ECE5554 – Computer Vision
Lecture 7a– Basic Segmentation

Creed Jones, PhD

Course update

- HW3 is due tonight – July 27 at 11:59 PM!

- HW4 has been posted - due August 3

- SPOT surveys on this course will open soon
  - open from August 6 through August 12
  - participation is completely anonymous and completely voluntary
  - I would appreciate your responses – especially comments that I can act on!

- Lecture 10 on Monday, August 8 will be asynchronous
  - No synchronous class session
  - I will be traveling
  - There will be three pre-recorded lectures, watch at your convenience
  - I will look for questions in Piazza

Final Exam will be Thursday, August 11

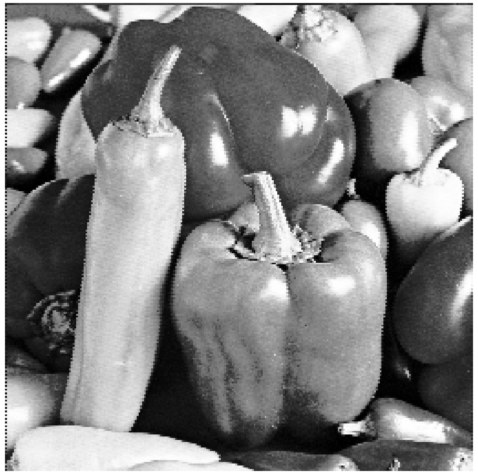
- The exam will be a collection of questions similar to the quiz questions, plus a few additional questions (may be a short calculation, a question requiring a few sentences in response, etc)

- There will be a two-hour window but I am designing the exam to require one hour or less
- I need a single two-hour window for the entire class
- Everyone email me this week and tell me all of the following time slots that work for you:
  - Thursday, August 11, 7 PM to 9 PM Eastern time
  - Thursday, August 11, 8 PM to 10 PM Eastern time
  - Thursday, August 11, 9 PM to 11 PM Eastern time
  - Thursday, August 11, 10 PM to 12 midnight Eastern time

Today’s Objectives

- Concept of Image Segmentation
- Global Thresholding
  - Laplacian thresholding
  - Kittler-Illingworth and Otsu methods
  - Histogram-based
- Local Thresholding
  - Bernsen’s method
  - Niblack’s method

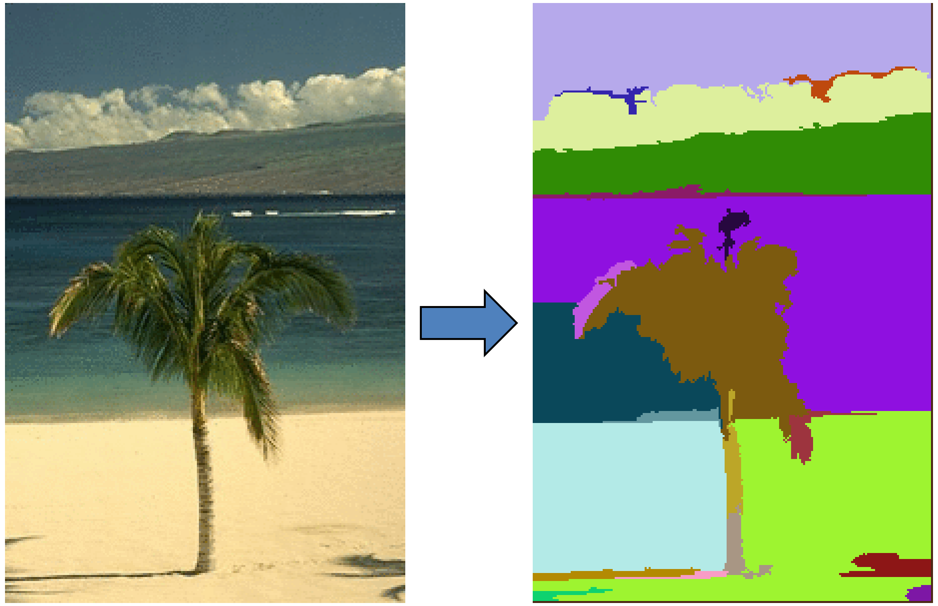
Image segmentation refers to the *partitioning* of an image into *meaningful* regions with respect to a particular application



- Segmentation may be based on gray values, color, texture, depth, motion, ...

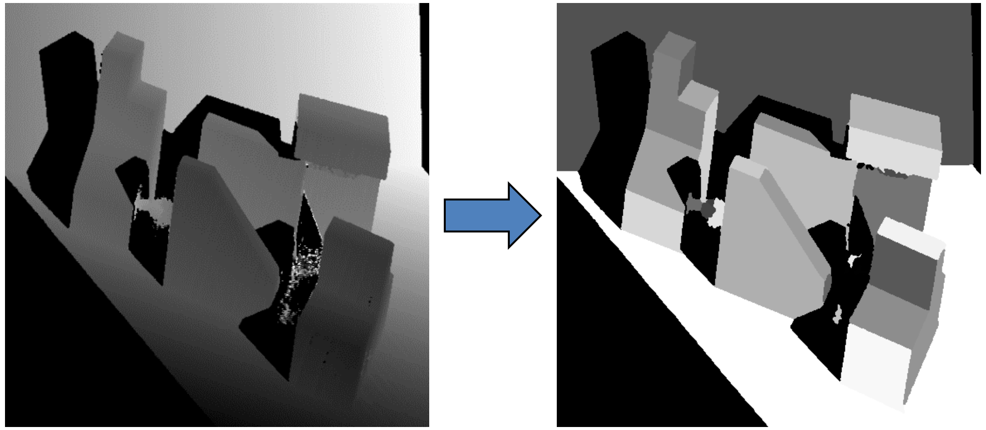
Intensity/color segmentation

(source: Felzenszwalb and Huttenlocher)



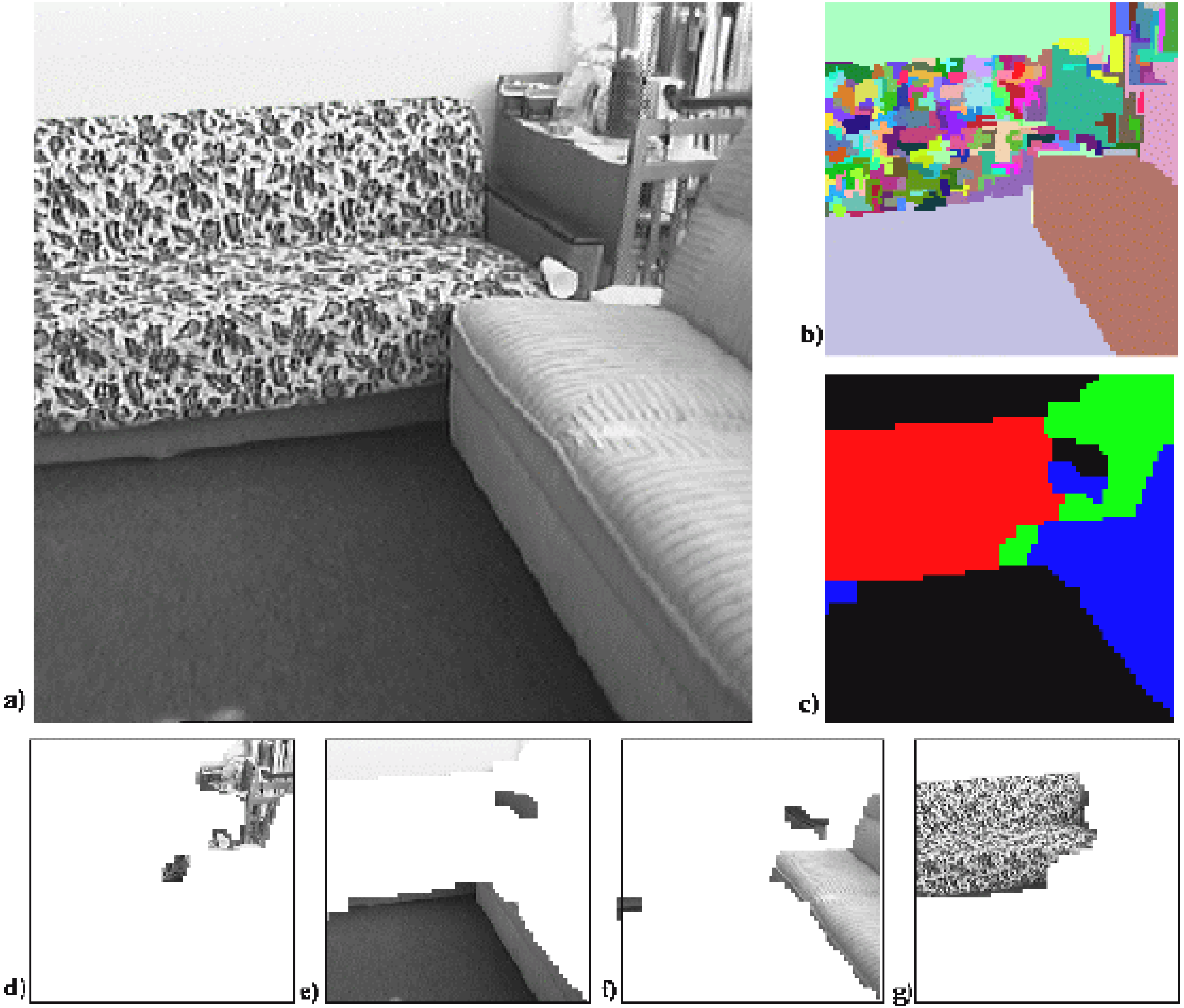
Range image segmentation

(source: Spann)



Scene understanding

(source: Spann)



Why study segmentation and grouping?

