Regions and Edges:

Image content at an elementary level

Main topics:

- Thresholding
- Histogram analysis

Regions and Edges

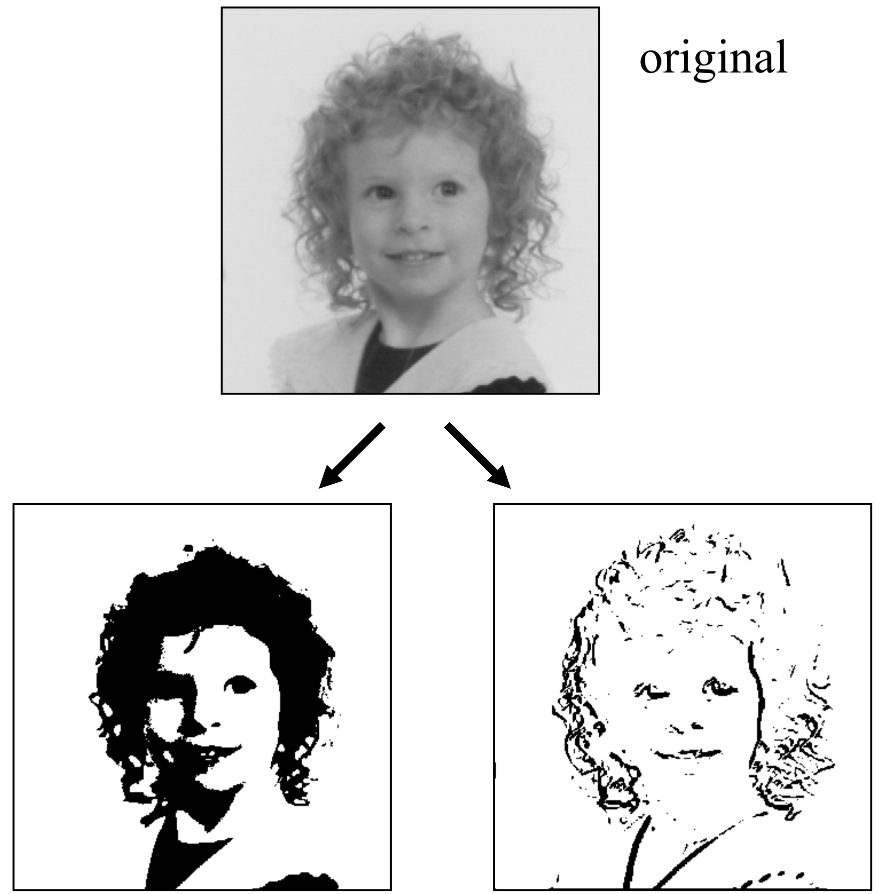
What do you see here?

What do you see here?

How can we begin to extract useful information from an image?

- Starting from the original grayscale image, it is easy to generate these 2 new images
- Although the amount of data is reduced considerably, the new images seem to retain some essential, elementary information that can be used for analysis
- Much image analysis depends on extracting ***features*** from images, and in many cases, the features are simply properties of *regions* and *edges* in an image

original



regions

intensity edges

Some definitions

- A **binary image** is an image in which each pixel assumes 1 of 2 possible values
- Typically these are called **foreground** and **background** values
- The process of assigning 1 of 2 values to each pixel is sometimes called **binarization**
- Commonly, but not always:
    - Foreground = 1 or 255
    - Background = 0

Image edges

- An **edge** in an image refers to a sudden change in pixel values
- The image at the lower right is sometimes called an "edge image"; the dark pixels indicate edge locations
- Many different edge-detection methods have been developed
- Most can be characterized as high-pass filters in the spatial domain

Image regions

- A **region** is a connected portion of an image in which the pixels are considered uniform
- The image at the lower right was generated by a procedure known as ***thresholding***
- We think of each connected set of foreground pixels as a separate region

Thresholding

- Compare every pixel value with a given constant *T*, which is known as the threshold value
- For my earlier example,
  - If the pixel value is <u>below</u> the threshold, then assign the pixel to the <u>foreground</u>
  - If the pixel value is <u>above</u> the threshold, then assign the pixel to the <u>background</u>

$$I_{NEW}(r,c) = \begin{cases} 1 & if\ I(r,c) \leq T \\ 0 & if\ I(r,c) > T \end{cases}$$

- In some situations, you'll need to reverse the assignment:

$$I_{NEW}(r,c) = \begin{cases} 1 & if\ I(r,c) > T \\ 0 & if\ I(r,c) \leq T \end{cases}$$

