EEE-6512: Image Processing and Computer Vision

November 6, 2017

Lecture #10: Segmentation

Damon L. Woodard

Dept. of Electrical and Computer Engineering

dwoodard@ece.ufl.edu

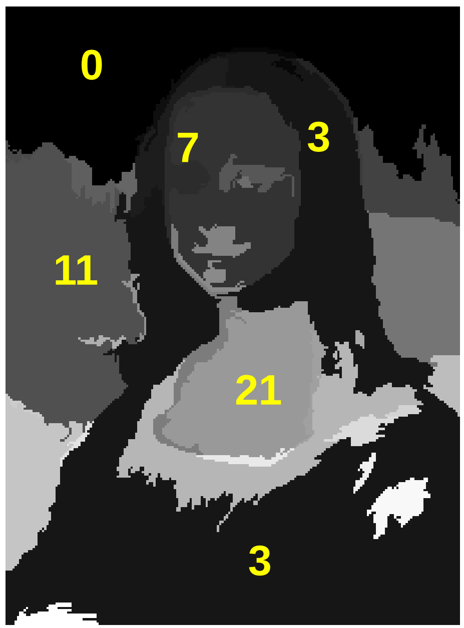
Chapter Outline

- Overview
- Thresholding
- Gestalt Psychology
- Image Segmentation
- Graph-Based Methods

What is segmentation?

- Segmentation divides an image into groups of pixels
- Pixels are grouped because they share some local property (gray level, color, texture, motion, etc.)

boundaries



labels

pseudocolors

mean colors

(different ways of displaying the output)

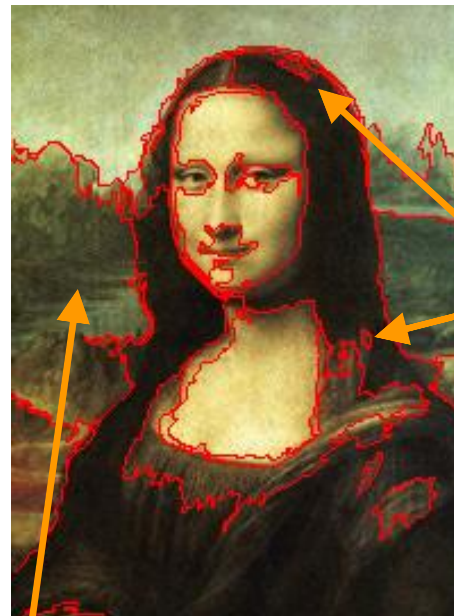
algorithm used: Pedro F. Felzenszwalb and Daniel P. Huttenlocher, Efficient Graph-Based Image Segmentation, IJCV, 59(2), 2004

Segmentation examples

*

* Pictures from Mean Shift: A Robust Approach toward Feature Space Analysis, by D. Comaniciu and P. Meer http://www.caip.rutgers.edu/~comanici/MSPAM

Two errors



oversegmentation (hair should be one group)

undersegmentation (water should be separated from trees)

Simplest Segmentation Problem

Foreground/Background Segmentation

Goal: Separate foreground objects from background.

Common approaches involve finding a **threshold value** which produces a image in which foreground pixels are separated from background pixels.

Thresholding

Global Thresholding

- Let τ be a single **global threshold**. Once τ has been determined, it is applied to every pixel in a straightforward manner:

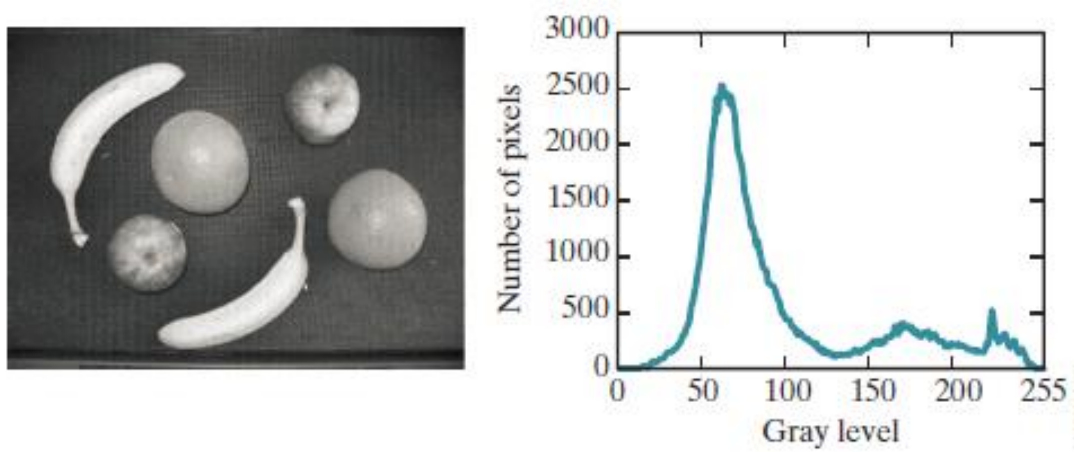
$$I'(x,y) = \begin{cases} \text{ON} & \text{if } I(x,y) > \tau \\ \text{OFF} & \text{otherwise} \end{cases}$$

- Two simple, widely used global thresholding technique are known as the Ridler-Calvard algorithm and Otsu's method.
- It is recommended to process image before determing thresholds
  - Smoothing to peaks in histogram more distinctive
  - Compute edges and apply algorithms only to pixels near edge (reduce asymmetry).