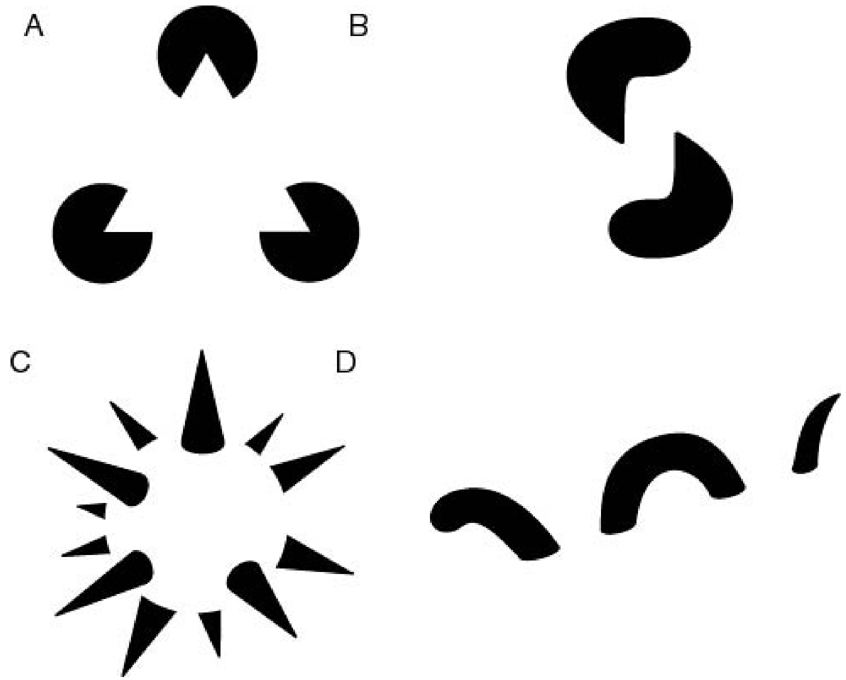
- To gain insights into human perception
- Important intermediate step toward higher-level computer vision
  – E.g., separation of an object from background will assist in object recognition
  – E.g., distinguishing road or hallway from rest of image will aid in path planning
- Many practical applications
  – Can assist in image search
  – Identifying objects in an image as precursor to size/shape measurements
  – Biomedical applications
  – Image/video compression

Inspiration from psychology

- The Gestalt school: Grouping is key to visual perception

  "The whole is greater than the sum of its parts"

subjective contours



Humans have an interesting tendency to explain by occlusion

http://en.wikipedia.org/wiki/Gestalt_psychology

Similarity (in shape, texture, …)

(slide credit: Grauman)

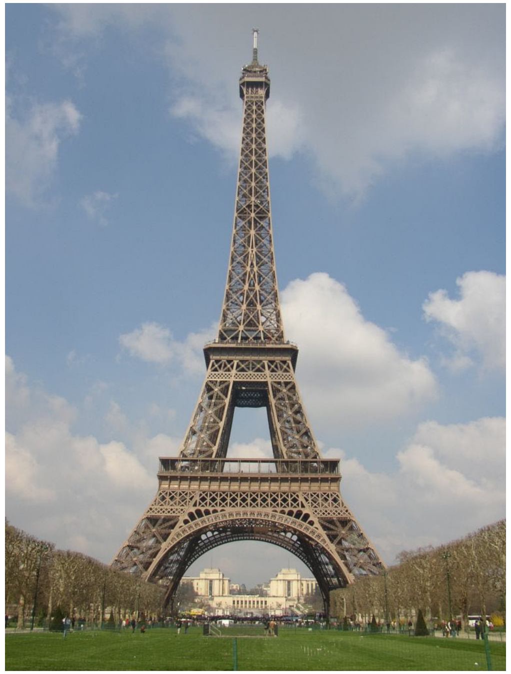
http://chicagoist.com/attachments/chicagoist_alicia/GEESE.jpg, http://wwwdelivery.superstock.com/WI/223/1532/PreviewComp/SuperStock_1532R-0831.jpg

Proximity

http://www.capital.edu/Resources/Images/outside6_035.jpg

(slide credit: Grauman)

Symmetry



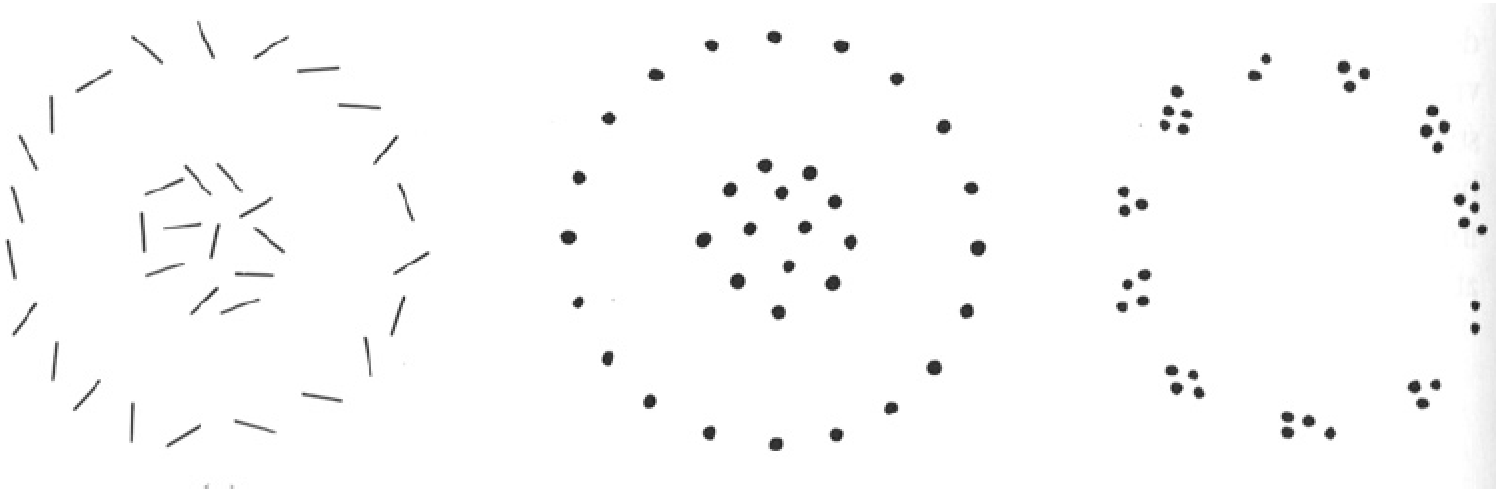
(slide credit: Grauman)

http://seedmagazine.com/news/2006/10/beauty_is_in_the_processingtim.php

Common fate

Image credit: Arthus-Bertrand (via F. Durand)

(slide credit: Grauman)



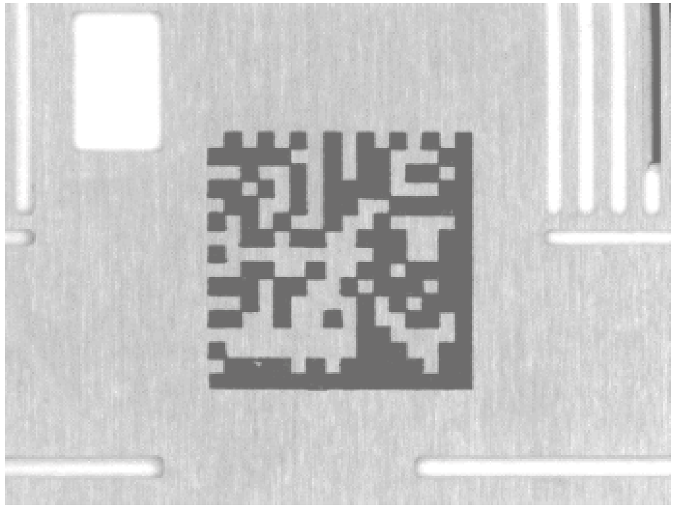
Should segmentation be done *top-down* or *bottom-up*?

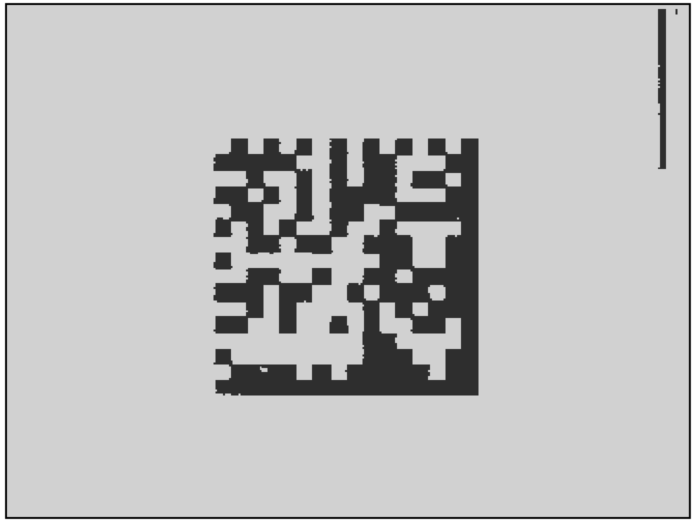
- Top-down vs. bottom-up segmentation
  - Top-down: pixels belong together because they are from the same object
  - Bottom-up: pixels belong together because they look similar

- In either case, it's hard to measure success
  - Depends on the application!

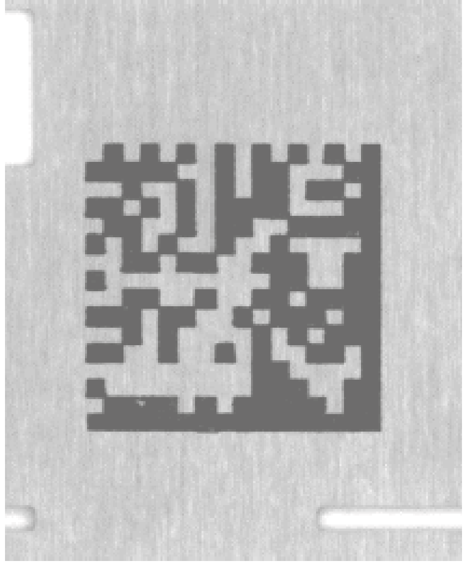
The simplest possible example of segmentation is Binarization based on pixel intensity

- We choose a range of gray levels to be mapped to WHITE, all others become BLACK
  - Most common method is to choose a threshold – all pixels higher than this become 255, others become 0 – called “Global Thresholding”



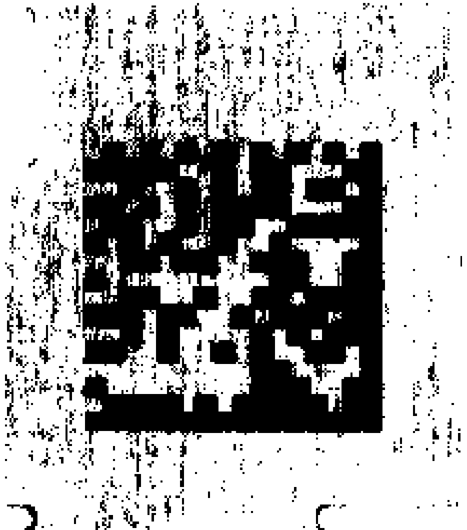


The choice of a threshold value is critical – lower threshold means that more pixels will be assigned as “foreground” or white



