Professor Brendan Morris, SEB 3216, brendan.morris@unlv.edu

ECG782: Multidimensional Digital Signal Processing

Spring 2014
TTh 14:30-15:45 CBC C313

Lecture 10
Segmentation
14/02/27

http://www.ee.unlv.edu/~b1morris/ecg782/

Outline

- Thresholding
  - Optimal thresholds
- Edge-Based Segmentation
  - Borders
  - Hough Transform
- Region-Based Segmentation
  - Merging and Splitting
- Template Matching
- Evaluation Issues

Segmentation

- Divide image into parts that correlate with objects or "world areas"
  - Important step for image analysis and understanding
- Complete segmentation
  - Disjoint regions corresponding to objects
  - $R = \bigcup_{i=1}^{S} R_i\,, \quad R_i \cap R_j = \emptyset, \quad i \neq j$
  - Typically requires high level domain knowledge
- Partial segmentation
  - Regions do not correspond directly to objects
  - Divide image based on homogeneity property
    - Brightness, color, texture, etc.
  - High-level info can take partial segmentation to complete
- Main goal is reduction in data volume for higher level processing

Segmentation Methods

- Global knowledge
  - Histogram of image features (e.g. intensity)
- Edge-based
- Region-based

- Edge and region are dual problems
  - Region defined by closed boundary (edges)
  - Use various characteristics
    - Brightness, texture, velocity field, etc.
    - Local properties

Thresholding

- Segment object from background
- $g(i,j) = \begin{cases} 1 & f(i,j) > T \\ 0 & f(i,j) \leq T \end{cases}$
    - T – threshold
    - 1 object and 0 background

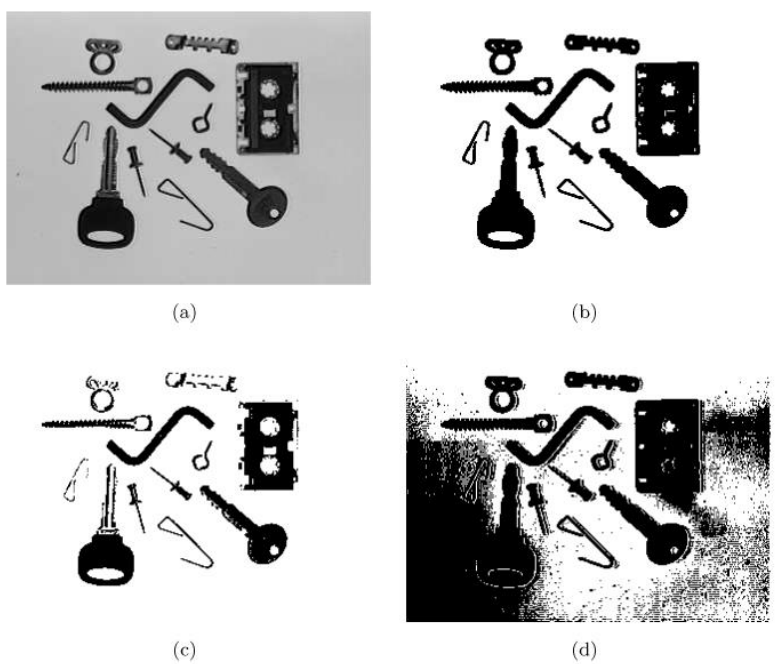


Figure 6.1: Image thresholding. (a) Original image. (b) Threshold segmentation. (c) Threshold too low. (d) Threshold too high. © *Cengage Learning 2015.*

- Requires the correct threshold of this to work
    - Difficulty to use a single global threshold
        - $T = \mathcal{T}(f)$
    - More often want adaptive threshold
        - $T = \mathcal{T}(f, f_c)$
            - f_c - is smaller image region (e.g. subimage)
- Many simple variants
    - Band thresholding - range of values for object
    - Multiband – multiple bands to give grayscale result

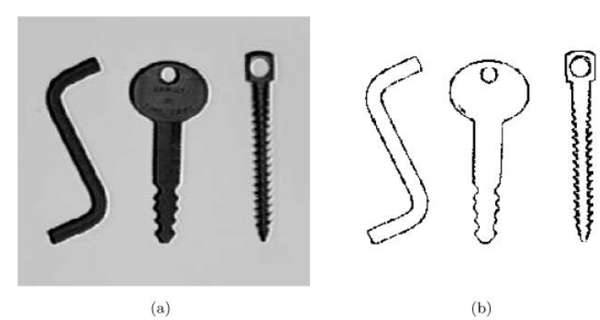


Figure 6.2: Image thresholding modification. (a) Original image. (b) Border detection using band-thresholding. © *Cengage Learning 2015.*

Threshold Detection Methods

- When objects are similar, the resulting histogram is bimodal
  - Objects one color and background another
  - Good threshold is between "peaks" in less probable intensity regions
    - Intuitively the lowest point between peaks

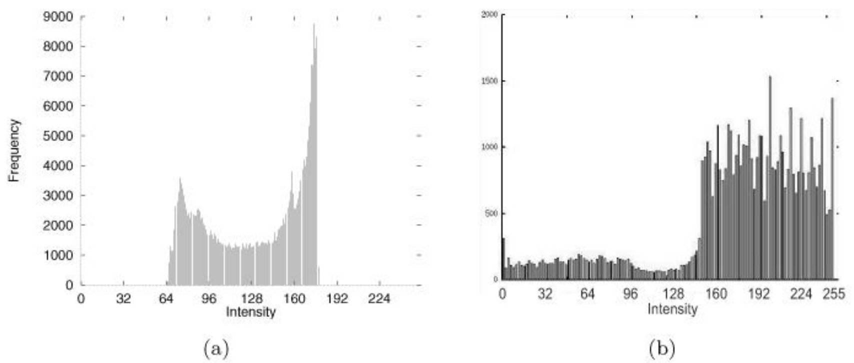


Figure 6.3: Bimodal histograms. (a) In cases with well-separable objects from the background, the shown histogram is clearly bimodal. (b) An example of a more shallow bimodal histogram (see top-left of Figure 6.5 for original image, in which the distinction between foreground and background has been deliberately perturbed). Note a wide, shallow peak whose distribution reaches from 0 to approximately 140, and a higher one more easily visible to the right. The distributions overlap in the gray-levels 100–160. © *Cengage Learning 2015.*

- In practice is difficult to tell if a distribution is bimodal
- There can be many local maxima
  - How should the correct one be selected?
- Notice also that since the histogram is global, a histogram for salt and pepper noise could be the same as for objects on background
- Should consider some local neighborhood when building the histogram
  - Account for edges

Optimal Thresholding

- Model the histogram as a weighted sum of normal probability densities
- Threshold selected to minimize segmentation error (minimum number of mislabeled pixels)
  - Gray level closest to minimum probability between normal maxima