Ridler-Calvard Algorithm

Let τ be a threshold, and let $\mu_{\triangleleft}$ be the mean gray level of all the pixels whose gray level is less than or equal to τ, while $\mu_{\triangleright}$ is the mean gray level of all the pixels whose gray level is greater than τ. If we assume that the background is darker than the foreground, then $\mu_{\triangleleft}$ is the mean of the background pixels, whereas $\mu_{\triangleright}$ is the mean of the foreground pixels.

Figure 10.1 LEFT: A grayscale image of several types of objects (fruit) on a dark background (conveyor belt). RIGHT: The graylevel histogram of the image.



Ridler-Calvard Algorithm (cont.)

Method:

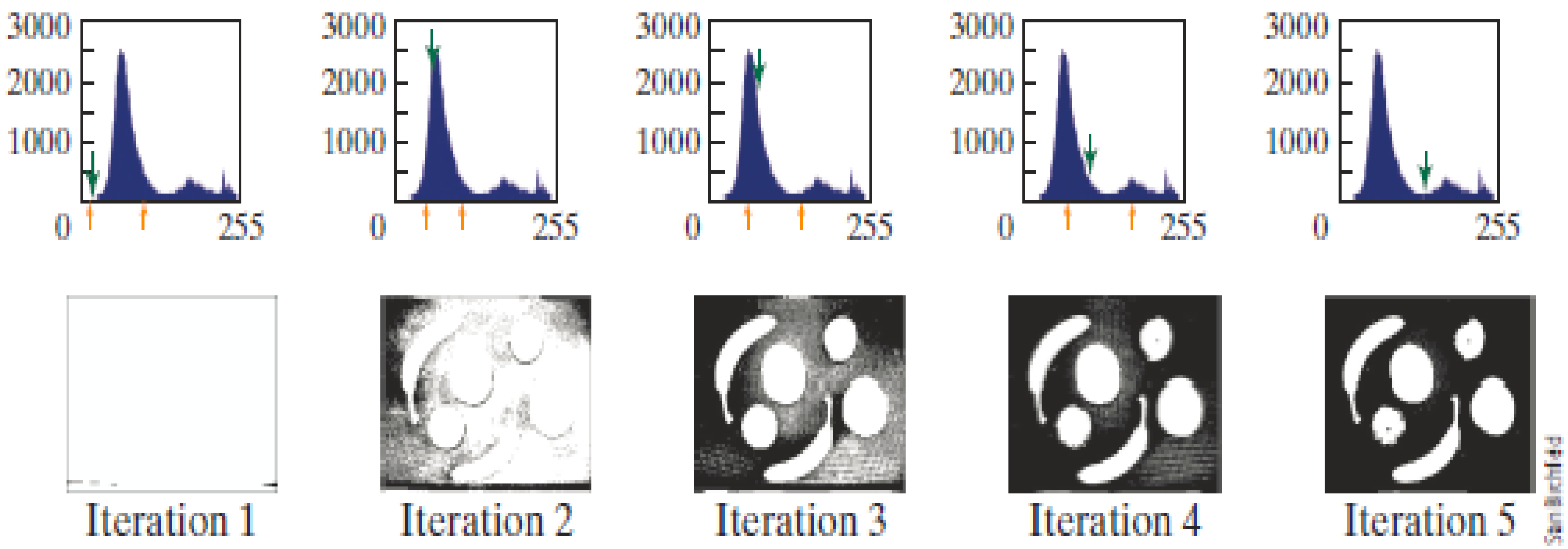
The Ridler-Calvard algorithm iteratively computes two means (using histogram moments) based on the current estimate of the threshold, then sets the threshold to the average of the two means.

Based on the assumption that the foreground and background gray levels are distributed as Gaussians with equivalent standard deviations.

$$\tau = \frac{(\mu_{\triangleleft} + \mu_{\triangleright})}{2}$$

Ridler-Calvard Algorithm (cont'd)

Figure 10.2 Step-by-step example of the Ridler-Calvard algorithm applied to the image of Figure 10.1. Note that even with an initial threshold far from the true solution, the algorithm converges in only five iterations. The top row shows the histogram. The green arrow pointing down indicates the threshold at each iteration, while the gold arrows pointing up indicate the two means. The bottom row shows the result of thresholding the image using the threshold for that iteration.