128

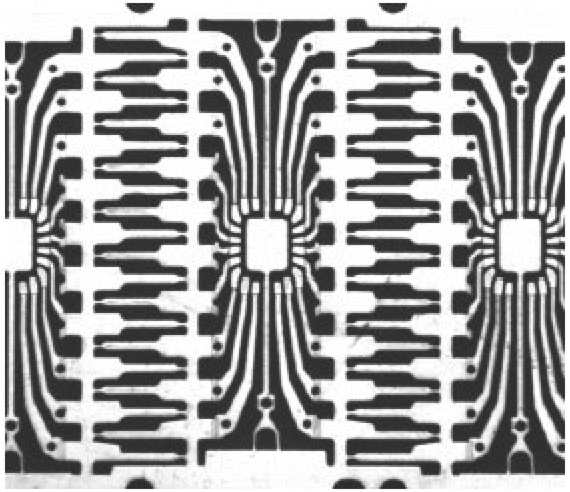
140

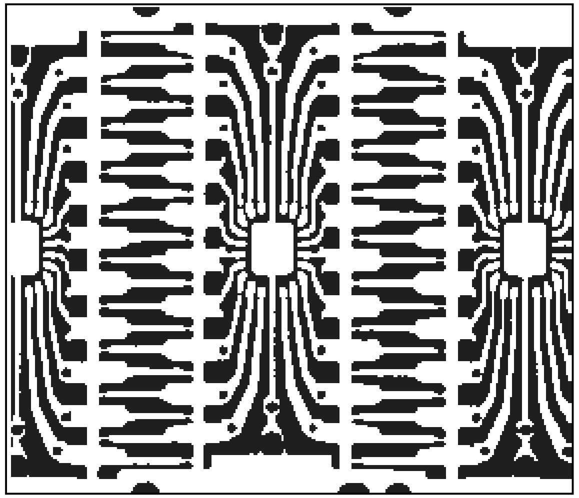


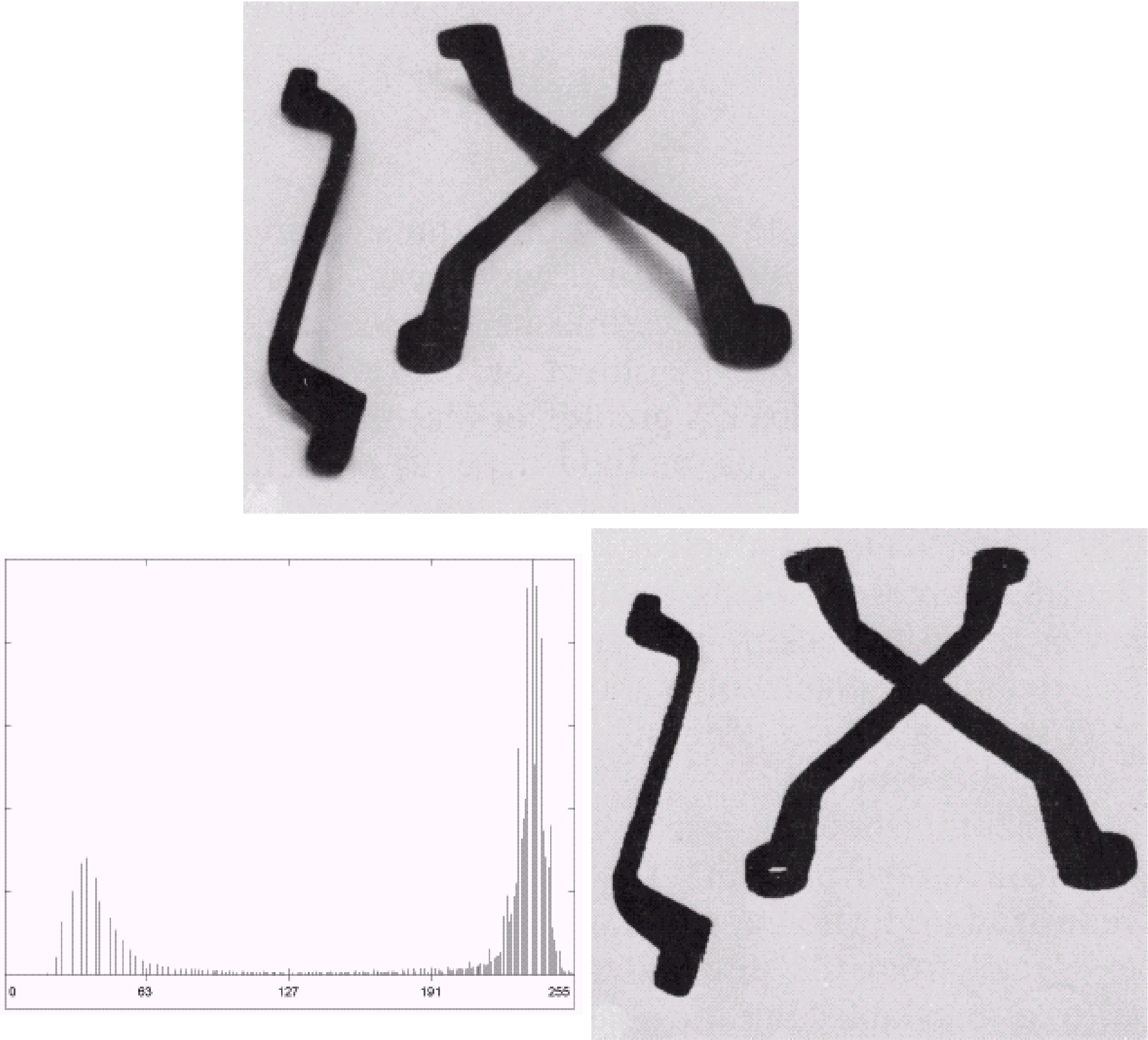
155

In a truly binarized image, the only values are 0 and 1; We often represent this as 0 and 255, so the "background" is total black and the foreground is total white

- To see if any of the legs in the leadframe are missing, total the number of white pixels – if too low, something is missing
- Note this won't tell which leg might be missing
- There are other problems too…



