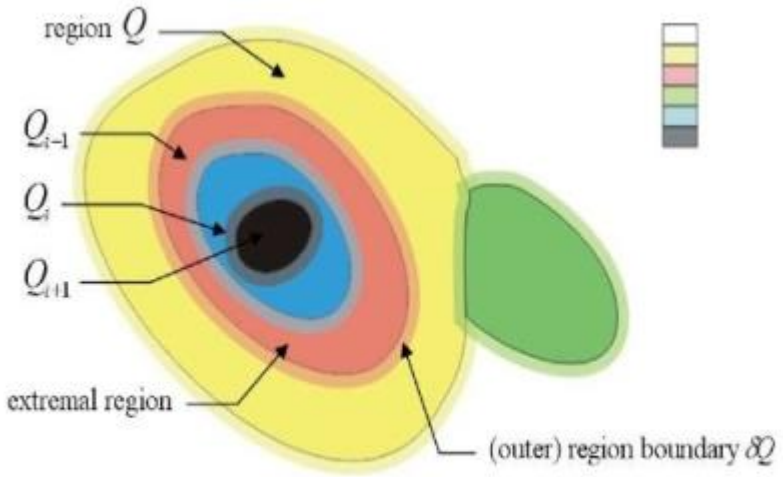
stable connected component of some gray-level sets of the image .

[Matas, Chum, Urba, Pajdla, 2002]

MSER Processing

- Step by step
  - Sweep threshold: from black to white
  - Extract connected component (extremal regions)
  - Find a threshold make extremal region “maximally stable”
  - Approximate a region with an ellipse (optional)
  - Keep those regions descriptors as features

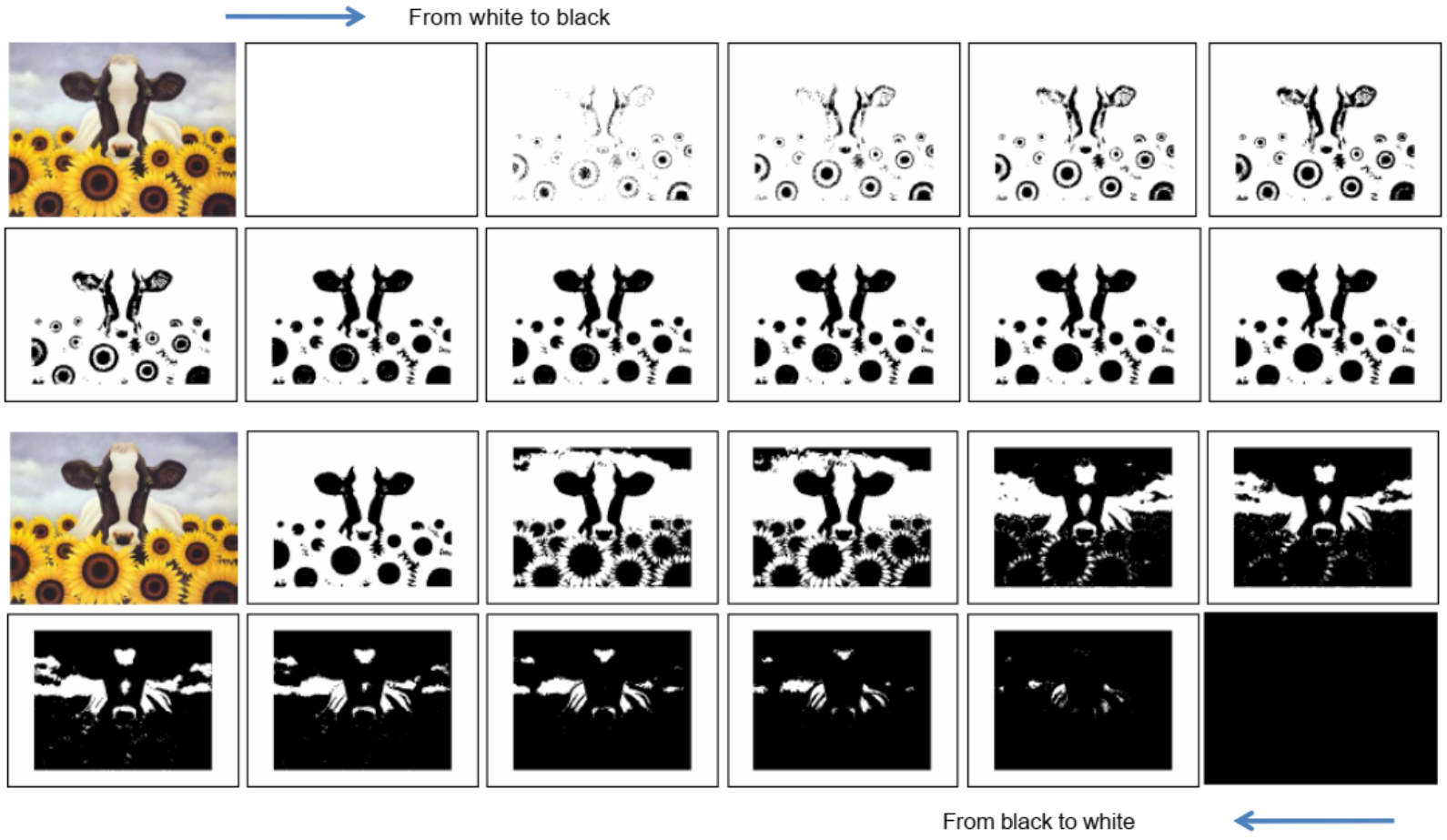
- Extremal region can be reject if
  - It is too big/mall
  - Too unstable
  - Too similar to its parent MSER