Otsu's Method

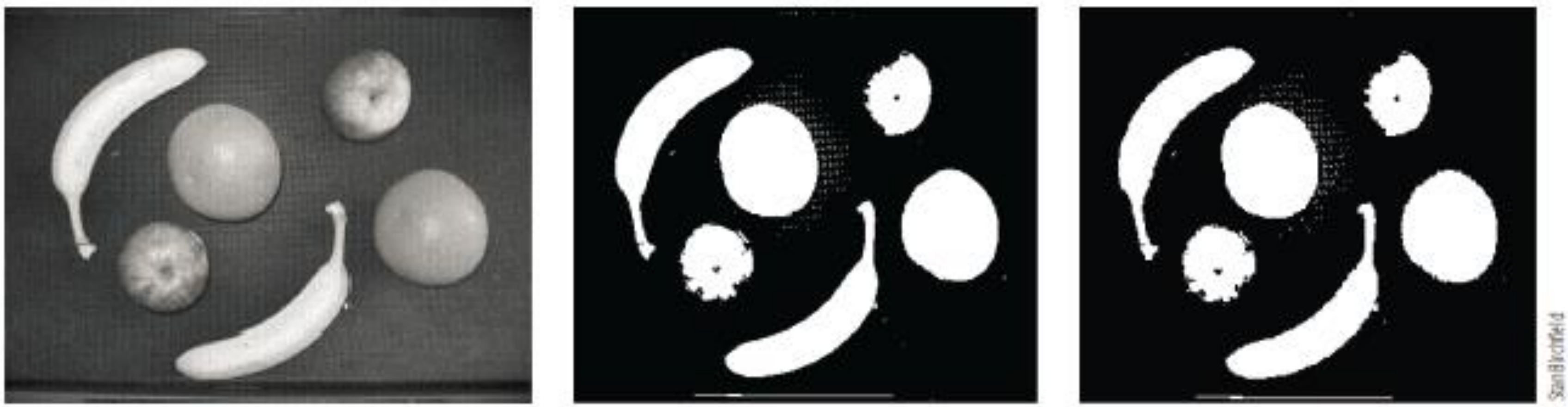
- The goal of Otsu's method is to find the threshold τ that minimizes the *within-class variance*, which is defined as the weighted sum of the variances of the two groups of pixels:

$$\sigma_w^2(\tau) \equiv p_{\blacktriangleleft}(\tau)\sigma_{\blacktriangleleft}^2(\tau) + p_{\triangleright}(\tau)\sigma_{\triangleright}^2(\tau)$$

- Relaxes the assumption that the two regions have the same variances.
- The key difference in the Otsu algorithm is that it performs an exhaustive search over all possible threshold values.

Otsu’s Method(cont’d)

Figure 10.3 From left to right: Input image, output of the Ridler-Calvard algorithm, and output of Otsu’s method. On this particular image, the outputs are almost indistinguishable.



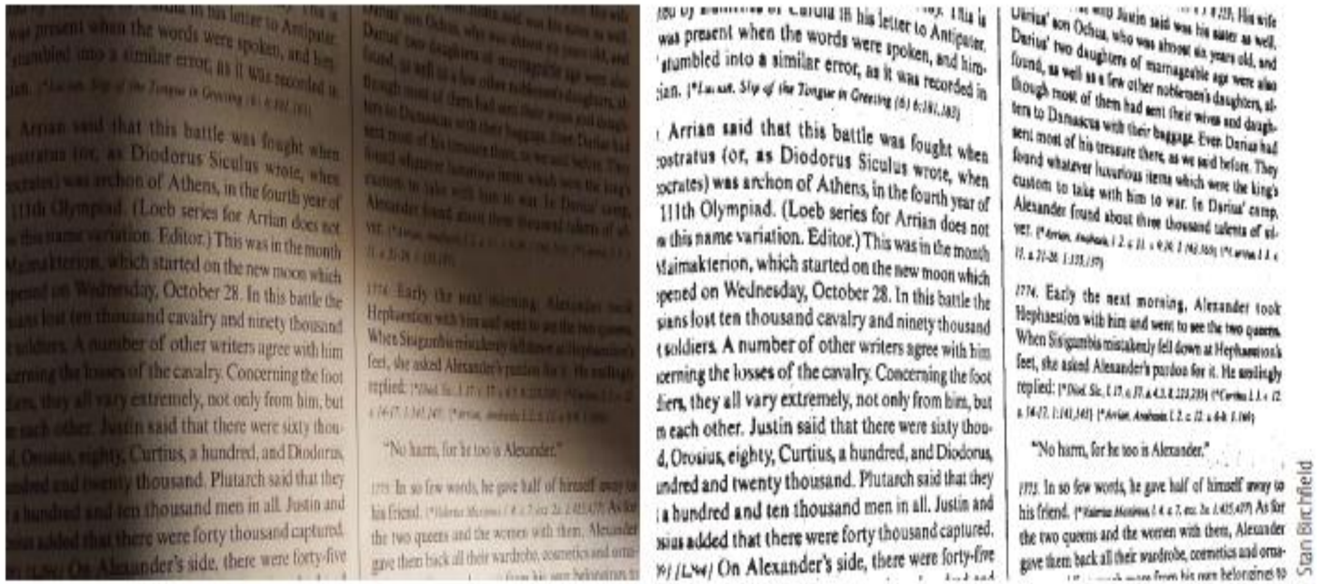
Adaptive Thresholding

- Global thresholding techniques such as the Ridler-Calvard algorithm and Otsu's method do not perform well when the image noise characteristics vary across the image.
- To overcome such difficulties, **adaptive thresholding** techniques are needed, in which the threshold used at any given pixel in the image is based upon local statistical properties in the neighborhood of the pixel:

$$I'(x,y) = \begin{cases} \text{ON} & \text{if } I(x,y) > \tau(x,y) \\ \text{OFF} & \text{otherwise} \end{cases}$$

Example of Adaptive Thresholding

Figure 10.4 Example of adaptive thresholding.