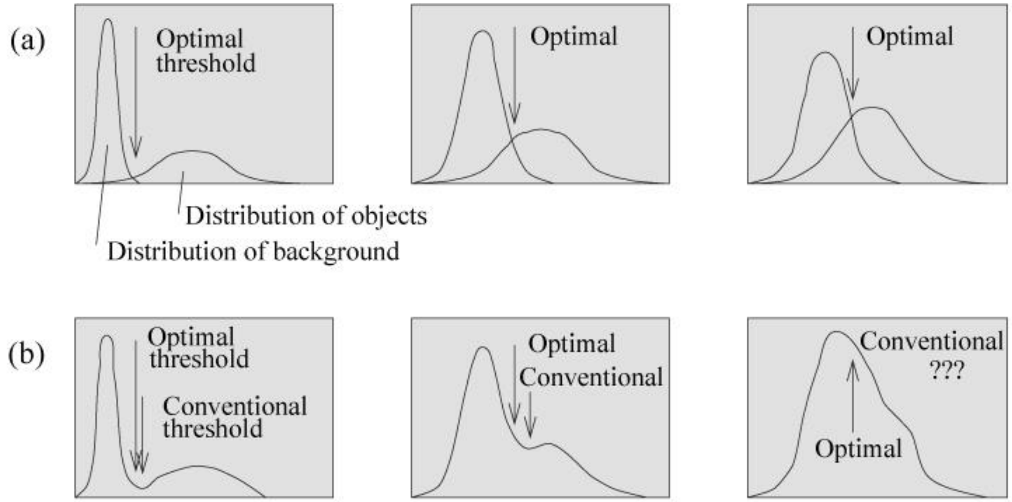


Figure 6.4: Gray-level histograms approximated by two normal distributions—the threshold is set to give minimum probability of segmentation error. (a) Probability distributions of background and objects. (b) Corresponding histograms and optimal threshold. © *Cengage Learning 2015.*

- Difficulties
  - Normal distribution assumption does not always hold
  - Hard to estimate normal parameters

- Useful tools:
  - Maximum-likelihood classification
  - Expectation maximization
    - Gaussian mixture modeling

Otsu's Algorithm

- Automatic threshold detection
  - Test all possible thresholds and find that which minimizes foreground/background variance
    - "Tightest" distributions

- Algorithm 6.2

1. Compute histogram H of image and normalize to make a probability
2. Apply thresholding at each gray-level t
   - Separate histogram into background B and foreground F
3. Compute the variance σ_B and σ_F
4. Compute probability of pixel being foreground or background
   - $w_B = \sum_{j=0}^{t} H(j)$
5. Select optimal threshold as
   - $\hat{t} = \min_t \sigma(t)$
   - $\sigma(t) = w_B \sigma_B(t) + w_F(t) \sigma_F(t)$

Figure 6.5: Top left, an image with artificially stretched white background—the image has also been showered with random noise. Top right, thresholded with Otsu's method: the histogram is shown in Figure 6.3, and the algorithm delivers $t = 130$. At bottom, the results of $t = 115, 130, 145$ on the trickiest part of the image; segmentation quality degrades very quickly.

Mixture Modeling

- Assume Gaussian distribution for each group
  - Defined by mean intensity and standard deviation
  - $h_{model}(g) = \sum_{i=1}^{n} a_i \exp\{-(g-\mu_i)^2/2\sigma_i^2\}$
- Determine parameters by minimizing mismatch between model and actual histogram with fit function
  - Match Gaussians to histogram
  - $F = \sum_{g \in G}\left(h_{model}(g) - h_{region}(g)\right)^2$
- Can use Otsu's as a starting guess
  - Limit search space

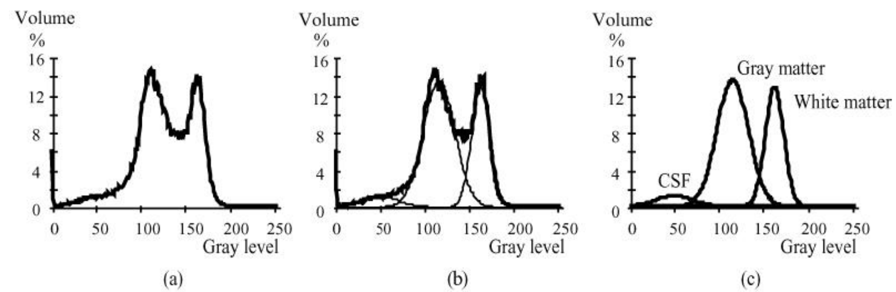


Figure 6.6: Segmentation of 3D T1-weighted MR brain image data using optimal thresholding. (a) Local gray-level histogram. (b) Fitted Gaussian distributions, global 3D image fit. (c) Gaussian distributions corresponding to WM, GM, and CSF. *Courtesy of R. J. Frank, T. J. Grabowski,*

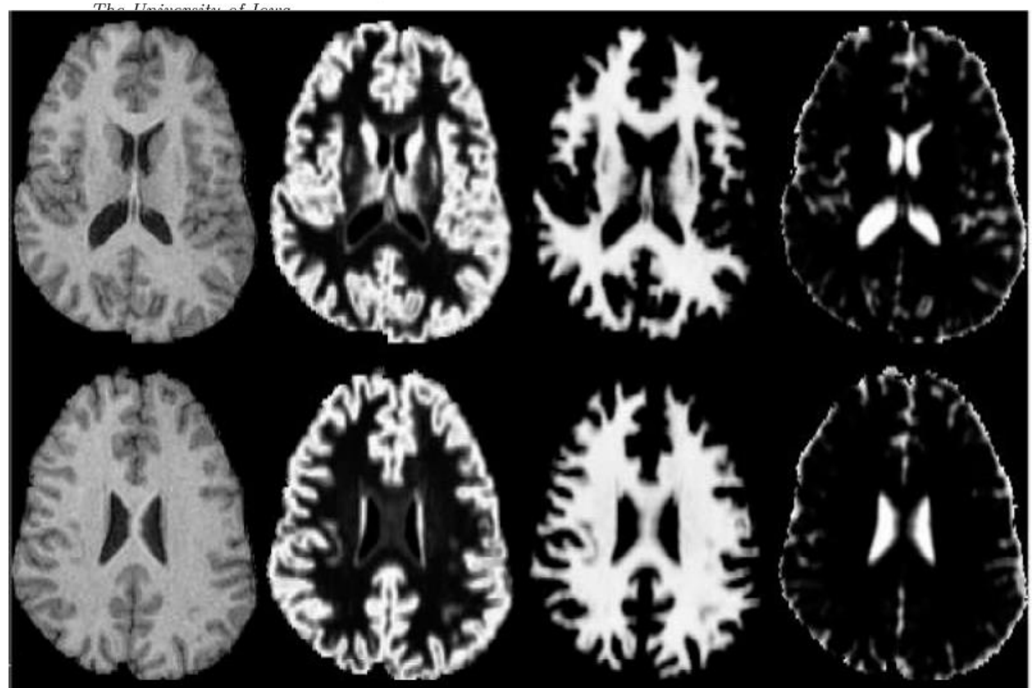
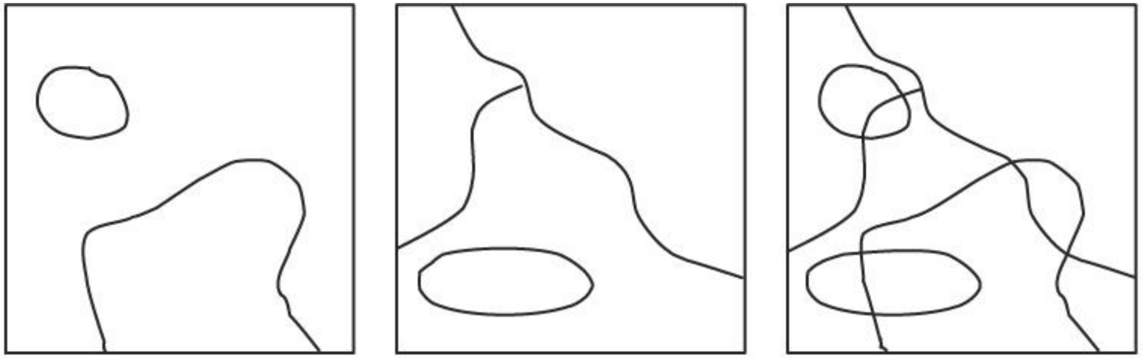