$$Q_{i^*} : i^* = \arg\min_i |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$$

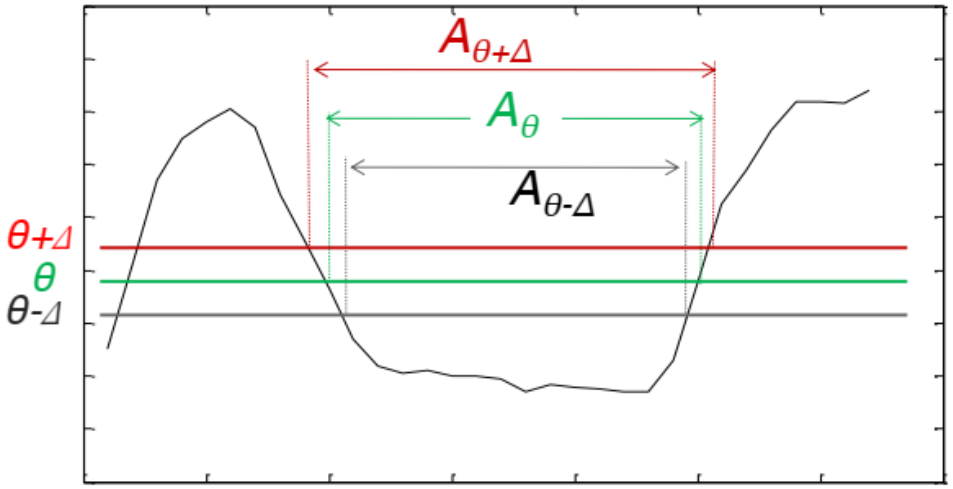
Sweeping Image threshold

- Apply a series of thresholds – one for each grayscale level.
- Threshold the image at each level to create a series of black and white images.
- One extreme will be all white, the other all black. In between, blobs grow and merge



MSER Processing

- Control how stable it is the region via Δ



Local minimum of $\left|\frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_{\theta}}\right| \rightarrow$ MSER

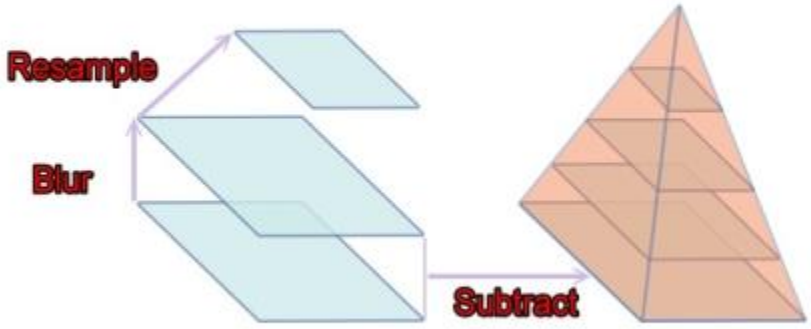
[Matas, Chum, Urba, Pajdla, 2002]