Adaptive Thresholding (cont'd)

- **Chow-Kaneko**: technique in which the image is divided into overlapping blocks, and the histogram is examined for each block to determine a threshold value for the block.
  - Interpolation between these threshold values then yields a threshold function defined over the entire image.

- **Niblack's method:** the threshold is given by:

$$\tau(x, y) = \mu(x, y) - k \cdot \sigma(x, y)$$

Hysteresis Thresholding

- The concept of **hysteresis thresholding** uses a low threshold τ_{low} and a high threshold τ_{high}.
- Any pixel is labeled ON if it is either above the high threshold or above the low threshold and connected to another pixel that is above the high threshold.

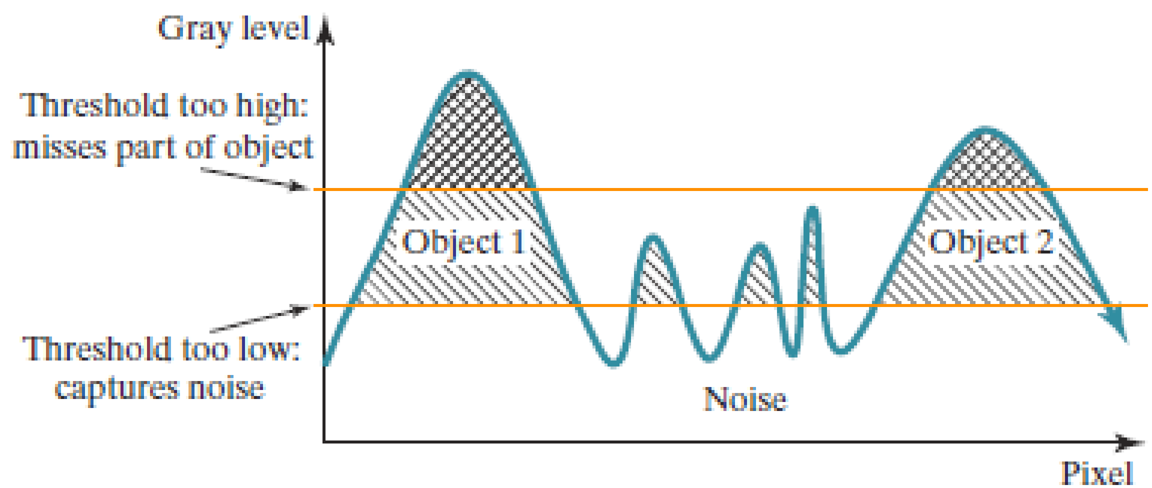
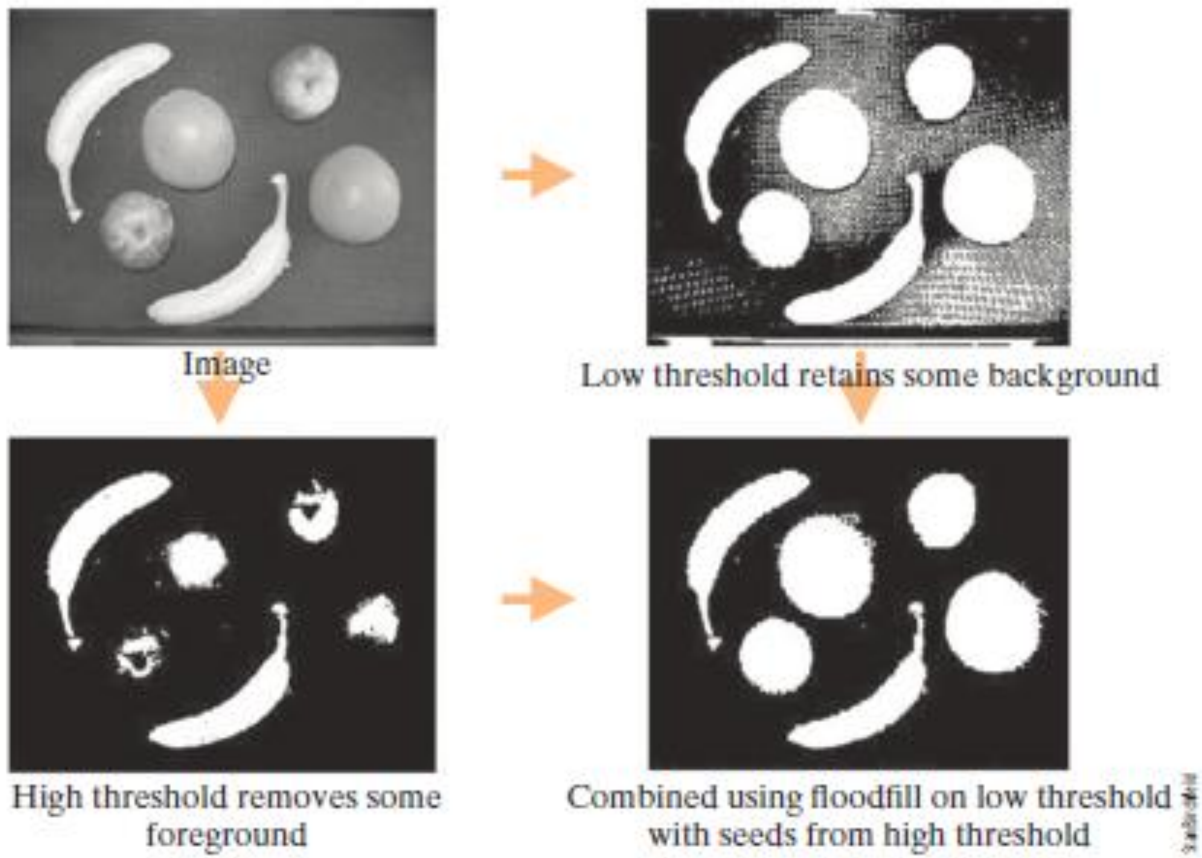