Figure 6.7: Optimal MR brain image segmentation. Left column: original T1-weighted MR images, two of 120 slices of the 3D volume. Middle left: Partial-volume maps of gray matter. The brighter the voxel, the higher is the partial volume percentage of gray matter in the voxel. Middle right: Partial-volume maps of white matter. Right column: Partial-volume maps of cerebro-spinal fluid. *Courtesy of R. J. Frank, T. J. Grabowski, The University of Iowa.*

Multi-Spectral Thresholding

- Compute thresholds in spectral bands independently and combine in a single image
  - Used for remote sensing (e.g. satellite images), MRI, etc.
- Algorithm 6.3

1. Compute histogram and segment between local minima on either side of maximum peak for each band
2. Combine segmentation regions into multispectral image
3. Repeat on multispectral regions until each region is unimodal



(a) (b) (c)

Figure 6.8: Recursive multi-spectral thresholding. (a) Band 1 thresholding. (b) Band 2 thresholding. (c) Multi-spectral segmentation. © *Cengage Learning 2015.*

Edge-Based Segmentation

- Early segmentation approach based off human perception
- Edge detecting operators are used to look for discontinuities in gray-level, color, texture, etc.
  - Edges must be further processed to better represent object borders as chains
- Better segmentation is available with prior knowledge
  - Can be used to evaluate confidence in segmentation
- Main problems are from noise or false edge response (edge where none exists, no edge where it exists)

Edge Image Thresholding

- Threshold based on edge magnitude
  - Still have difficulty determining a suitable threshold

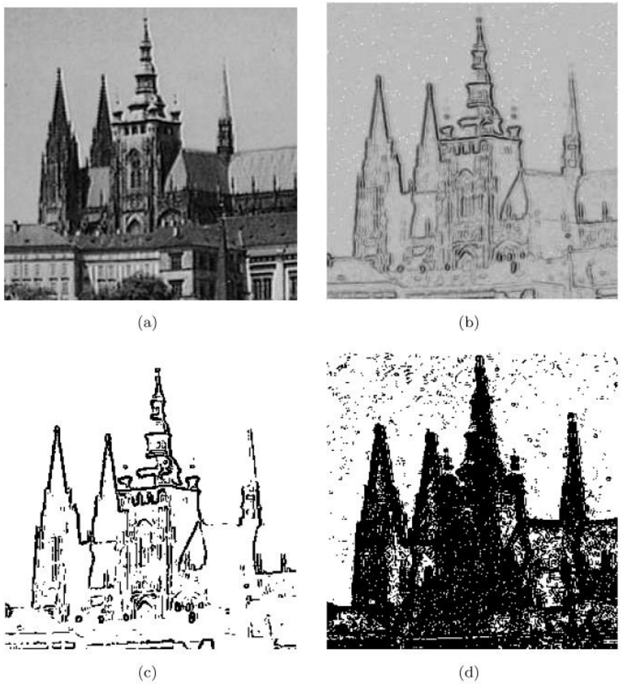


Figure 6.9: Edge image thresholding. (a) Original image. (b) Edge image (low contrast edges enhanced for display). (c) Edge image thresholded at 30. (d) Edge image thresholded at 10. © *Cengage Learning 2015.*

- Edges tend to get thickened using thresholds
- Can improve results:
  - Non-maximal suppression – to thin edges
  - Hysteresis thresholds – to remove noise and focus on long strong edges

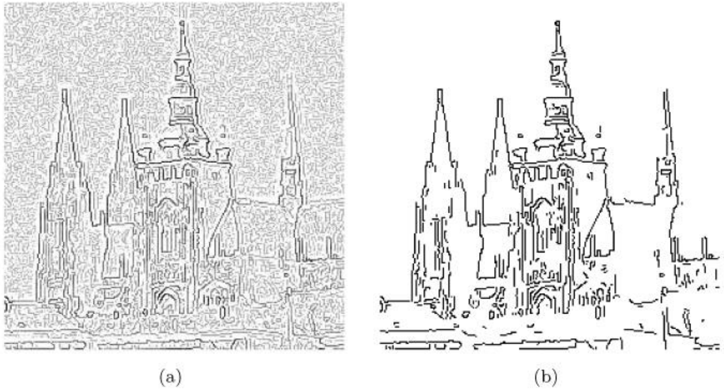
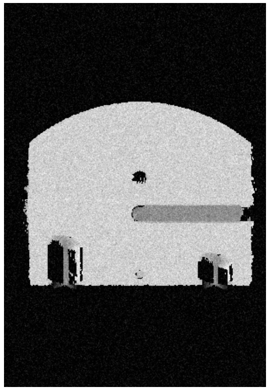


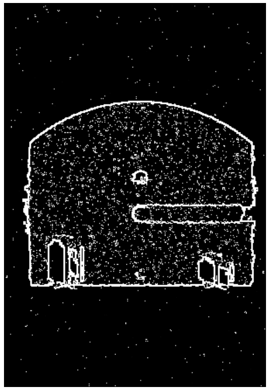
Figure 6.11: (a) Non-maximal suppression of the data in Figure 6.9b. (b) Hysteresis applied to (a); high threshold 70, low threshold 10. © *Cengage Learning 2015.*