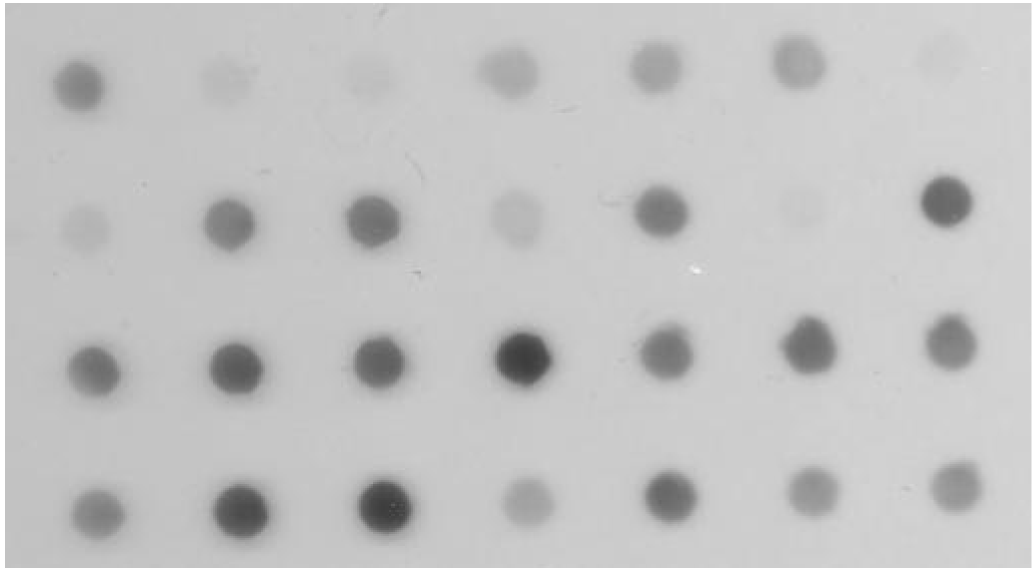


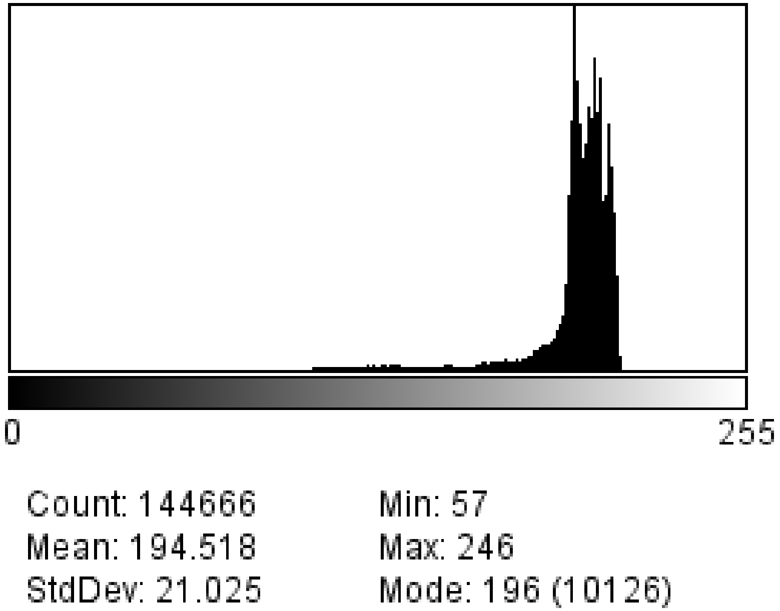


a
b c

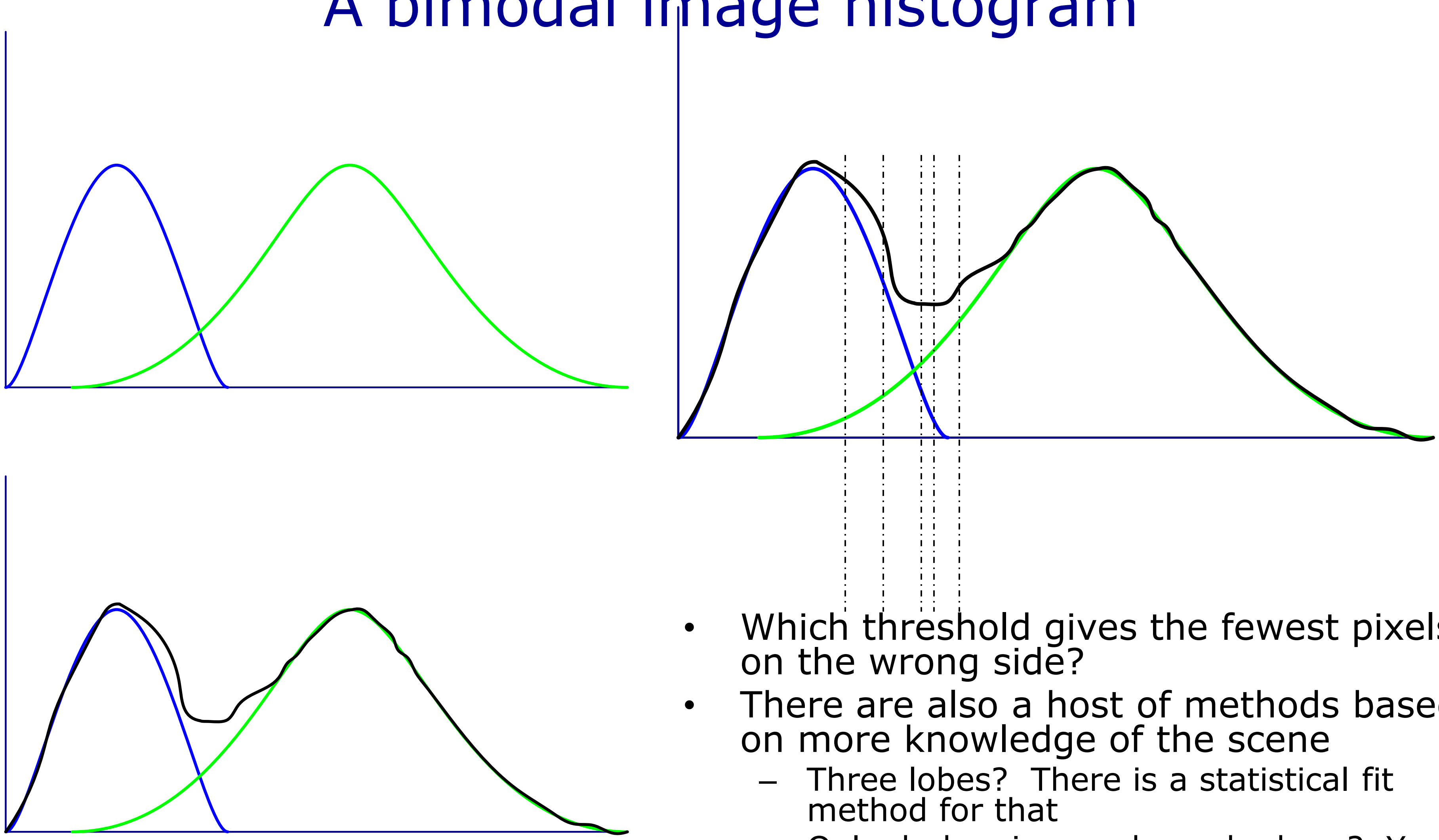
FIGURE 10.28 (a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

Global thresholding has two challenges – how do we choose a threshold, and what if one threshold won't separate all of the foreground objects?





A bimodal image histogram



- Which threshold gives the fewest pixels on the wrong side?
- There are also a host of methods based on more knowledge of the scene
  - Three lobes? There is a statistical fit method for that
  - Only dark noise, such as shadows? You can derive a method for that situation.

Why can’t we find a threshold?

- Perhaps the scene really has more than two clusters of gray levels (not a bimodal histogram)
- Perhaps there is noise in the image, and while we can separate the foreground from the background, it is not possible to find T such that:

  foreground – noise > T

  AND

  background + noise < T

  Noise can be random, or have some pattern (like an illumination gradient)

Multi-modal histogram

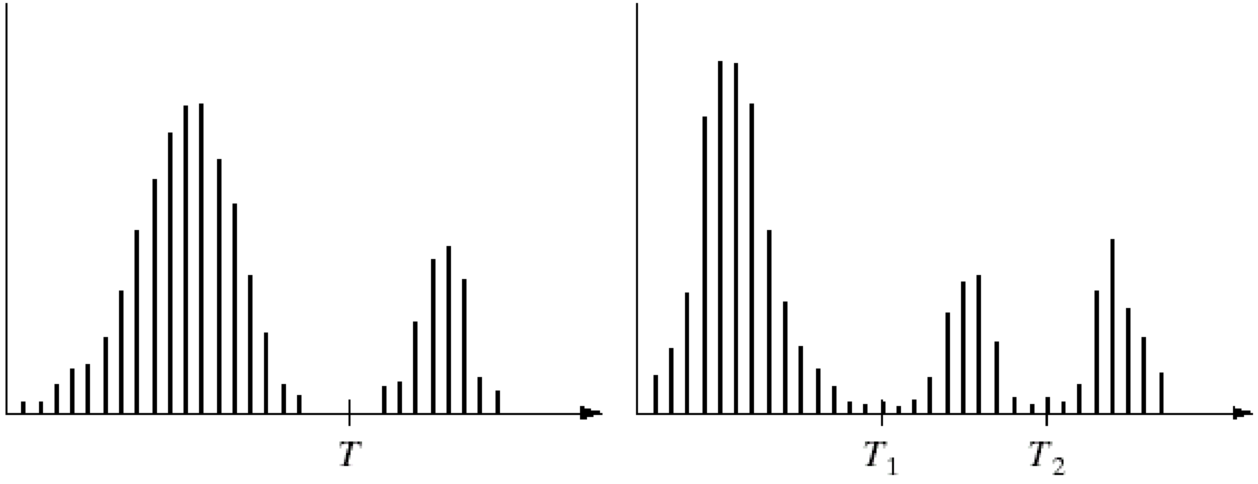
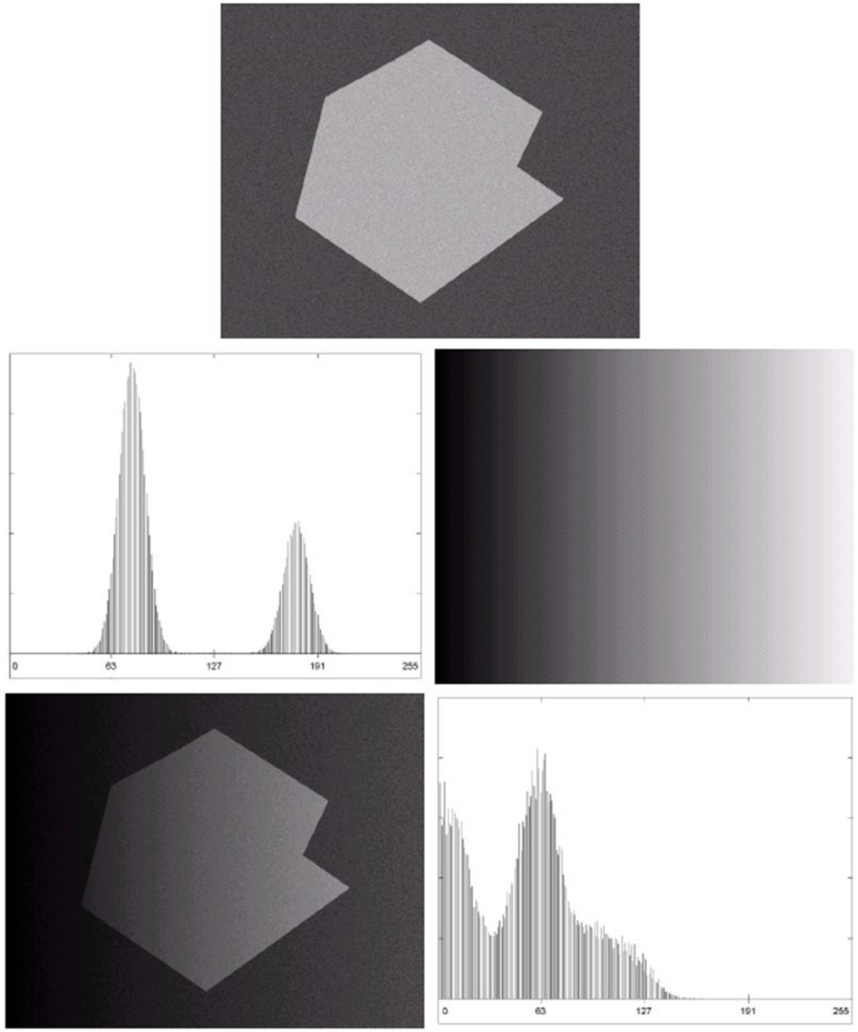


FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

- We may assume that the three lobes represent three scene regions
  - perhaps a white and a gray object on black background
- So we need to decide which objects should be separated from which

Computation of a global threshold (suitable for the entire image) is possible but may involve compromise