Figure 10.5 An illustration of hysteresis thresholding, also known as double thresholding.

Hysteresis Thresholding (cont'd)

Figure 10.6 An example of hysteresis thresholding.



Multilevel Thresholding

- **Multilevel thresholding:** when the graylevel histogram has multiple peaks, with distinct valleys between the peaks, it is desired to assign a different output value to each peak.
- It is straightforward to extend Otsu's method to the case of multiple levels, leading to the **multilevel Otsu method.**

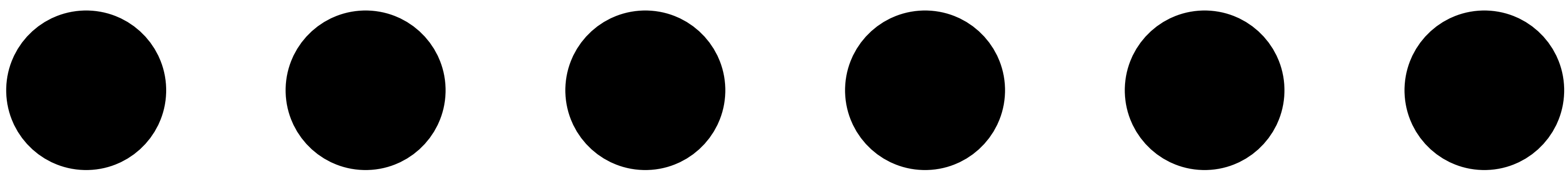
$$I'(x,y) = \begin{cases} v_1 \ \ if\ I(x,y) \le \ \tau_1 \\ v_3\ if\ I(x,y) > \ \tau_2 \\ v_2 \ \ otherwise \end{cases}$$

- In case of three levels:

$$\sigma_b^2 = p_1\left(\mu_1 - \mu\right)^2 + p_2\left(\mu_2 - \mu\right)^2 + p_3\left(\mu_3 - \mu\right)^2$$

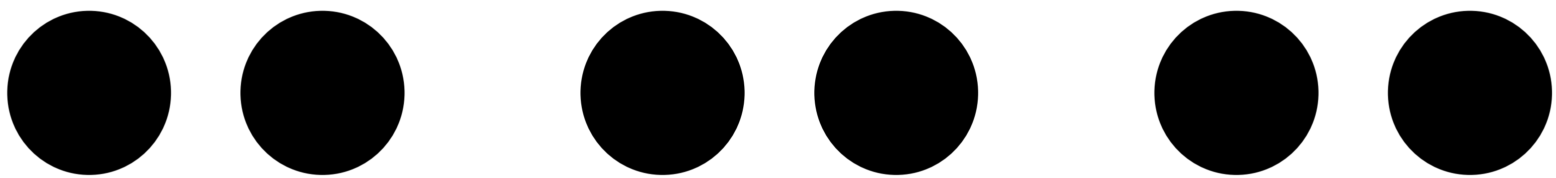
Gestalt Psychology

An experiment:
What do you see?



Just six dots

Now what do you see?

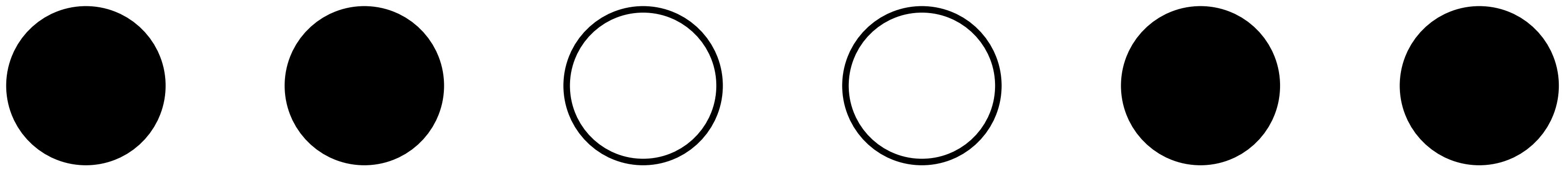


Three groups of dot pairs

Why?