Examples of thresholding

$T = 124$

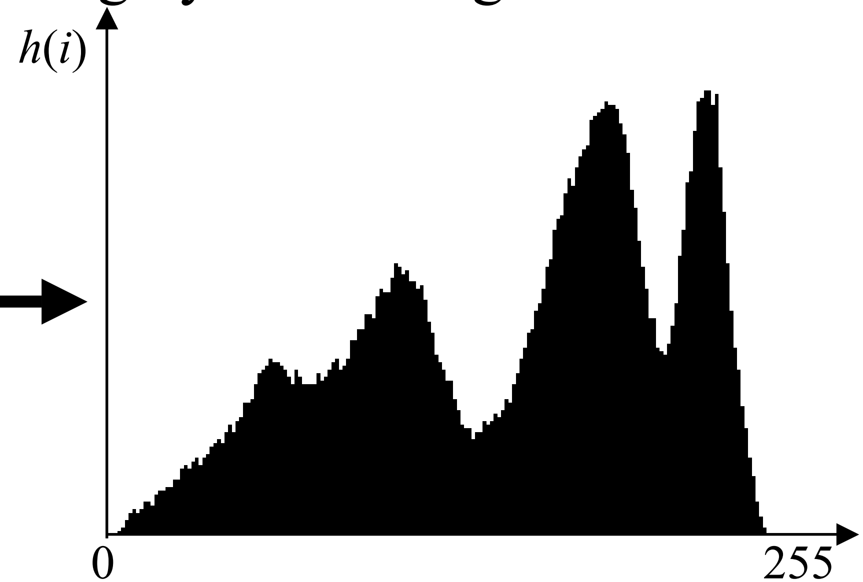
$T = 137$

$T = 159$

$T = 213$

How should we select the threshold value?

- One approach:
  Compute a **histogram** of the gray-scale image



- A histogram is a representation of the statistical distribution of observed values
- Analyze the histogram to identify significant concentrations of intensity values, and select a threshold

Histograms

- For an image, histogram $h(i)$ indicates the number of pixels having value i
- We could define the histogram for an image as follows:

$$h(i) = \frac{1}{N} \sum_{r} \sum_{c} p(r, c, i)$$

where $p(r,c,i) = \begin{cases} 1 & \text{if} \quad I(r,c) = i \\ 0 & \text{otherwise} \end{cases}$