Canny Edge Detection

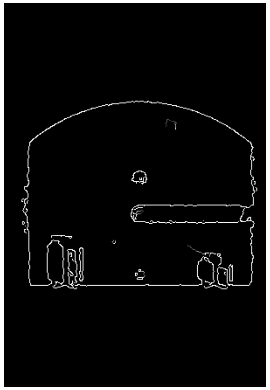
- Popular edge detection algorithm that produces a thin lines
- 1) Smooth with Gaussian kernel
- 2) Compute gradient
  - Determine magnitude and orientation (45 degree 8-connected neighborhood)



object



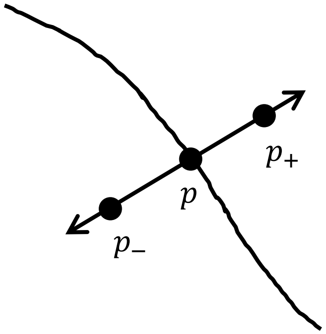
Sobel

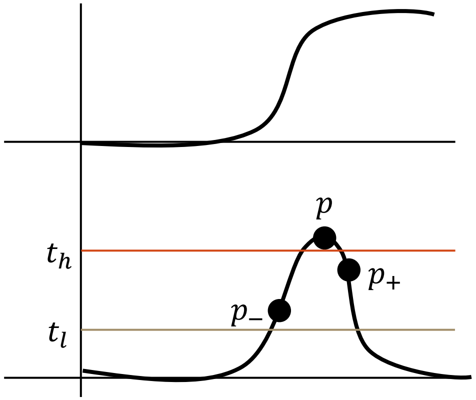


Canny

http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm

- 3) Use non-maximal suppression to get thin edges
  - Compare edge value to neighbor edgels in gradient direction





- 4) Use hysteresis thresholding to prevent streaking
  - High threshold to detect edge pixel, low threshold to trace the edge

Edge Relaxation

- Improve estimates by accounting for local relationships
  - Iterative update process
    - Typically only needs a few iterations
    - Can be computationally expensive, but parallelizable

- Edge detector is an initialization
- Use edge direction to indicate local neighbor pixels that are likely to be edges
  - Isolated responses are not likely to be an edge (continuity)

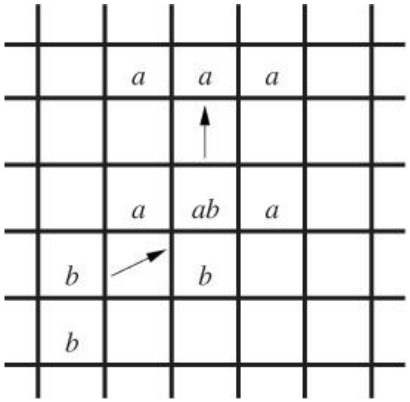


Figure 6.12: The principle of edge relaxation: two directed strong edges—oriented upwards (compass N) and up-right (compass ENE)—are indicated by arrows. Pixels marked *a* receive 'encouragement' for edges directed N; pixels marked *b* receive encouragement for edges directed ENE. Note one pixel receives encouragement to join the marked edges. © *Cengage Learning 2015.*

Border Tracing

- Use labeled image (regions) to find border edges
- Inner border
  - Part of the region
- Outer border
  - Not part of the region
  - Useful for region-based shape properties
    - Perimeter, compactness, etc.

- Border tracing considers neighbor pixels to add to lis border pixels

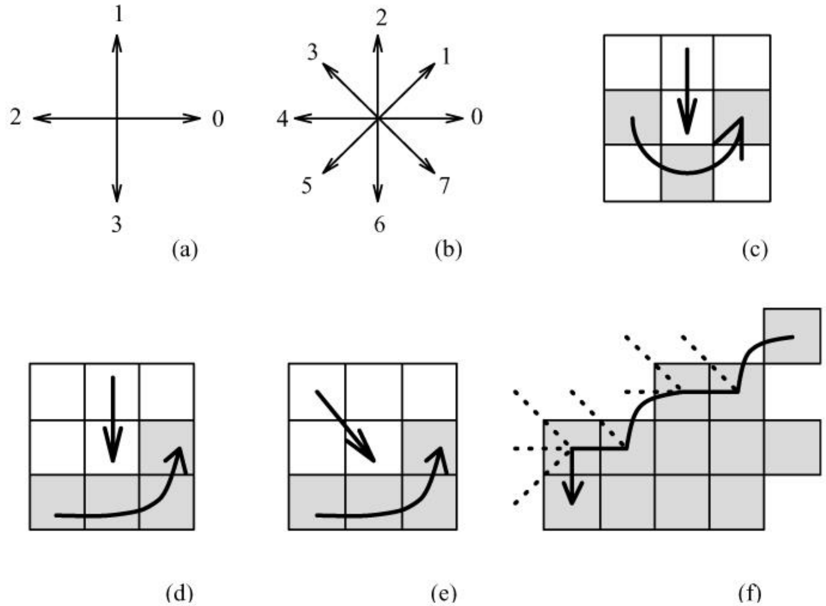


Figure 6.13: Inner boundary tracing. (a) Direction notation, 4-connectivity. (b) 8-connectivity. (c) Pixel neighborhood search sequence in 4-connectivity. (d), (e) Search sequence in 8-connectivity. (f) Boundary tracing in 8-connectivity (dotted lines show pixels tested during the border tracing). © *Cengage Learning 2015*.

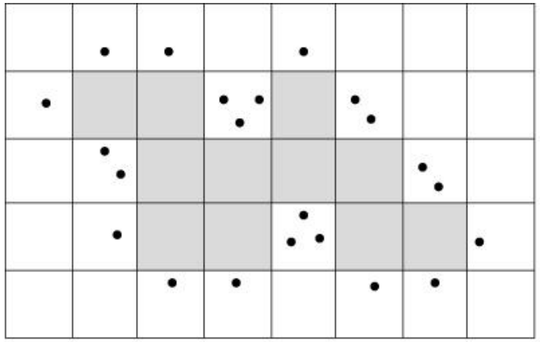


Figure 6.14: Outer boundary tracing; • denotes outer border elements. Note that some pixels may be listed several times. © *Cengage Learning 2015*.