Dots that are close together ("proximity")
are grouped together by the human visual system

And now?

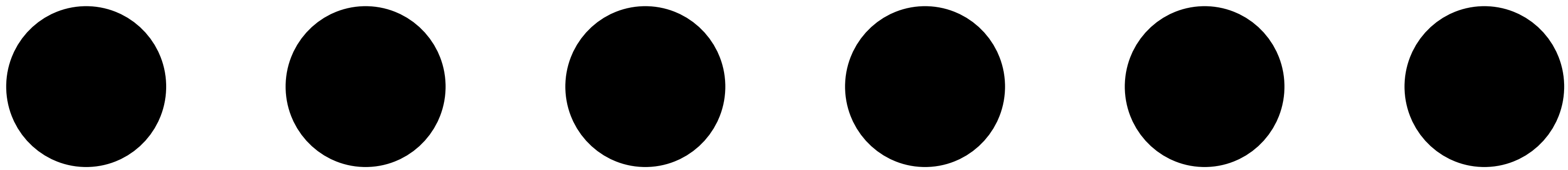


Again, three groups of dot pairs

Why?

Dots are similar in appearance (“similarity”)

How about now?

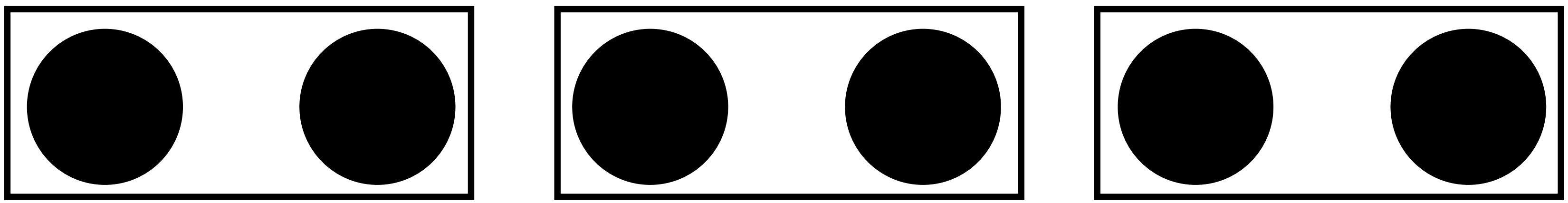


Again, three groups of dot pairs

Why?

Dots move similarly ("common fate")

Last one

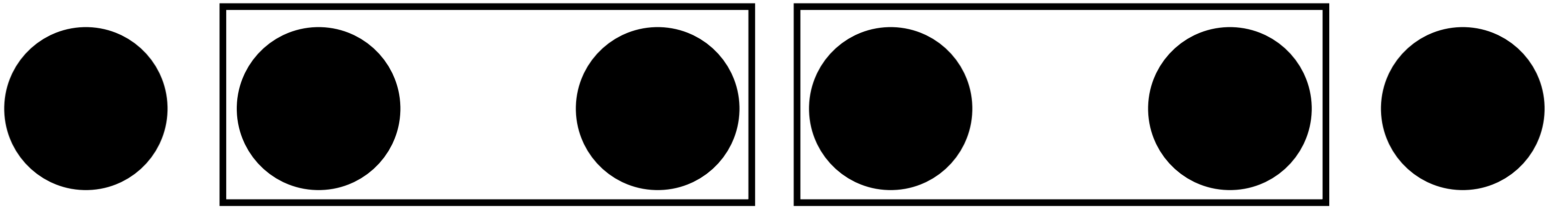


Again, three groups of dots

Why?

Dots are enclosed together (“common region”)

But wait!