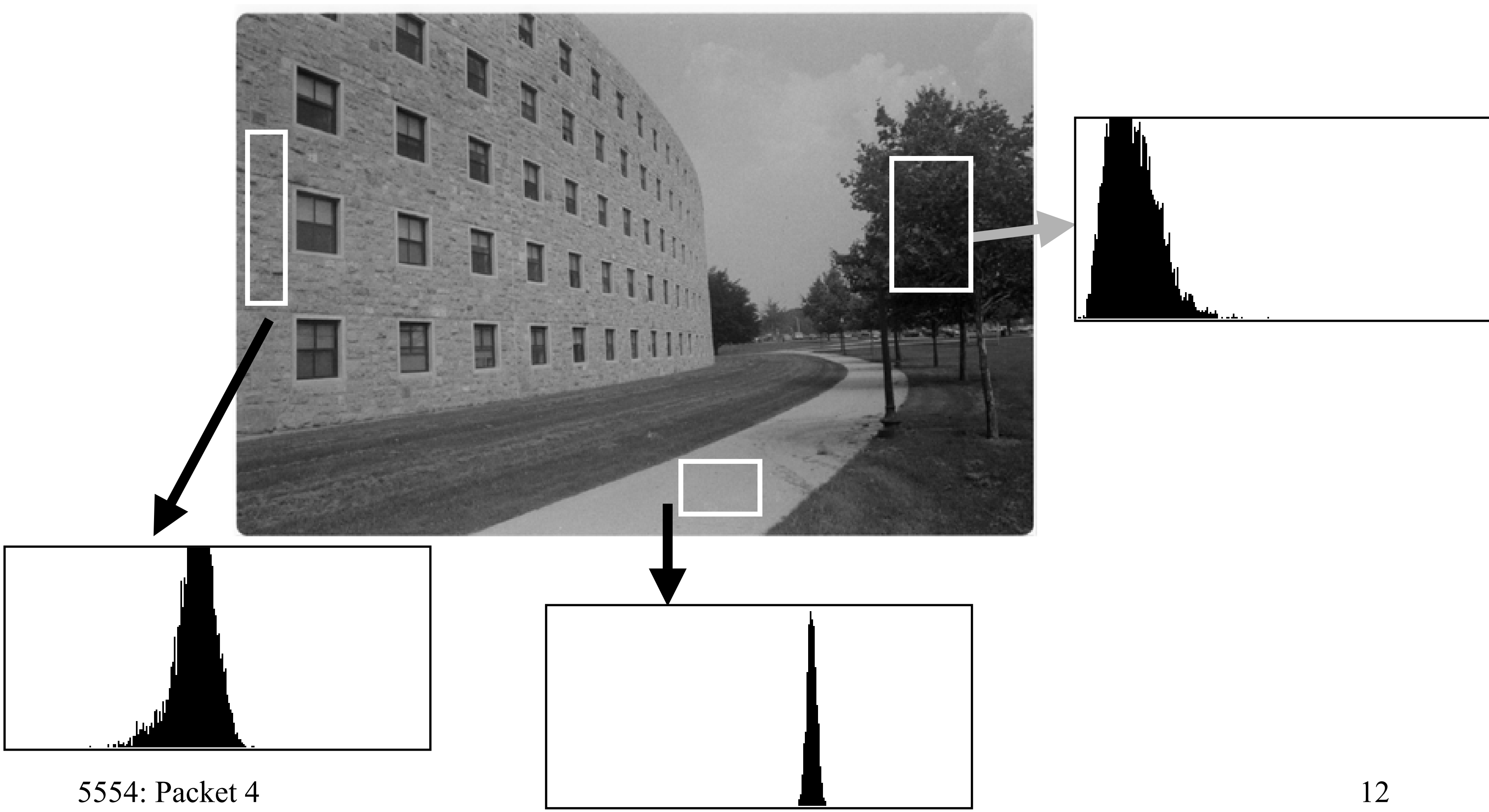
and N is the number of pixels in I

- For convenience, the division by N is often omitted

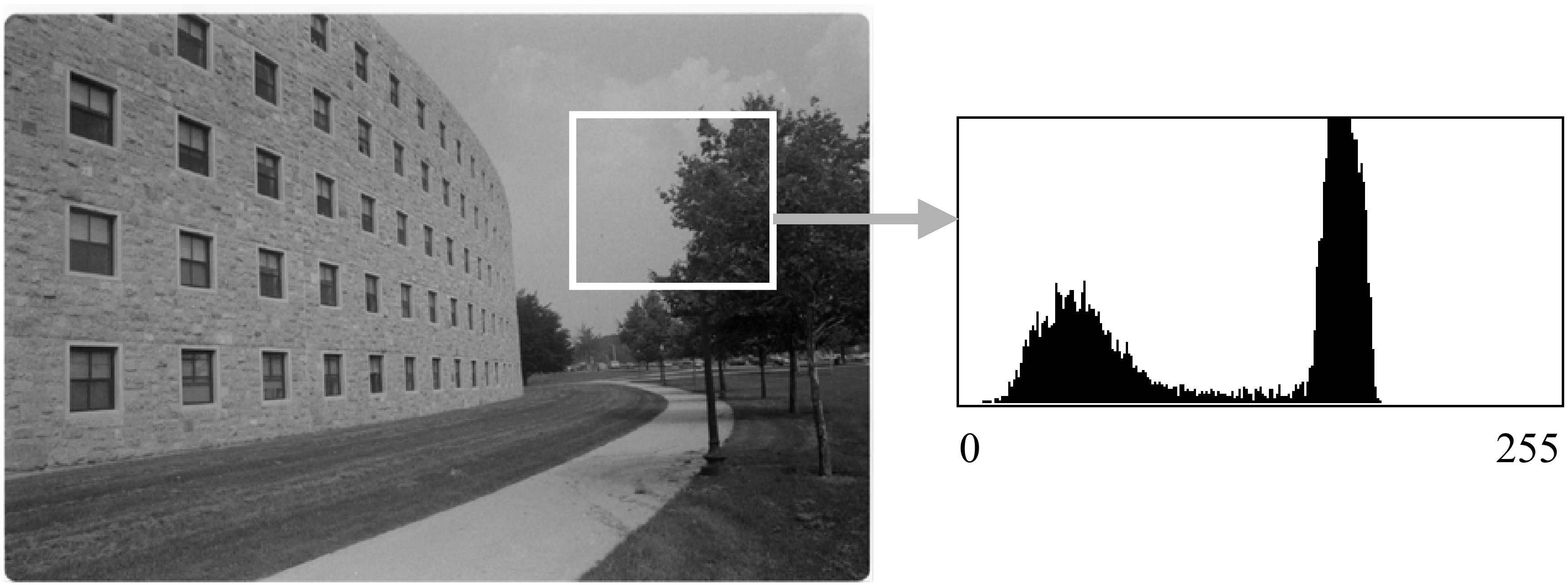
Histogram Properties

- A histogram is often used as an estimate of the probability distribution of image intensities
- In some cases, it is useful to think of a histogram as the sum of several Gaussian distributions
- A histogram does not contain information about the position of image contents
- We can think of a histogram as a vector
- Histograms are not limited to image intensities; we could compute the histogram of *any* set of numbers
- In general, coarser measurement intervals could be used