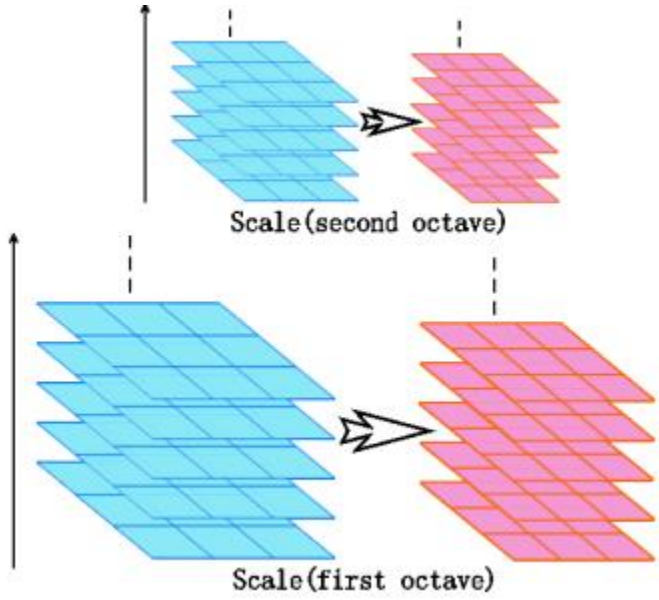
A_θ the area of the extremal region

$A_{\theta-\Delta}, A_{\theta+\Delta}$ is the different in the area of two regions with $\pm\Delta$ level

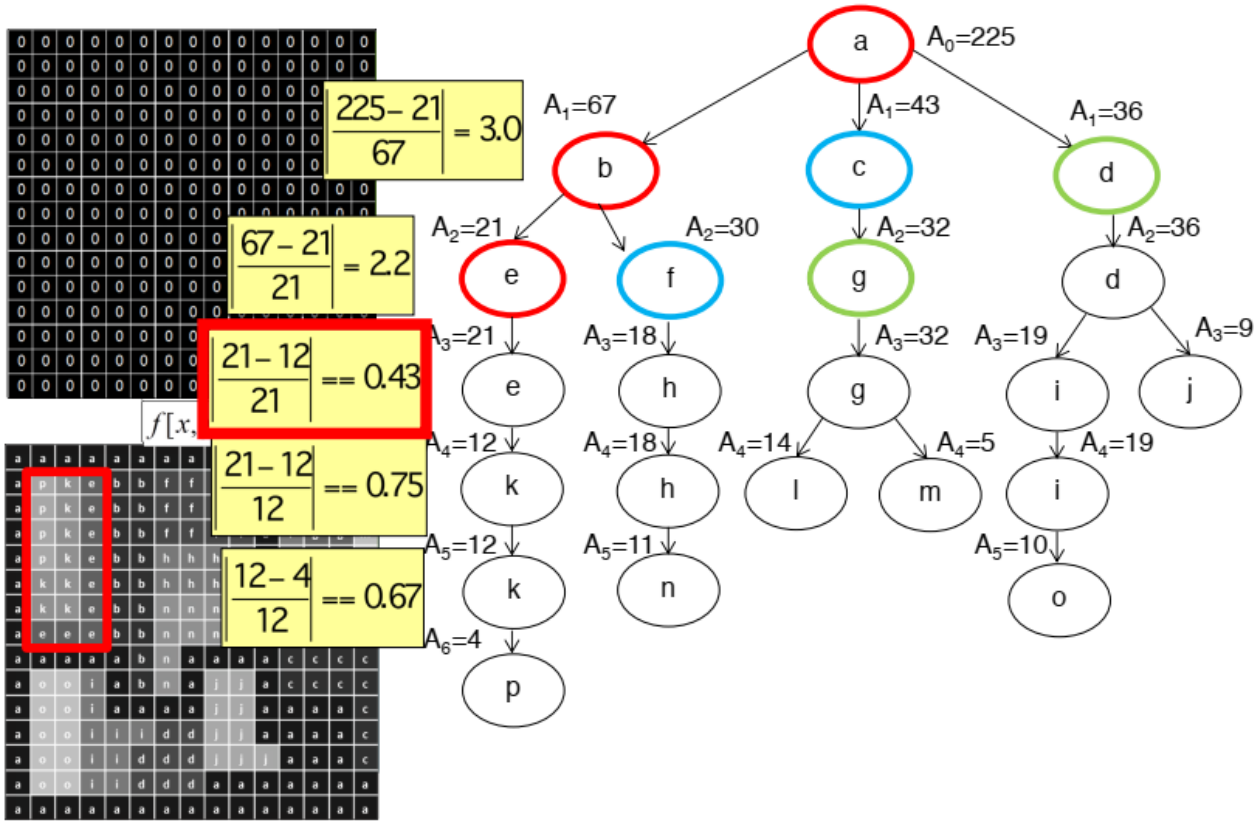
Multi-resolution MSER

- Step1 : Construct a scale pyramid with one octave between scales
- Step2 : Detect MSERs separately at each resolution
- Step3 : Duplicate MSERs are removed by eliminating fine scale MSERs with similar locations and sizes as MSERs detected at the next coarser scale