Note that the “common region” can overwhelm the “proximity” tendency

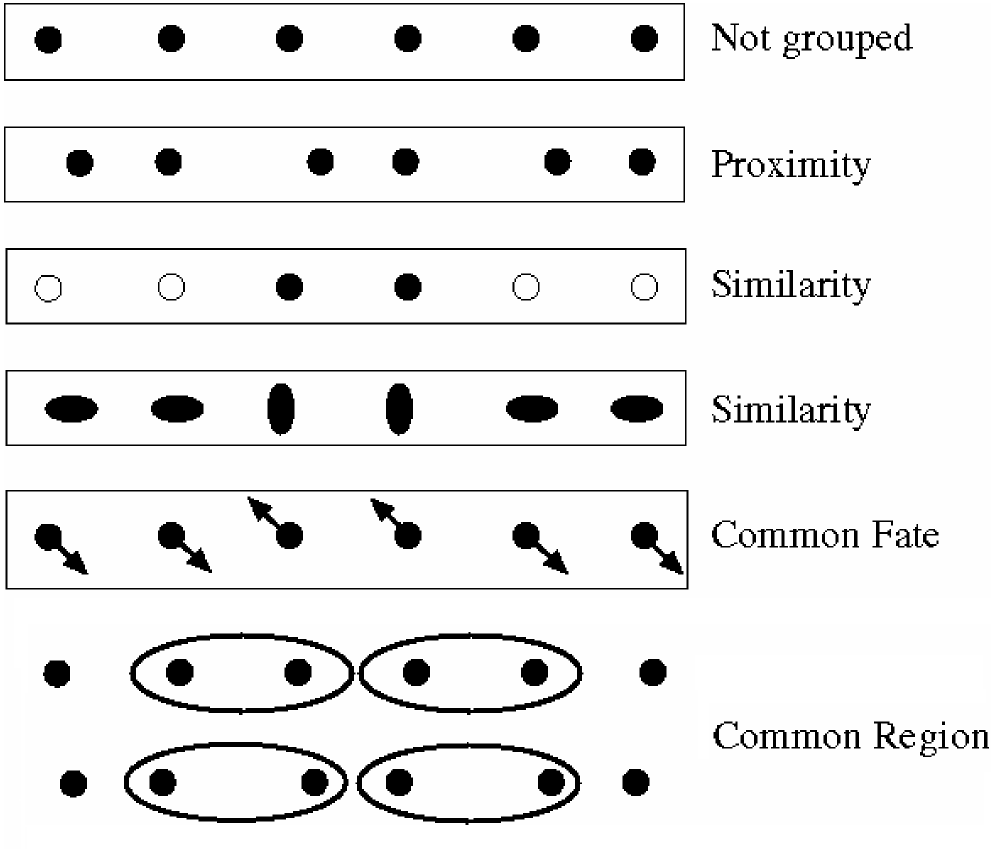
Gestalt psychology

Gestalt school of psychologists emphasized grouping as the key to understanding visual perception.

Recall: Context affects how things are perceived

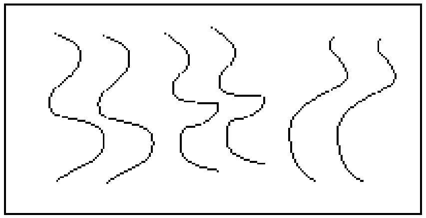
***gestalt* – whole or group**

***gestalt qualitat* – set of internal relationships that makes it a whole**

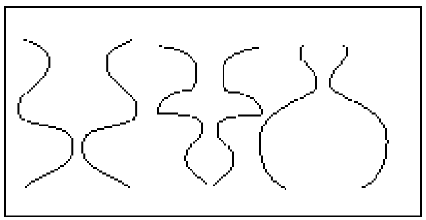


Max Wertheimer, Laws of Organization in Perceptual Forms, 1923
http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm

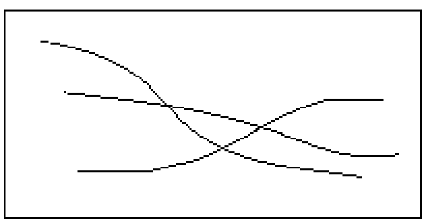
Gestalt psychology (cont.)