Border Detection

- Various border detection extensions exist
- Extended borders – single common border
- Gray-level borders – path of high gradient pixels
  - Technique can be extended to multispectral images and temporal image sequences using multidimensional gradients
- Graph search – pixels are nodes and arcs (costs) are based of edge magnitude and direction
  - Borders are optimal path through weighted graph
  - Can be solved efficiently using dynamic programming

Hough Transform

- Segmentation viewed as the problem of finding objects
  - Must be of known size and shape
- Typically hard to do because of shape distortions
  - Rotation, zoom, occlusion
- Search for parameterized curves in image plane
  - $f(x, a) = 0$
    - a – n-dimensional vector of curve parameters
  - Each edge pixel "votes" for different parameters and need to find set with most votes

Hough Transform for Lines

- Original motivation for Hough transform
- Lines in the real-world can be broken, collinear, or occluded
  - Combine these collinear line segments into a larger extended line
- Hough transform creates a parameter space for the line
  - Every pixel votes for a family of lines passing through it
  - Potential lines are those bins (accumulator cells) with high count
- Uses global rather than local information

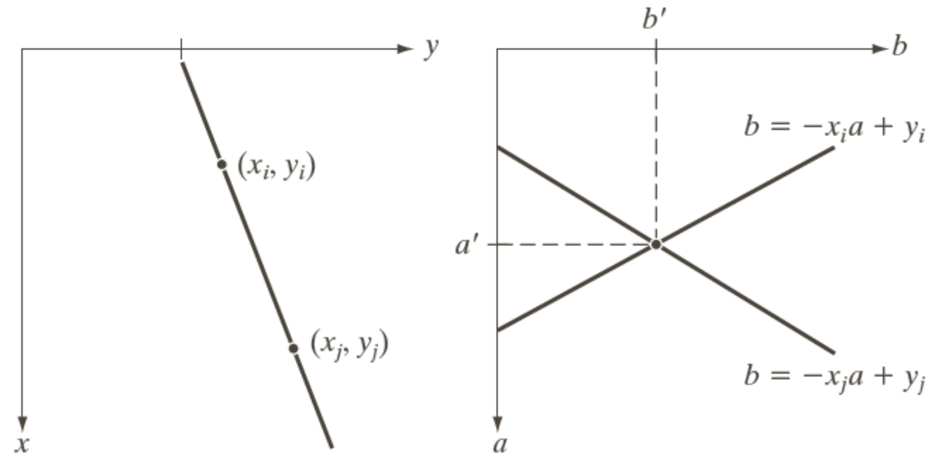
- See `hough.m, radon.m` in Matlab

Hough Transform Insight

- Want to search for all points that lie on a line
  - This is a large search (take two points and count the number of edgels)
- Infinite lines pass through a single point (x_i, y_i)
  - $y_i = ax_i + b$
    - Select any a, b
- Reparameterize
  - $b = -x_i a + y_i$
  - ab-space representation has single line defined by point (x_i, y_i)

a b

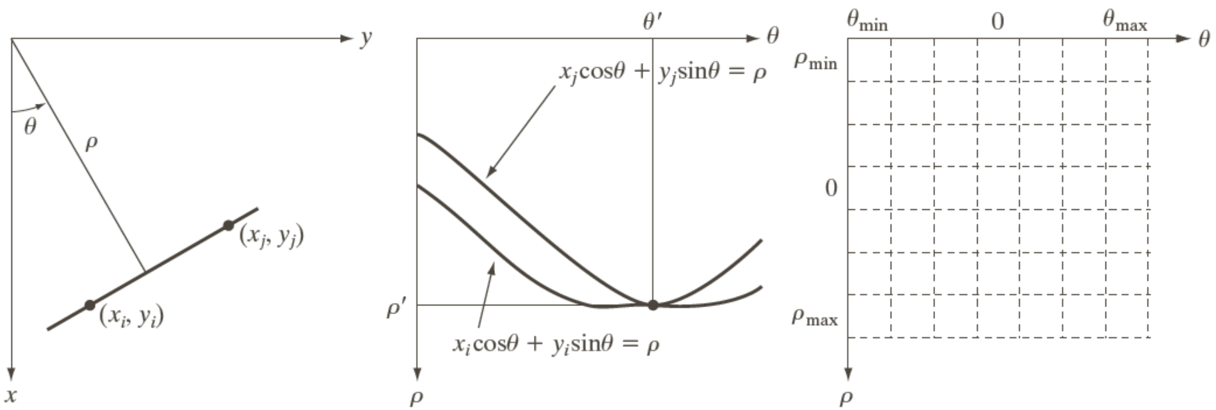
FIGURE 10.31 (a) xy-plane. (b) Parameter space.



- All points on a line will intersect in parameter space
  - Divide parameter space into cells/bins and accumulate votes across all a and b values for a particular point
  - Cells with high count are indicative of many points voting for the same line parameters (a, b)

Hough Transform in Practice

- Use a polar parameterization of a line – why?



- After finding bins of high count, need to verify edge
  - Find the extent of the edge (edges do not go across the whole image)

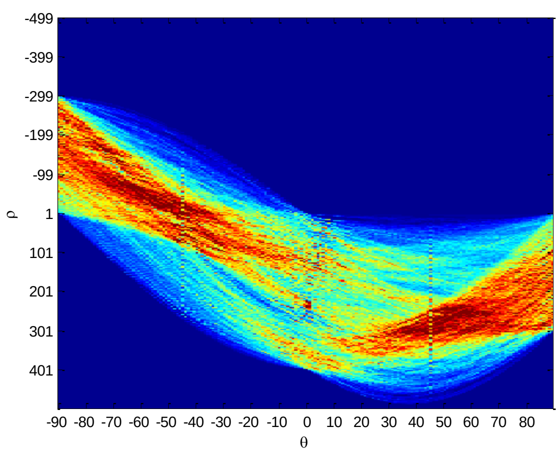
- This technique can be extended to other shapes like circles