Component tree of an image



Local minima of sequence

$$\left| \frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_\theta} \right|$$

$\theta = \Delta,\ \Delta + 1,\ \ldots \rightarrow$ MSERs

MSER Example

Dark MSERs, Δ=15

Original image

Bright MSERs, Δ=15

Dark MSERs, Δ=15

Original image

Bright MSERs, Δ=15

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 8$

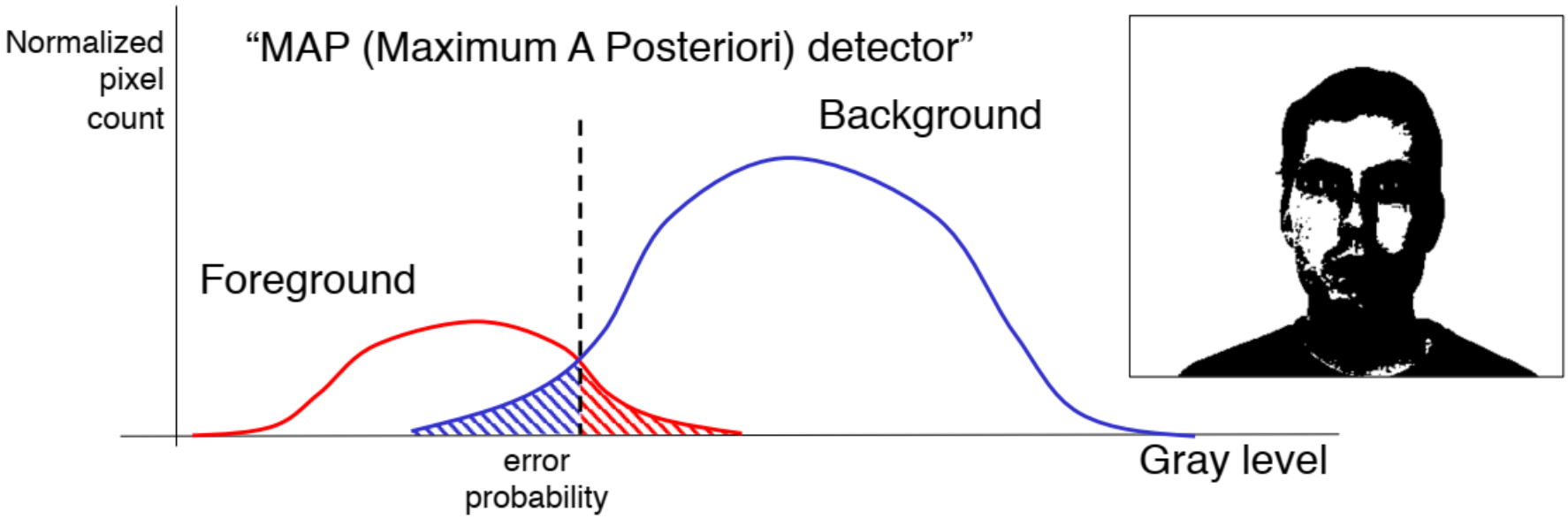
Level Set

MSER Properties

- MSER perform well on
  - images containing homogeneous regions with distinctive boundaries.
  - small regions
- MSER doesn't work well with images with any motion blur
- Multi-resolution MSER provides better
  - Robustness to large scale changes and blurred images
  - Improves matching performance over large scale changes & for blurred images
- Good repeatability
- Affine invariant
- A smart implementation makes it one of the fastest region detectors