Histograms can be very different for different parts of an image

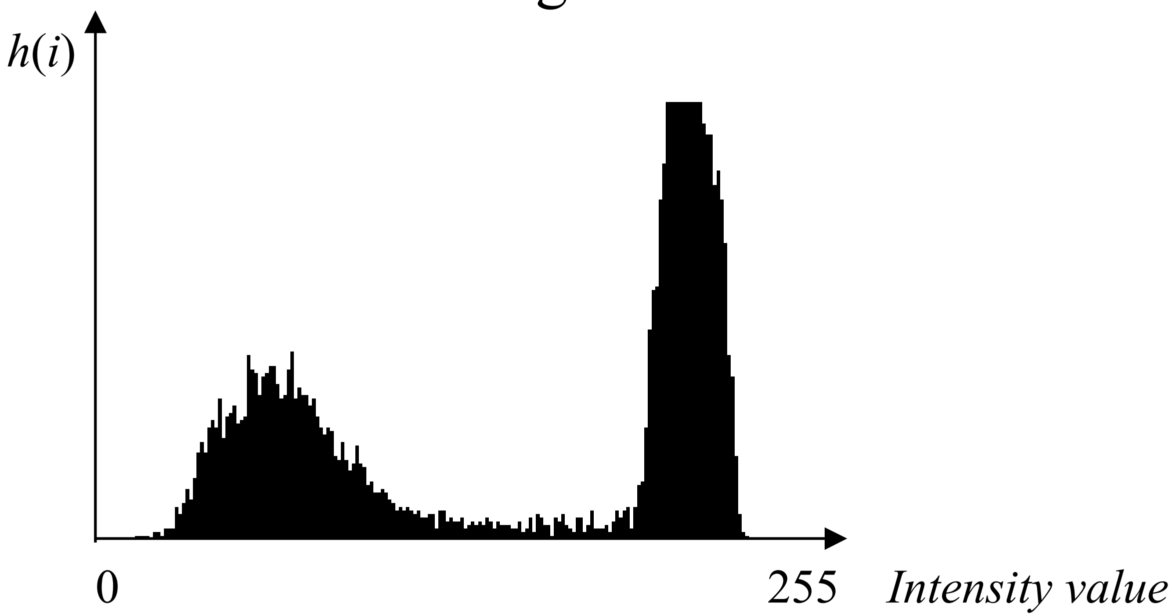


A common assumption is that histograms are “bimodal”



Histogram analysis for automatic threshold selection

- Usually, assume that the histogram is bimodal



- The high-level idea:
  - Identify the 2 highest most significant concentrations of intensity values
  - Select a threshold that separates the 2 concentrations

Otsu’s method (1979)

- Assumptions:
  - There are 2 natural groups of intensity values
  - The variance should be small within a group
- Goal:
  - Minimize the “within-group” variance
- High-level approach:
  - Consider all possible threshold values t
  - (Notice that each choice of t separates the histogram into 2 groups)
  - For each t, compute the variances for the 2 groups
  - Select the value of t that minimizes the expected value of group variance
- Otsu found an efficient way to do the computations

Computations

$$q_1(t) = \sum_{i=0}^{t} h(i)$$ = estimated probability that pixel value falls in group 1

$$q_2(t) = \sum_{i=t+1}^{255} h(i)$$ = estimated probability that pixel value falls in group 2

Select t to minimize

$$\sigma_W^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

"within-group" variance

variance of group 1

variance of group 2

Computations

$$\mu_1(t) = \sum_{i=0}^{t} ih(i) / q_1(t)$$

$$\mu_2(t) = \sum_{i=t+1}^{255} ih(i) / q_2(t)$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 h(i) / q_1(t)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{255} [i - \mu_2(t)]^2 h(i) / q_2(t)$$

Other Methods

- Iterative Selection (Ridler 1978):
    - Use the mean pixel value as threshold
    - Calculate mean values for two regions
    - Use the average of the two values as the new threshold
    - Continue till no change
- Minimize Information Measure(Kittler & Illingworth)

$$H = \frac{1 + \log 2\pi}{2} - q_1 \log q_1 - q_2 \log q_2 + \frac{1}{2}\left(q_1 \log \sigma_1^2 + q_2 \log \sigma_2^2\right)$$

Other Methods-2

- Fuzzy Method
  - *Minimize Fuzziness*
- Maximize Entropy
- Regional Thresholds
  - Subdivide the image into sub-images and threshold

Other Methods-3

- Relaxation Method
  - Use Mean Value as an initial threshold for all pixels
  - Create a probability of foreground for each pixel based on the distance from the mean
  - Use compatibility measure of 1 or –1 if the pixel is same as the neighbor.
  - Combine with all neighbors
  - Iteratively update probabilities using compatibility till we achieve convergence
- Moving Average Method
    - Convert to single array and threshold comparing to moving average