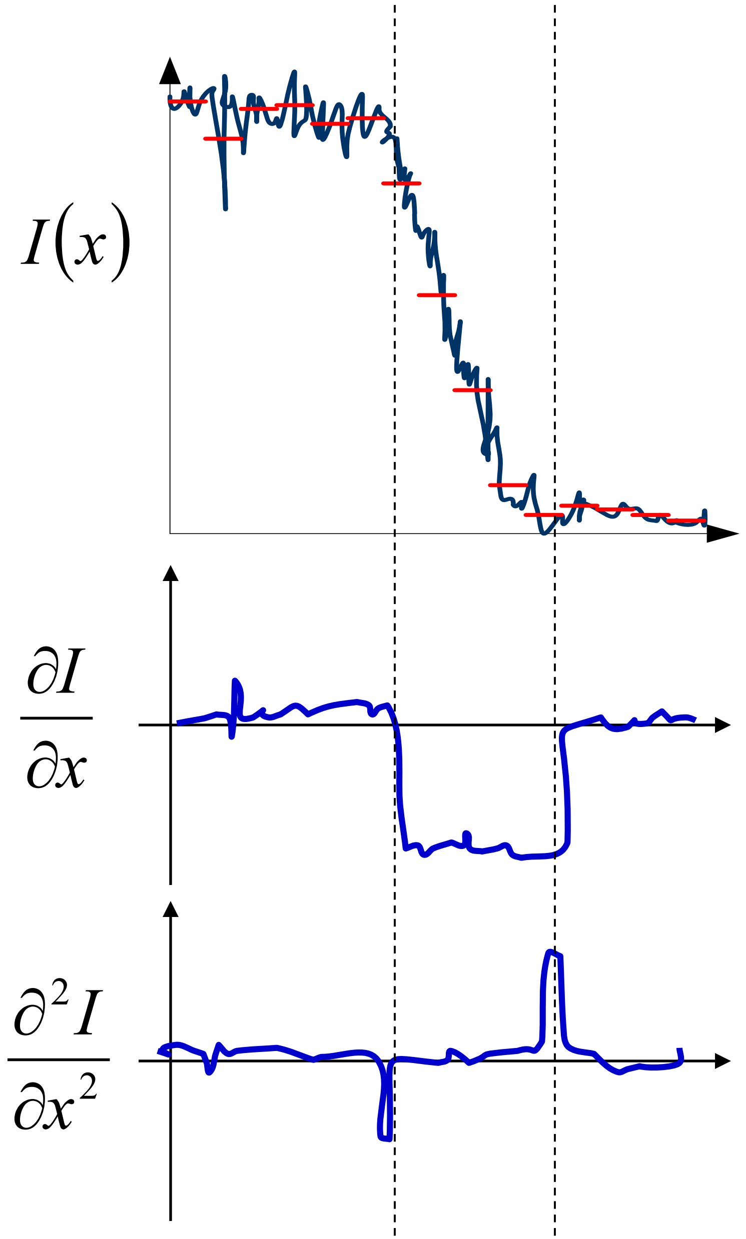
- As we have seen, if we can choose an appropriate gray-level to divide the foreground from the background, then binarization is simple
  - Done globally or in an adaptive (local) manner
- Let's examine some methods to choose a threshold in a more sophisticated manner, based on the image content
  - We will see that more sophistication will call for us to make some assumptions



a
b c
d e

FIGURE 10.27 (a) Computer generated reflectance function. (b) Histogram of reflectance function. (c) Computer generated illumination function. (d) Product of (a) and (c). (e) Histogram of product image.

- Let's examine more closely the model of an image edge…
- The 2nd derivative is large positive at the "bottom" of an edge and large negative at the "top" of an edge
- So, large positive values of the Laplacian will occur at the bottoms of edges…
- and large negative values of the Laplacian will occur at the tops of edges



Laplacian-based thresholding finds the most common gray level at which edges occur

- Form a histogram of the Laplacian magnitude for the whole image
- Choose t at the top n% (10?) in this histogram
- For each point in the image –
  - If the magnitude of the Laplacian is above t, add the original gray level at that point to histogram H1
  - If the magnitude of the Laplacian is below -t, add the original gray level at that point to histogram H2
- H1 is now a histogram of the gray levels at the "bottom" of the edges, and H2 is a histogram of the gray levels at the "top" of the edges
- Use some application-tailored method to choose a threshold – a common one is the average of the medians of the two histograms
- (Weszka 1974)

When might we use Laplacian thresholding?

- Images should be truly bimodal
- No severe illumination gradients
  - After all, this method only finds one threshold for the area it operates on!
- This method can be very good for extremely noisy images, or images where the edges are sometimes weak or broken
- The underlying assumption: all significant edges in the ROI have a similar profile and intensity levels – top and bottom

The Kittler-Illingworth and Otsu methods assume that the image is essentially binary and finds the optimum threshold from statistical analysis of the histogram

- Kittler and Illingworth assumed that the histogram is composed of two lobes, and that they are (because of noise and other artifacts) Gaussian in shape
  - with different means and variances
- Otsu simplified their method
- We want to choose a threshold that optimizes a specific “cost function”, or a function that measures the effectiveness of reducing crossover between the foreground and background portions of the histogram

Otsu's method

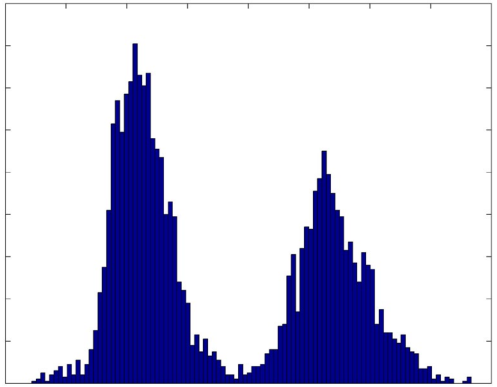
- Find the threshold t that maximizes…

$$O(t) = P_1(t)P_2(t)[\mu_2(t) - \mu_1(t)]^2$$

- where:

$$P_1(t) = \frac{\sum_{i=0}^{t-1} H(i)}{\sum_{i=0}^{N-1} H(i)} \qquad P_2(t) = \frac{\sum_{i=t}^{N-1} H(i)}{\sum_{i=0}^{N-1} H(i)}$$

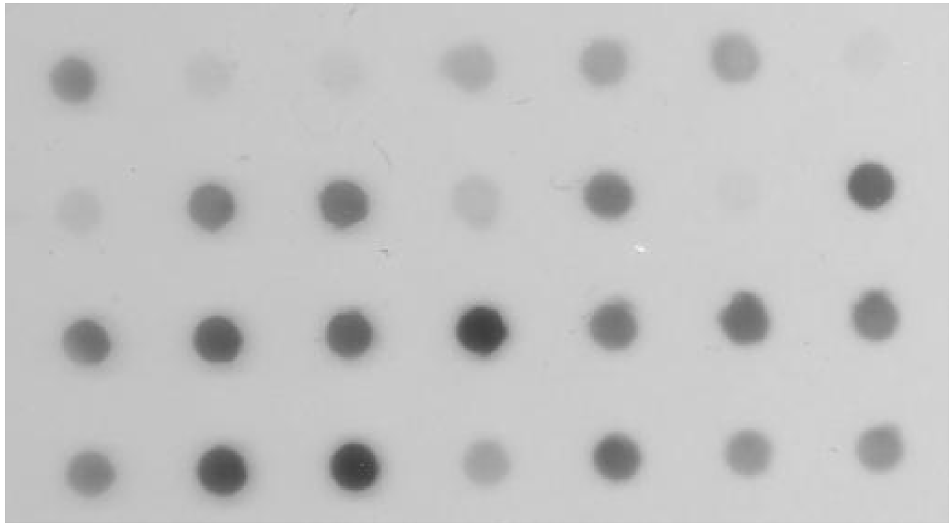
$$\mu_1(t) = \frac{1}{P_1(t)}\sum_{i=0}^{t-1} iH(i) \qquad \mu_2(t) = \frac{1}{P_2(t)}\sum_{i=t}^{N-1} iH(i)$$

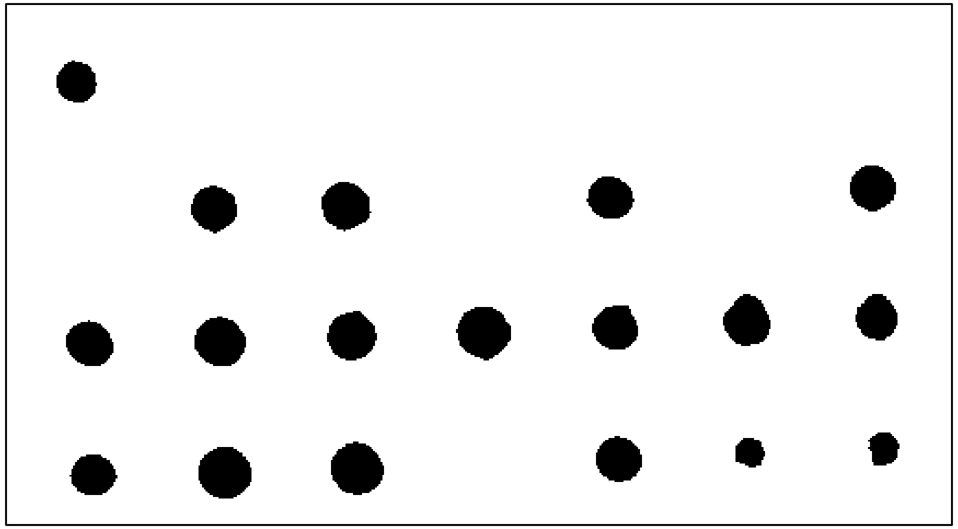


```python
def getOtsu(img, H, nbins):
     # Otsu's method minimizes the intraclass variance on the histogram
    P1 = 0
    P2 = 0
    g1 = 0
    g2 = 0
    maxcoeff = 0

    for index in range(nbins):
        P2 = P2 + H[index]
        g2 = g2 + index*H[index]

     # a big loop from 0 to nbins - where the max is found
    for index in range(nbins):
        P1 = P1 + H[index];
        P2 = P2 - H[index];
        if ( (P1 != 0) and (P2 != 0) ):
            g1 = g1 + index*H[index]
            g2 = g2 - index*H[index]
            mu1 = g1/P1
            mu2 = g2/P2
            coeff = P1 * P2 * (mu2 - mu1)**2            # the Otsu calculation
            if (coeff > maxcoeff):
                maxcoeff = coeff;
                result = index;
    return result;
```

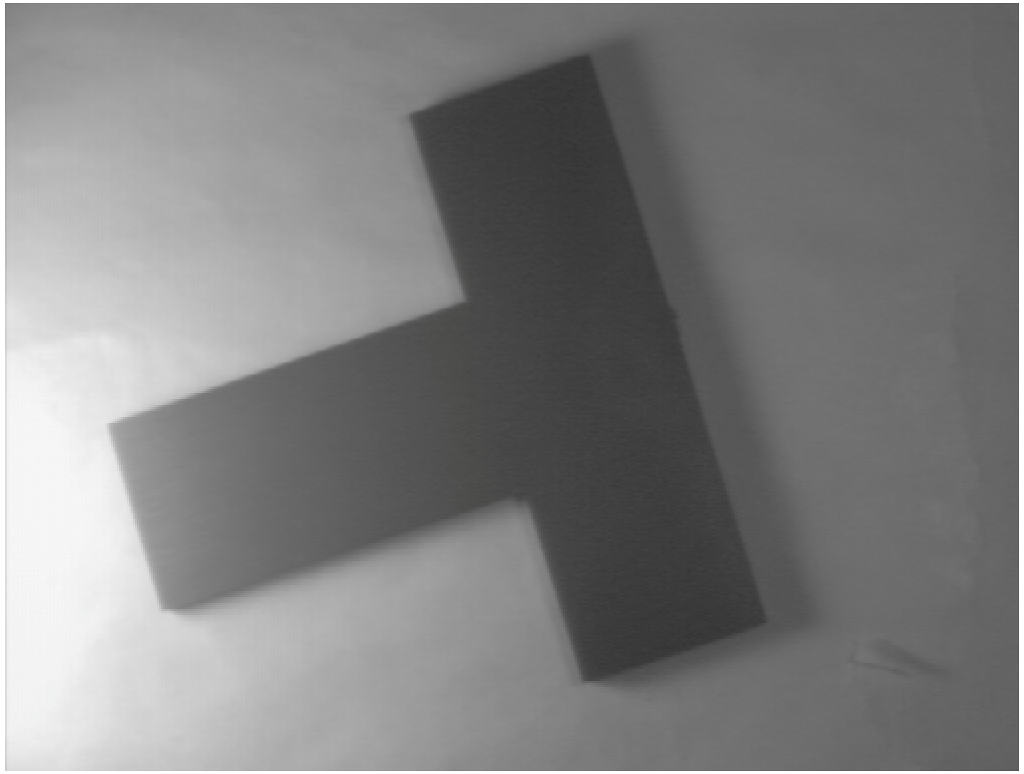
Otsu's method on the puppies image (threshold is 120)