http://www.vlfeat.org/overview/mser.html

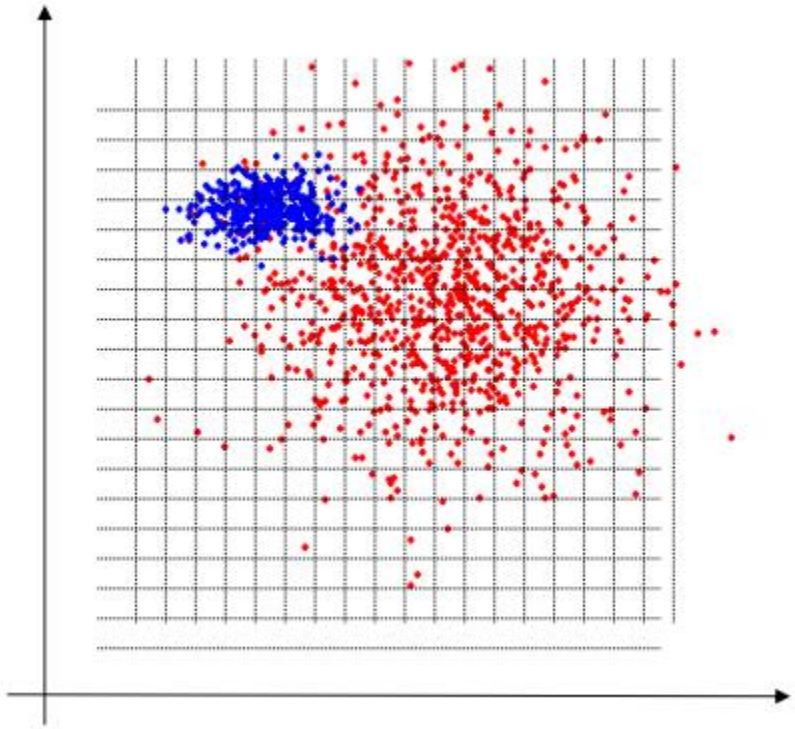
Supervised Thresholding



If errors BG➔FG and FG➔BG are associated with different costs:
"Bayes minimum risk detector" is optimal.

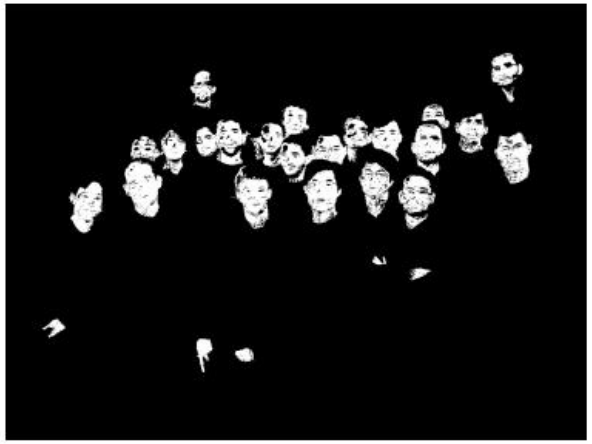
Multidimensional MAP detector

- Training
  - Provide labelled set of training data
  - Subdivide n-dimensional space into small bins
  - Count frequency of occurrence for each b in and class in training set, label bin with most probable class
  - (Propagate class labels to empty bins)
- For test data: identify bin, look up the mos t probable class

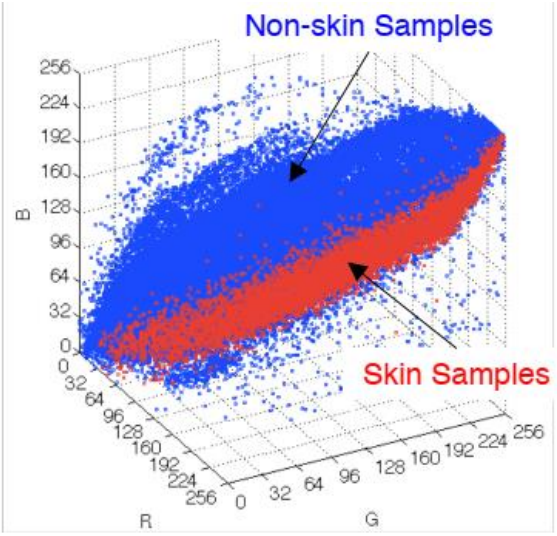


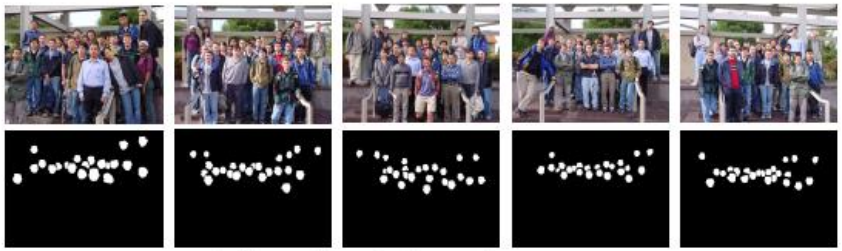
MAP detector in RGB-space

Original image



Skin color detector





Five training images