Hough Transform Example I

Input image

Grayscale

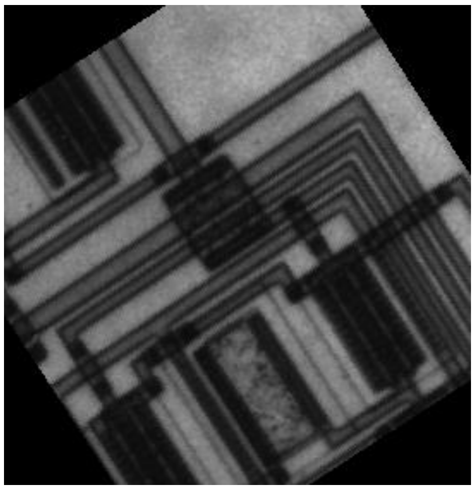
Canny edge image

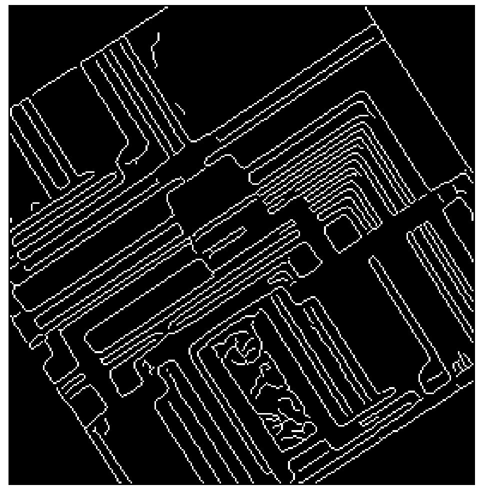


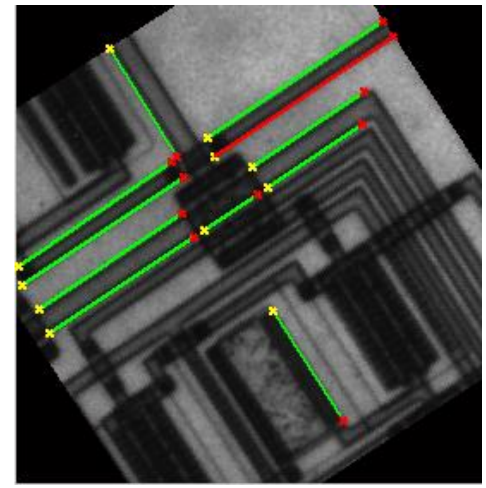
Hough space

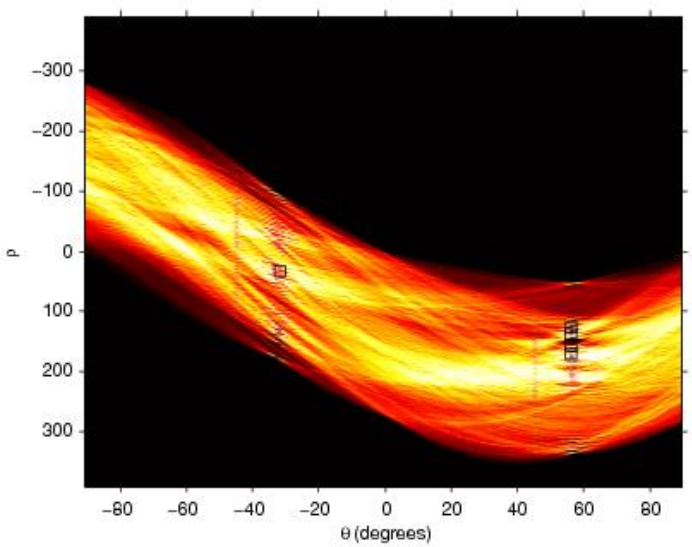
Top edges

Hough Transform Example II









http://www.mathworks.com/help/images/analyzing-images.html

Hough Transform for Circles

- Consider equation of circle
  - $(x_1 - a)^2 + (x_2 - b)^2 = r^2$
    - (a, b) – center of circle
    - r – radius

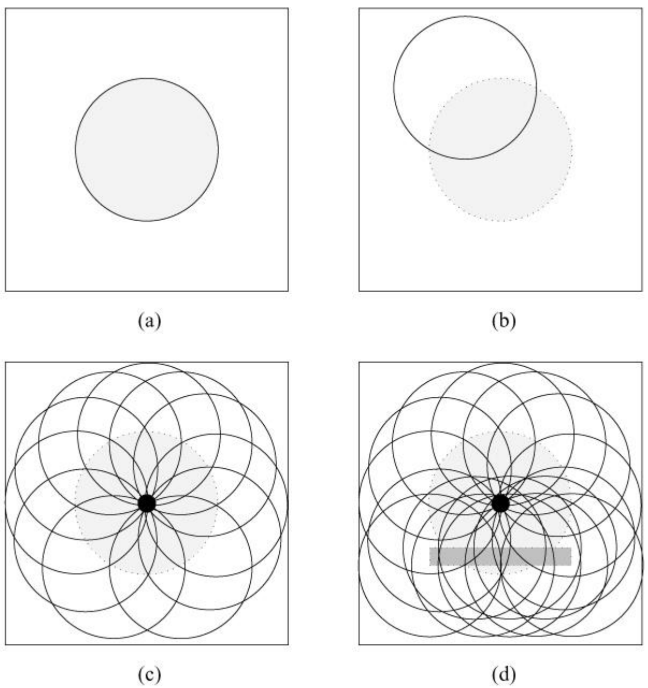


Figure 6.31: Hough transform—example of circle detection. (a) Image of a dark circle, of known radius r, on a bright background. (b) For each dark pixel, a potential circle-center locus is defined by a circle with radius r and center at that pixel. (c) The frequency with which image pixels occur 0n circle-center loci is determined—the highest-frequency pixel represents the center of the circle (marked by •). (d) The Hough transform correctly detects the circle (marked by •) in the presence of incomplete circle information and overlapping structures. (See Figure 6.32 for a real-life example.) © *Cengage Learning 2015.*

- Each edgel votes for a circle of radius r at center (a, b)
- Accumulator array is now 3-dimensional
  - Usually for fixed radius circle

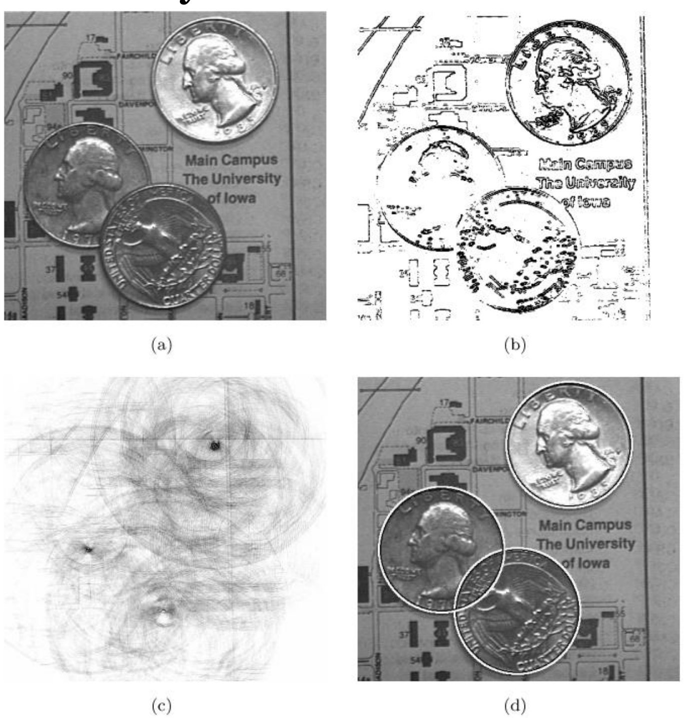


Figure 6.32: Hough transform—circle detection. (a) Original image. (b) Edge image (note that the edge information is far from perfect). (c) Parameter space. (d) Detected circles. © *Cengage Learning 2015.*

Hough Transform Considerations

- Practical only for 3-dimensions
  - Exponential growth of accumulator array
- Use gradient information to simplify process
  - Only accumulate limited number of bins
  - Accounts for local consistency constraints
    - Line pixels should be in edge direction (orthogonal to gradient direction)
- Weight accumulator by edge magnitude
  - Consider only the strongest edges
- "Back project" strongest accumulator cells of each pixel to remove other votes
  - Sharpen accumulator response
- Line tracing
  - Find endpoints of line