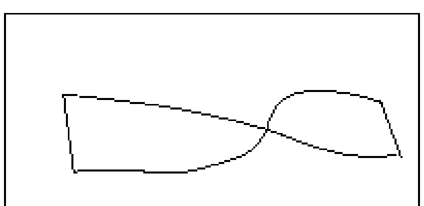
Parallelism



Symmetry

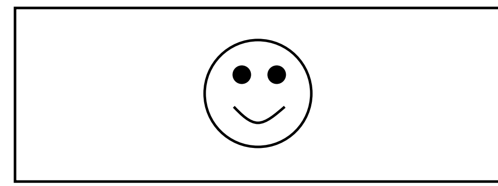


Continuity

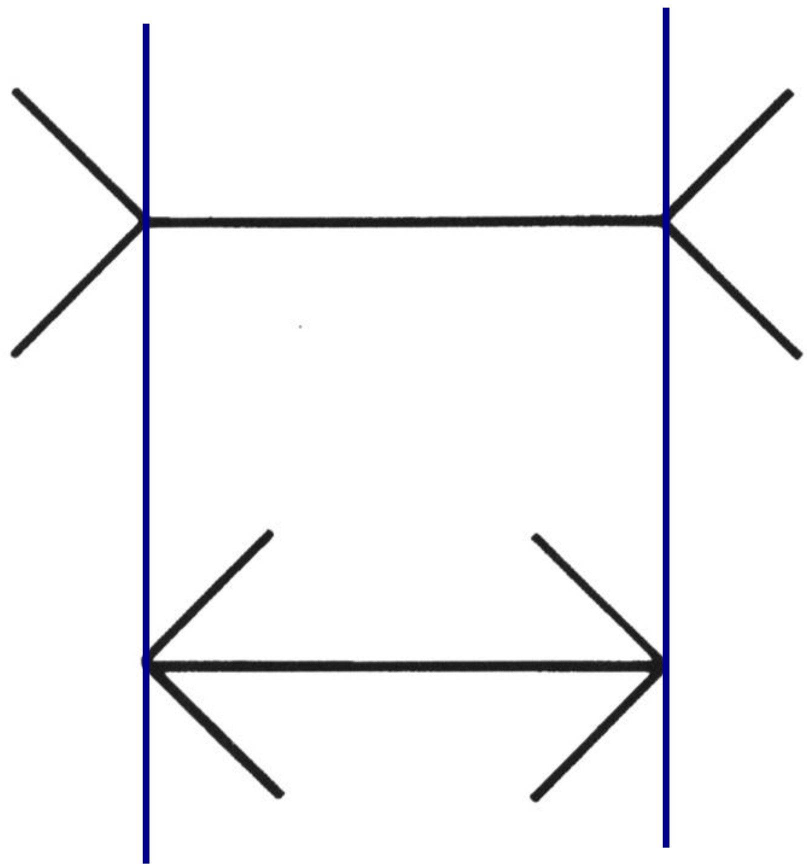


Closure

Familiar configuration

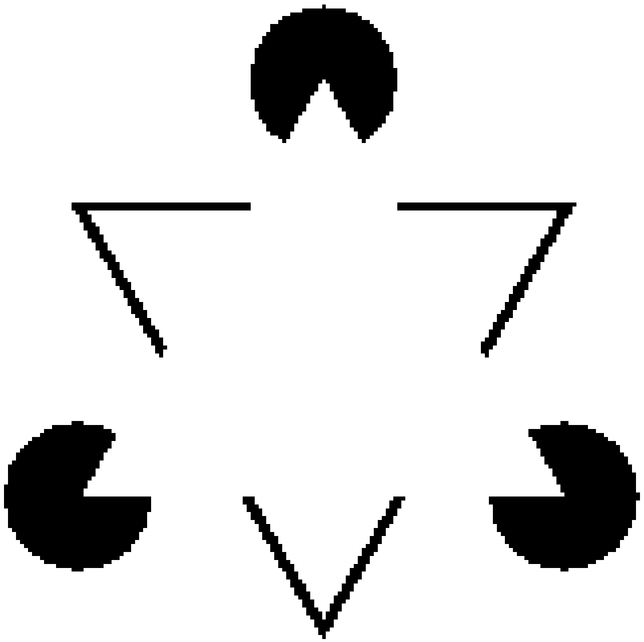


Muller-Lyer illusion



Lines are perceived as components of a whole rather than as individual lines.

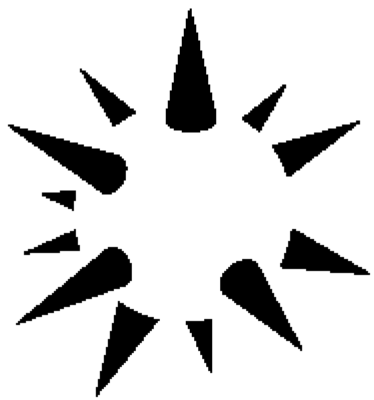
Can you see anything invisible?



These are illusory contours, formed by grouping the circles

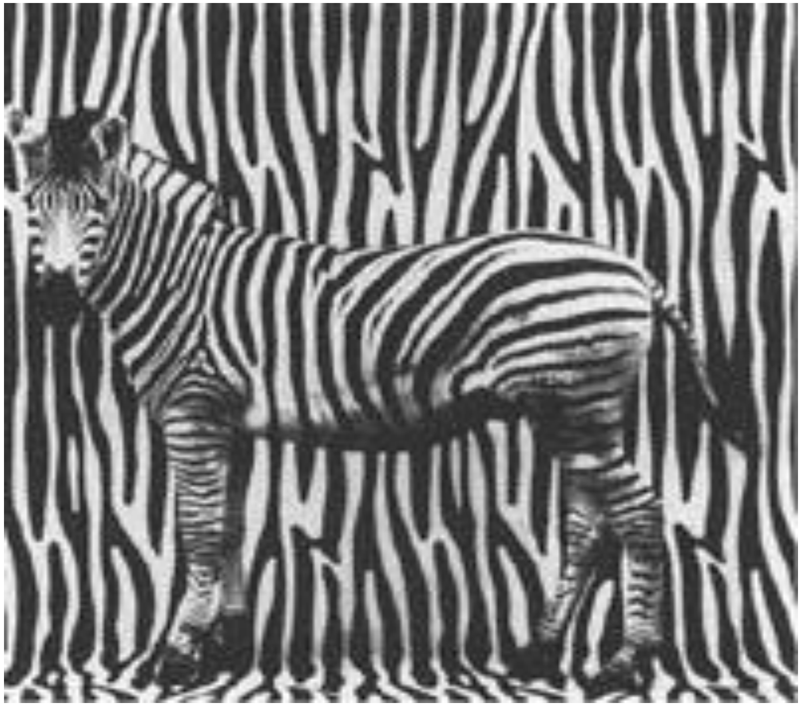
This is the well-known Kanizsa triangle

More illusory contours



Grouping by invisible completions

Two final examples



What role is top-down playing?

from http://www-static.cc.gatech.edu/classes/AY2007/cs4495_fall/html/materials.html

Questions?