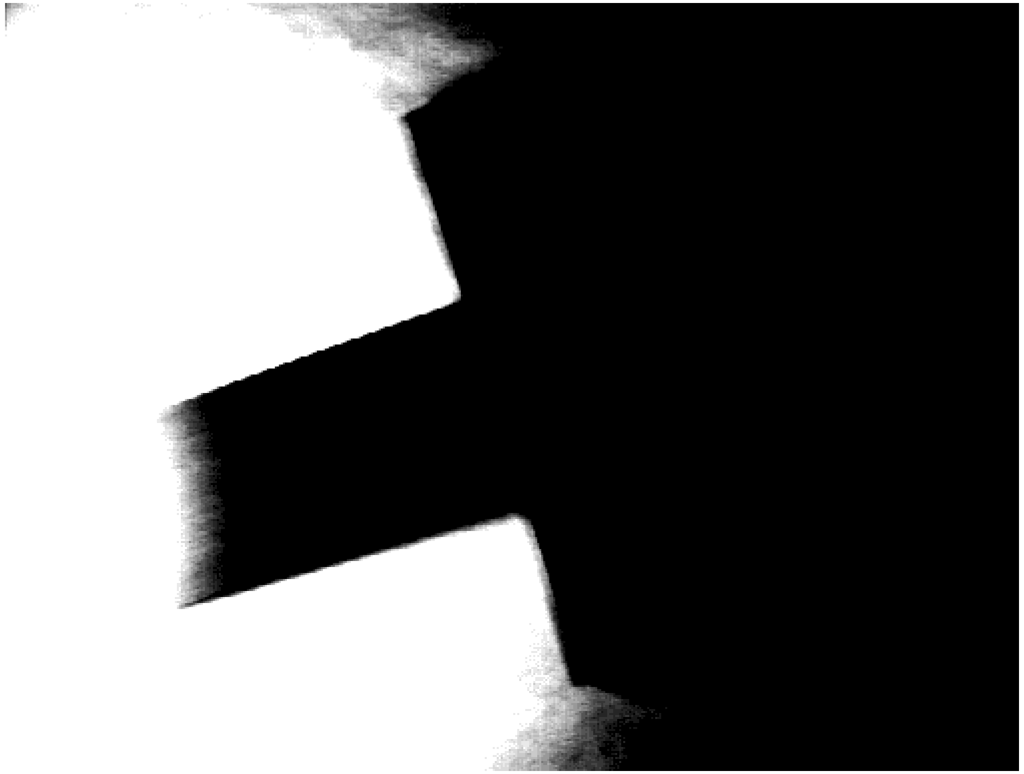
The dot blots image is very difficult to threshold – no single value will segment all of the dots (threshold given by Otsu here is 155)



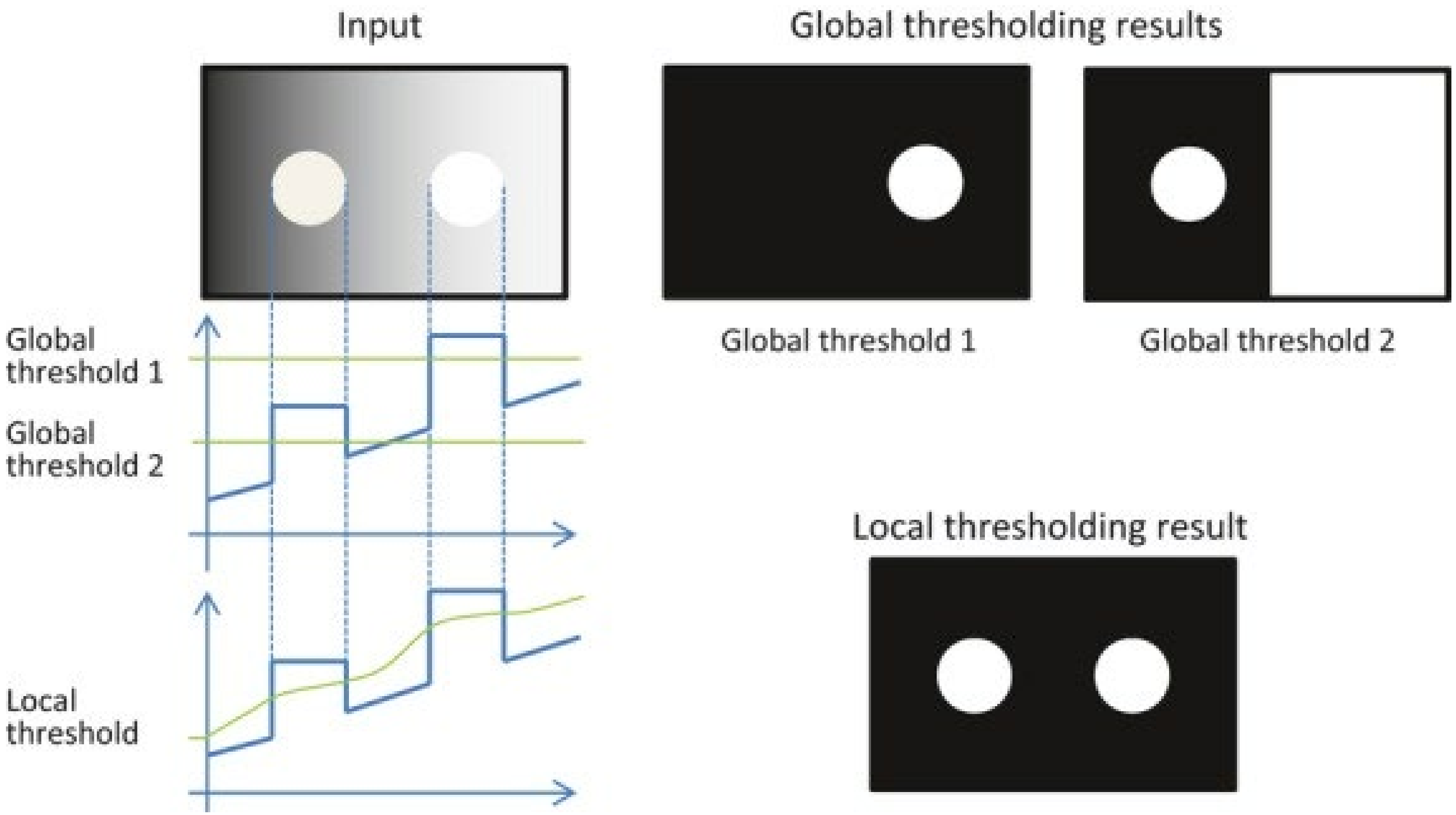


A global threshold is often (usually?) inadequate due to illumination gradients, differences in object levels and other variations across the image