Region-Based Segmentation

- Regions are areas defined inside of borders
  - Simple to go back and forth between both
  - However, segmentation techniques differ
- Region growing techniques are typically better in noisy images
  - Borders are difficult to detect
- A region is defined by a homogeneity constraint
  - Gray-level, color, texture, shape, model, etc.
  - Each individual region is homogeneous
  - Any two regions together are not homogeneous

Region Merging

- Start with each pixel as a region and combine regions with a merge criterion
  - Defined over adjacent regions (neighborhood)
- Be aware the merging results can be different depending on the order of the merging
  - Prior merges change region relationships

- Simplest merge methods compute statistics over small regions (e.g. 2×2 pixels)
  - Gray-level histogram used for matching

Region Merging Via Boundary Melting

- Utilize crack information (edges between pixels)
- Merge regions if there are weak crack edges between them

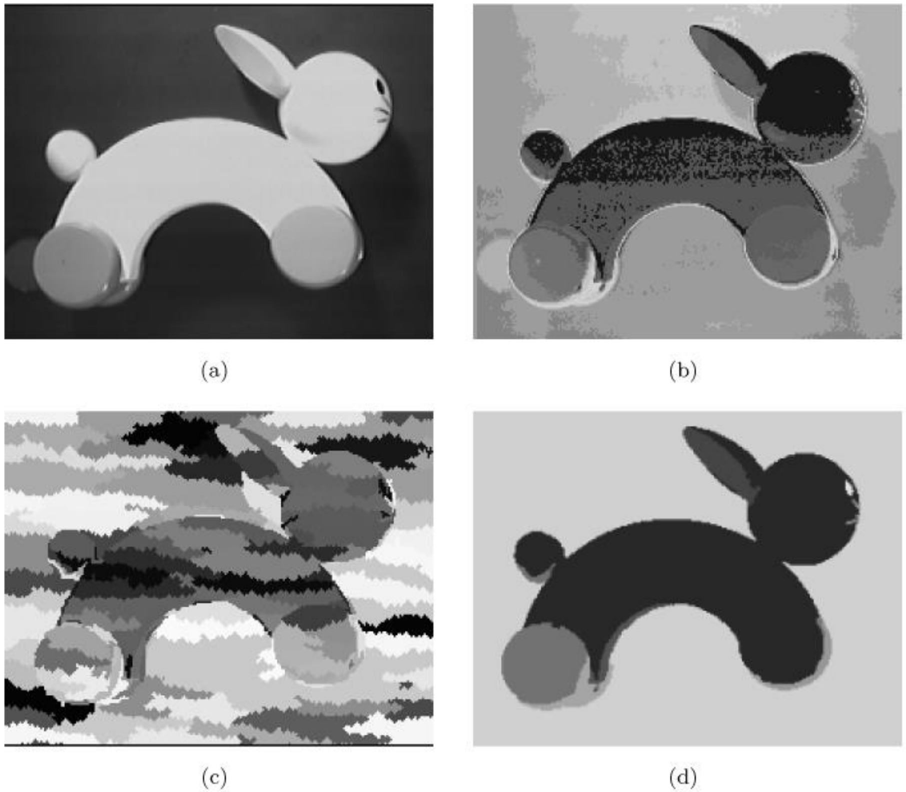


Figure 6.40: Region merging segmentation. (a) Original image. (b) Pseudo-color representation of the original image (in grayscale). (c) Recursive region merging. (d) Region merging via boundary melting. *Courtesy of R. Marik, Czech Technical University.*

Region Splitting

- Opposite of region merging
  - Start with full image as single region and split to satisfy homogeneity criterion
- Merging and splitting do not result in the same regions
  - A homogenous split region may never have been grown from smaller regions
- Use same homogeneity criteria as in region merging

Split and Merge

- Try to obtain advantages of both merging and splitting
- Operate on pyramid images
  - Regions are squares that correspond to pyramid level
  - Lowest level are pixels
- Regions in a pyramid level that are not homogeneous are split into four subregions
  - Represent higher resolution a level below
- 4 similar regions are merged into a single region at higher pyramid level
- Segmentation creates a quadtree
  - Each leaf node represents a homogenous region
    - E.g. an element in a pyramid level
  - Number of leaf nodes are number of regions

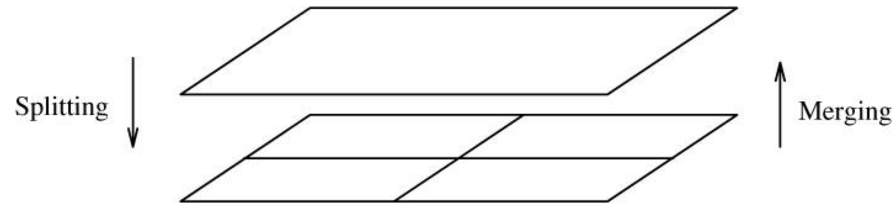


Figure 6.41: Split-and-merge in a hierarchical data structure. © *Cengage Learning 2015.*

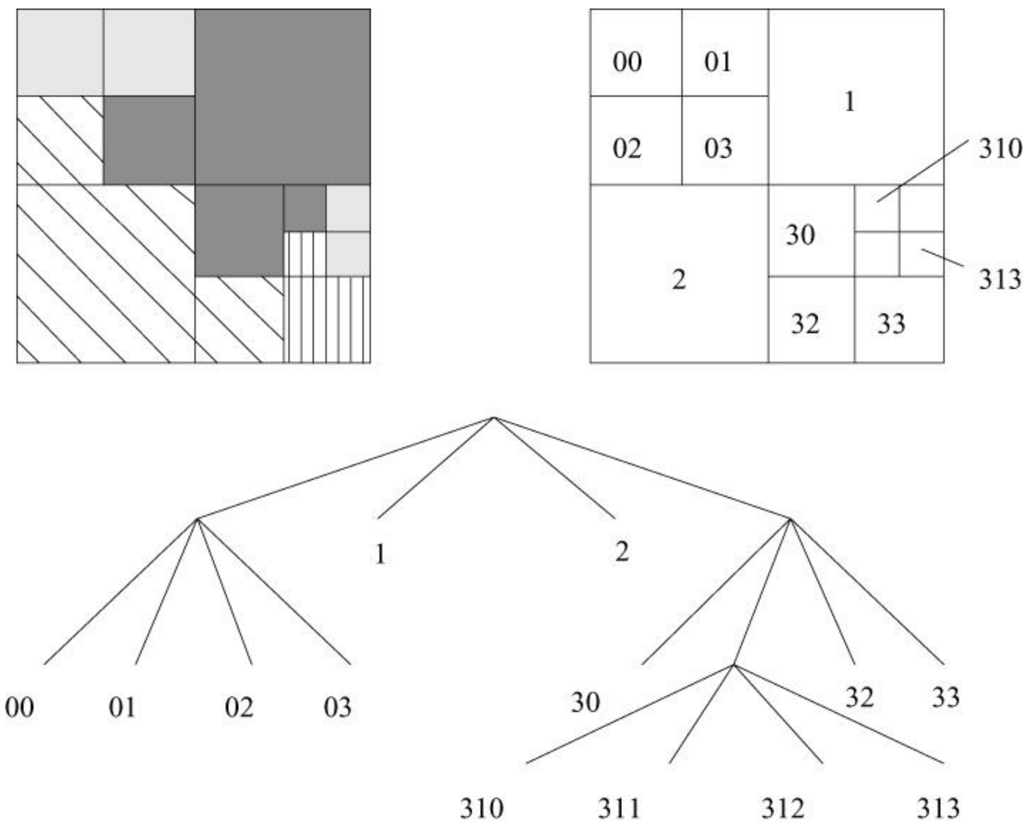
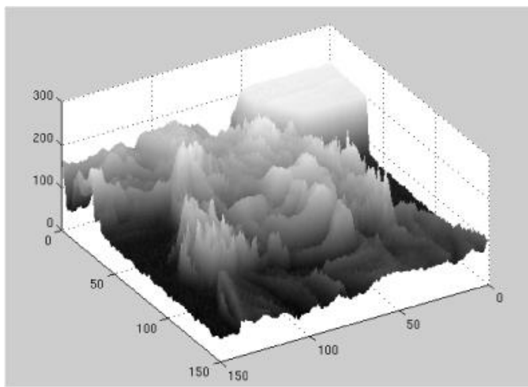