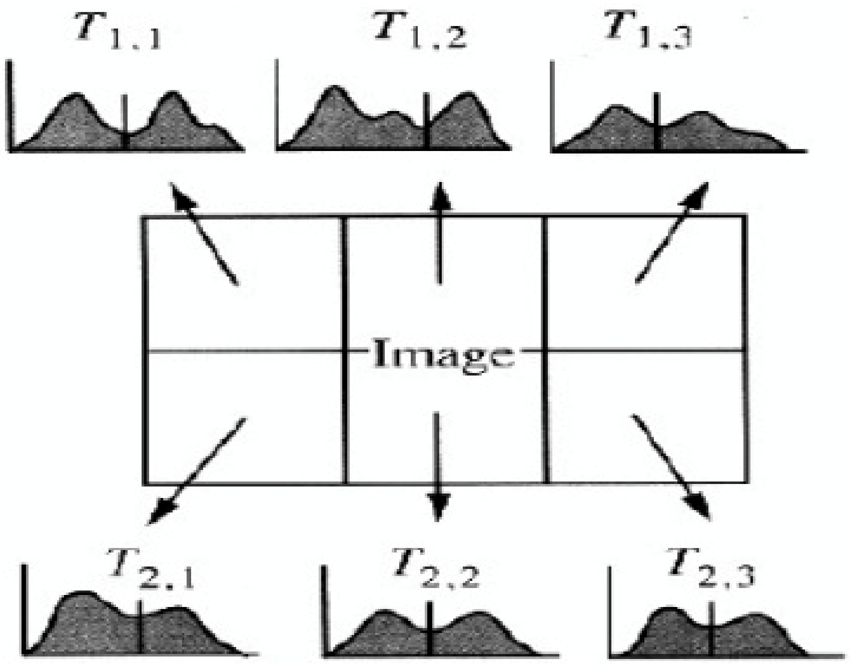


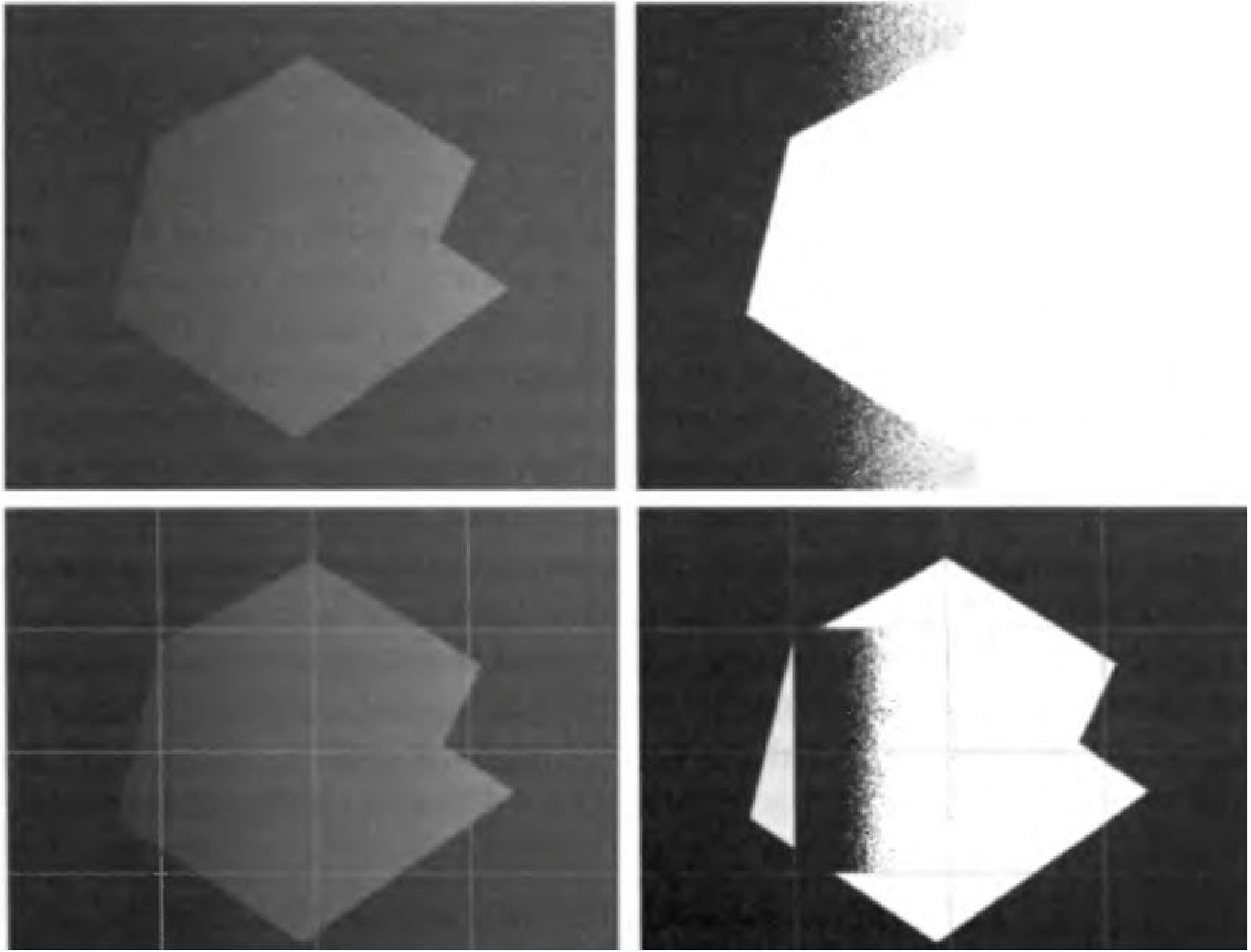
Illumination gradients are a particular source of difficulty; showing the need for a changing threshold across the image



Several local thresholding methods exist to calculate different thresholds for subregions of the image

- A simple approach is to divide the image into rectangular subregions and threshold each one individually
- This can work well for certain types of images
  - Poor for rapidly changing illumination
  - Poor for regions that don't contain both foreground and background



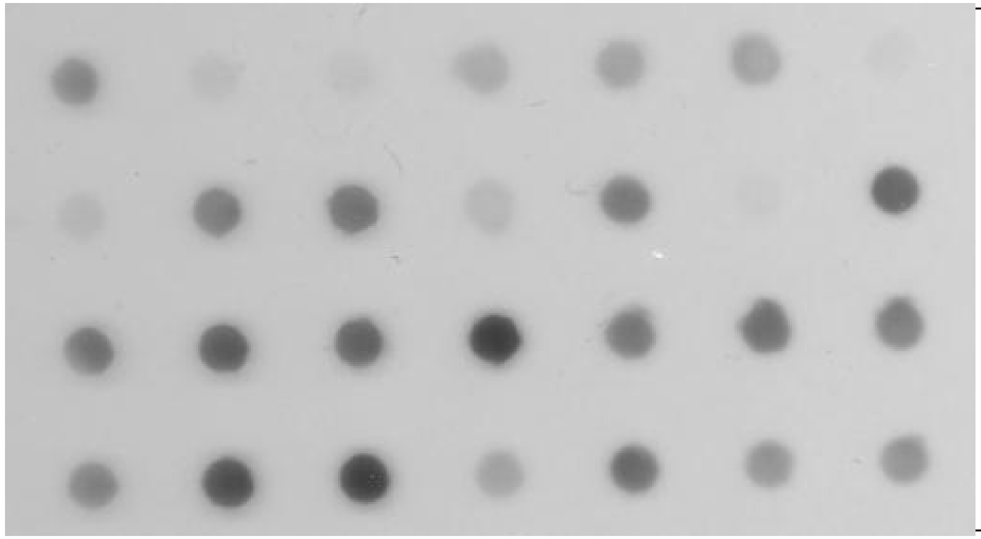


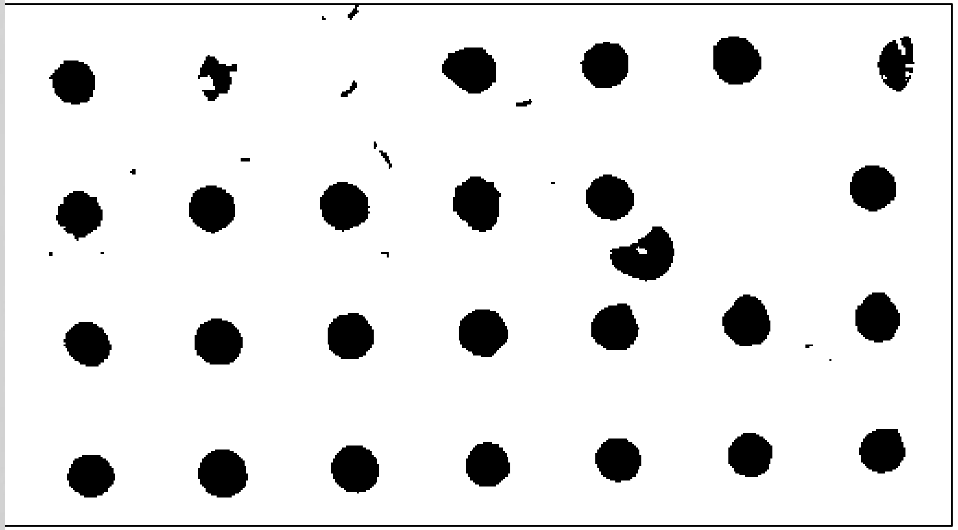
Bernsen's method calculates a different threshold value for each pixel location in the image, as the average of the local min and max intensity values

$$thr(c,r) = \frac{\max\limits_{region} I + \min\limits_{region} I}{2}, \text{if } \left(\max\limits_{region} I - \min\limits_{region} I\right) > \alpha$$

- If the difference between the local max and the local min is large enough, we average them to get the threshold
- If the difference (the local contrast) is low, then we can assume that the region only contains foreground or background
  - In this case, they are usually assigned to the background

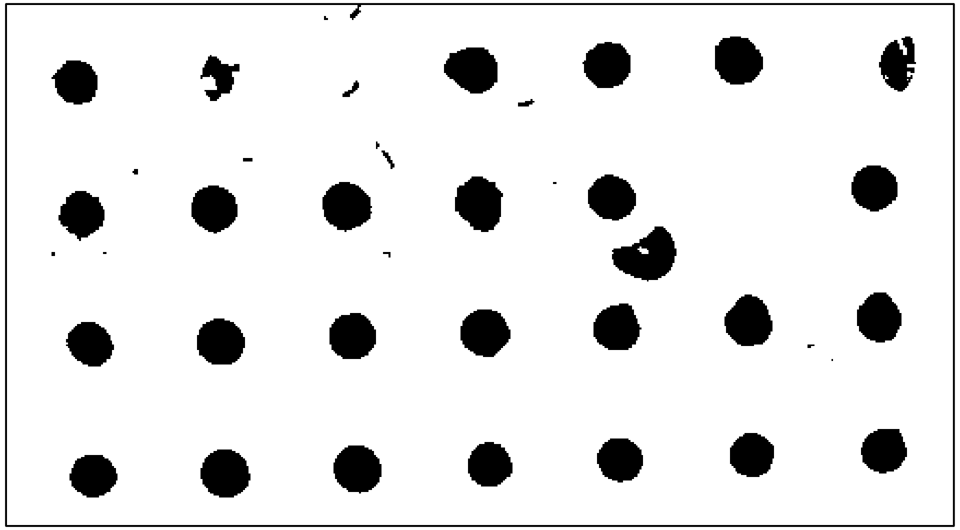
Bernsen’s method can be superior to global methods, but the selection of region size (radius) is critical



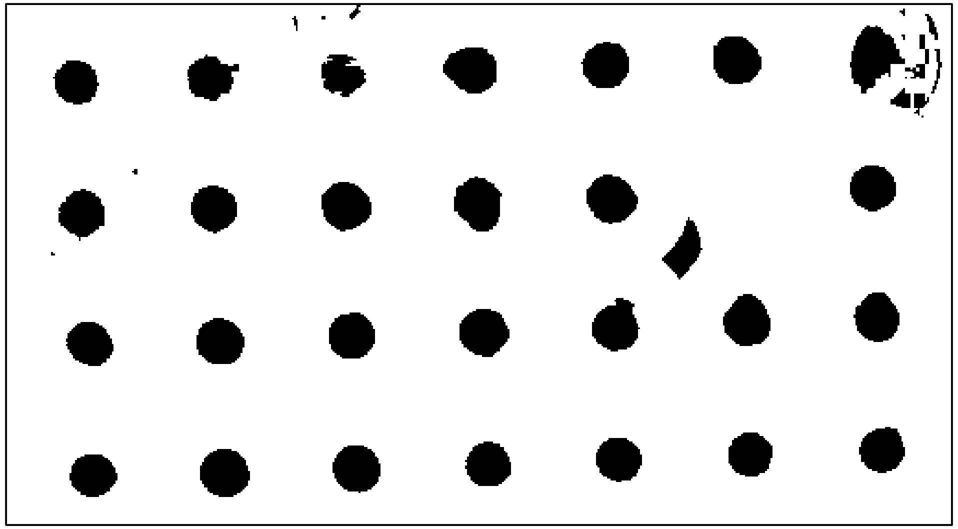


Bernsen's method can be superior to global methods, but the selection of region size (radius) is critical

radius = 15



radius = 30



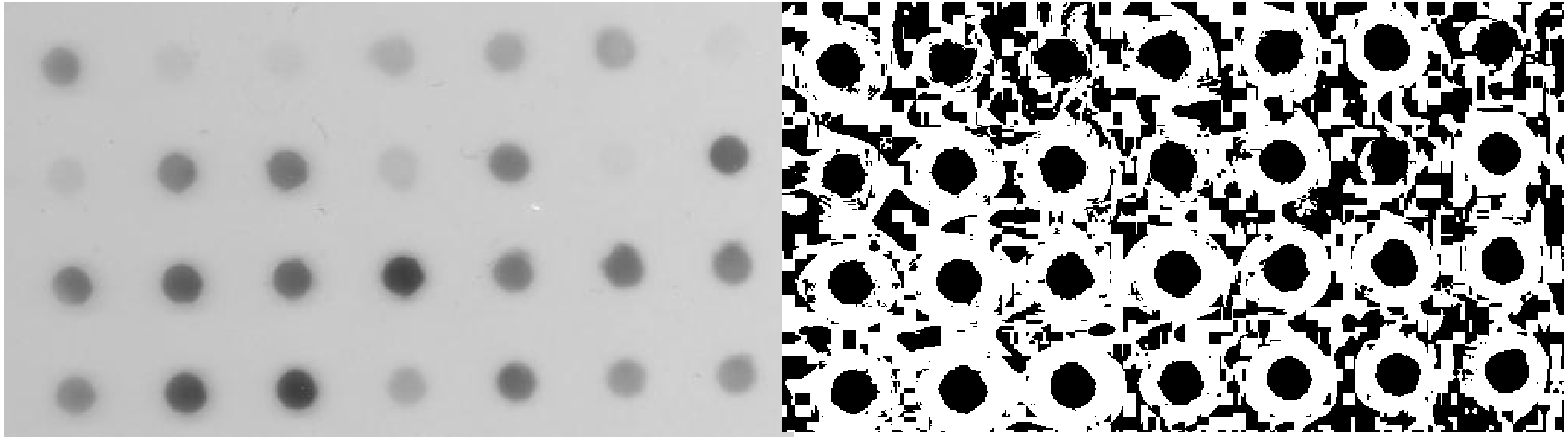
Niblack's method computes a local threshold as the local mean intensity plus a constant times the local standard deviation

$$T = \mu_R + k\sigma_R$$

$$thr(c,r) = \frac{1}{N_R}\left(\sum_R I(c,r) + k\sqrt{N_R\sum_R (I(c,r))^2 - (\sum_R I(c,r))^2}\right)$$

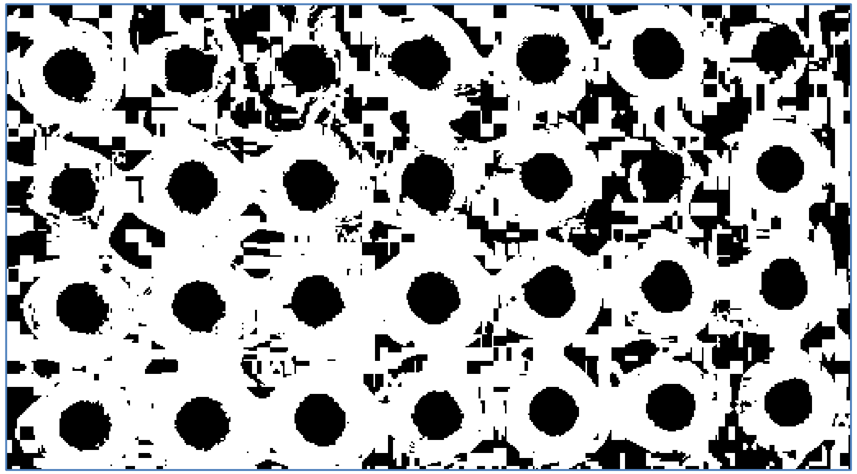
- The value k determines how much above (or below) the local mean we choose the threshold
- This value depends on the application
- One author uses k=-0.2 for processing ancient manuscript images

Niblack's method is also sensitive to the region size but the result of binarization shows different anomalies

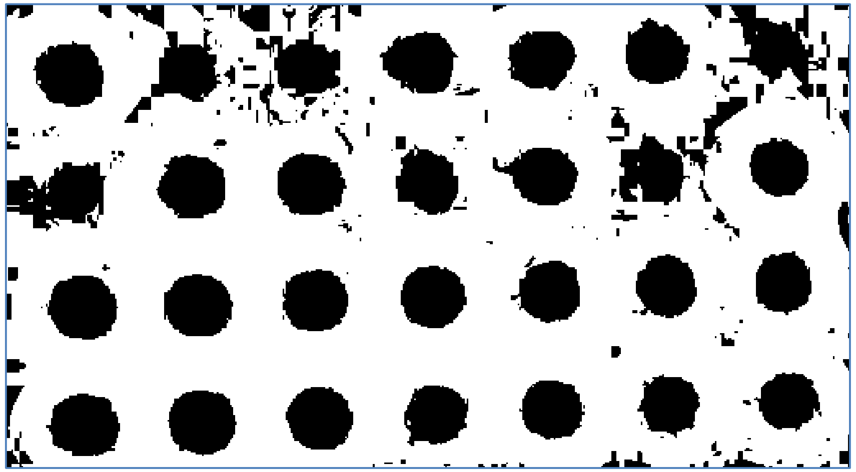


Niblack’s method can be superior to global methods, but the selection of region size (radius) is critical

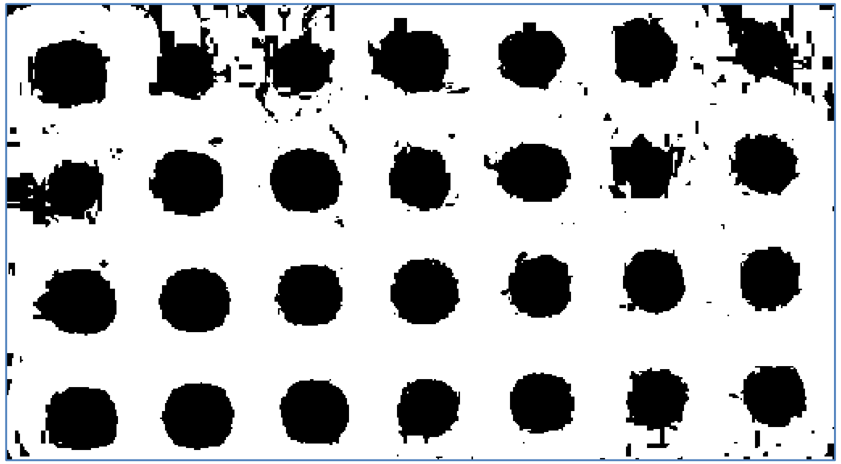
radius = 20



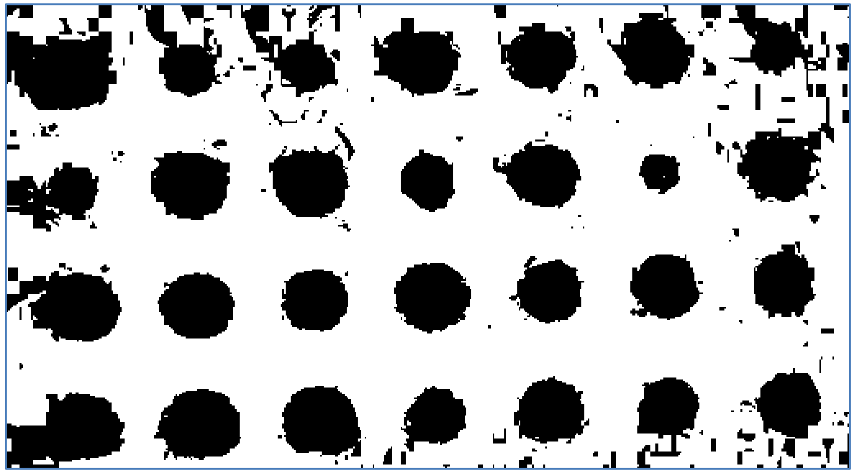
radius = 40



radius = 60



radius = 80



This paper does a good job of summarizing a number of the other local adaptive thresholding algorithms

- Comparison of Niblack inspired Binarization methods for ancient documents
- https://pdfs.semanticscholar.org/76bd/5997bdd6d0a2611c590f5cf30f95c88e18ad.pdf

Comparison of Niblack inspired Binarization methods for ancient documents

Khurram Khurshid[1], Imran Siddiqi[1], Claudie Faure[2], Nicole Vincent[1]
[1]Laboratoire CRIP5 – SIP, Université Paris Descartes
45, rue des Saints-Pères, 75006 Paris France
{khurshid ; siddiqi ; nicole.vincent} @ math-info.univ-paris5.fr
[2]UMR CNRS 5141 - GET ENST, 46 rue Barrault, 75634 Paris
cfaure@enst.fr

ABSTRACT

In this paper, we present a new sliding window based local thresholding technique 'NICK' and give a detailed comparison of some existing sliding-window based thresholding algorithms with our method. The proposed method aims at achieving better binarization results, specifically, for ancient document images. NICK has been inspired from the Niblack's binarization method and exhibits its robustness and effectiveness when evaluated on low quality ancient document images.

Keywords: Ancient documents, local binarization, Niblack

1. INTRODUCTION

The importance of digital libraries for information retrieval cannot be denied. The ancient historical books contain invaluable knowledge but it is very time consuming to search the required information in these paper books. Different

Step back and consider what we've done – because we couldn't choose a suitable global parameter (threshold), we have methods that require choice of a different suitable global parameter (radius)

How have we made the situation any better?

1. These methods are useful for images that deal with intensity variations across the image, where the image's *detail size* may be more consistent
2. It turns out that most images are less sensitive to values of radius than to values of threshold
3. Is there a place for <u>adaptive radius methods?</u> (I have never seen any research on this, but it's an interesting idea)

Today’s Objectives

- Concept of Image Segmentation
- Global Thresholding
  - Laplacian thresholding
  - Kittler-Illingworth and Otsu methods
  - Histogram-based
- Local Thresholding
  - Bernsen’s method
  - Niblack’s method