Figure 6.42: Segmentation quadtree. © *Cengage Learning 2015.*

Watershed Segmentation

- Topography concepts
    - Watersheds are lines dividing catchment basins
- Region edges correspond to high watersheds
- Low gradient areas correspond to catchment basins
    - All pixels in a basin are simply connected and homogeneous because they share the same minimum



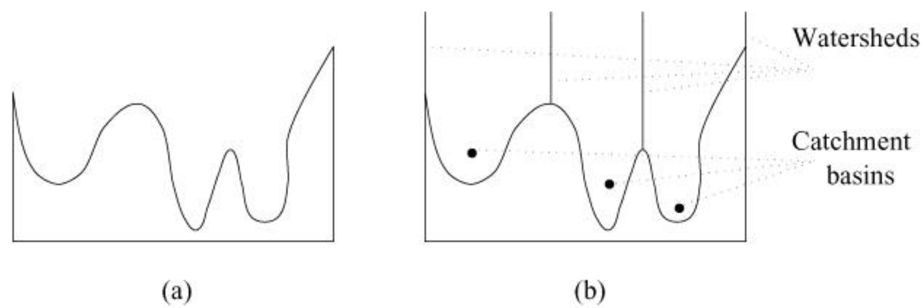


Figure 6.44: One-dimensional example of watershed segmentation. (a) Gray-level profile of image data. (b) Watershed segmentation—local minima of gray-level (altitude) yield catchment basins, local maxima define the watershed lines. © *Cengage Learning 2015.*

Watershed Computation

- Can build watersheds by examining gray-level values from lowest to highest
- Watersheds form when catchment basins merge

- Raw watershed results in oversegmentation
- Use of region markers can improve performance
  - Matlab tutorial

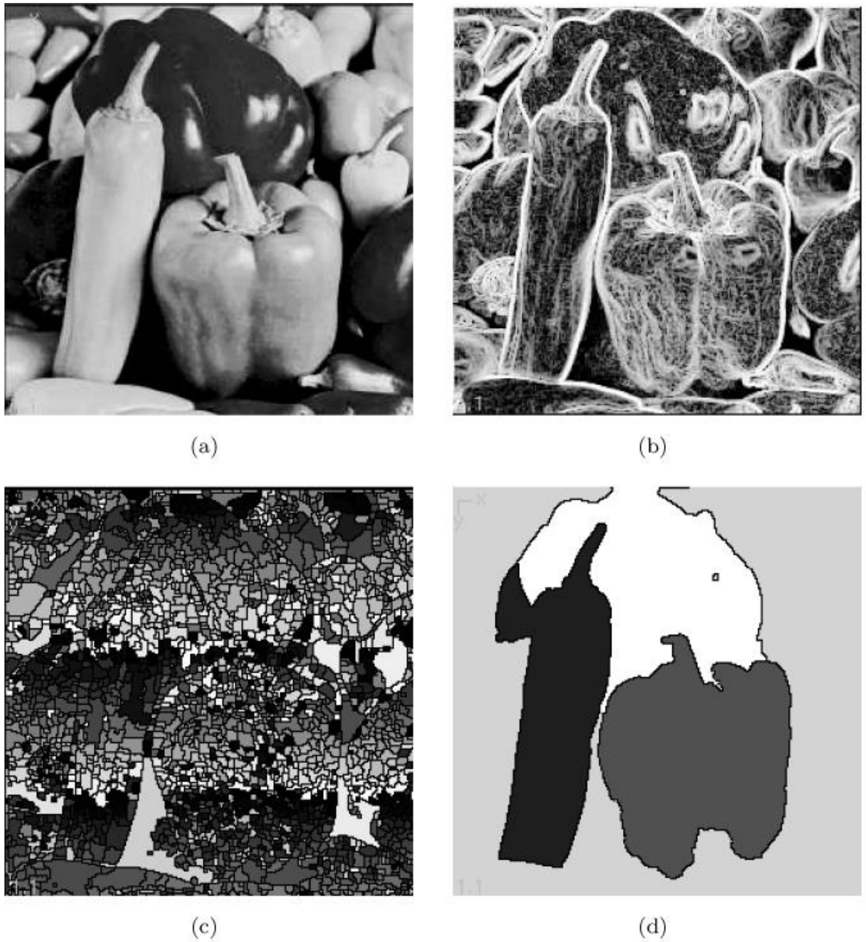


Figure 6.46: Watershed segmentation. (a) Original;. (b) Gradient image, 3 × 3 Sobel edge detection, histogram equalized. (c) Raw watershed segmentation. (d) Watershed segmentation using region markers to control oversegmentation. *Courtesy of W. Higgins, Penn State University.*

Matching

- Basic approach to segmentation by locating known objects (search for patterns)
  - Generally have a model for object of interest
- Various examples of matching
  - Different sophistication
- Optical character recognition (OCR)
  - Template matching when font is known and image carefully aligned
- Font-independent OCR
  - Match pattern of character
- Face recognition
  - Match pattern of face to image
  - More variability in appearance
- Pedestrian behavior matching
  - Explain what a pedestrian is doing

Template Matching

- Try to find template image in larger test image
- Minimize error between image and shifted template

$$E(\mathrm{x}) = \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j} - I_{x_a+i,x_b+j})^2 = 0\,, \tag{6.29}$$

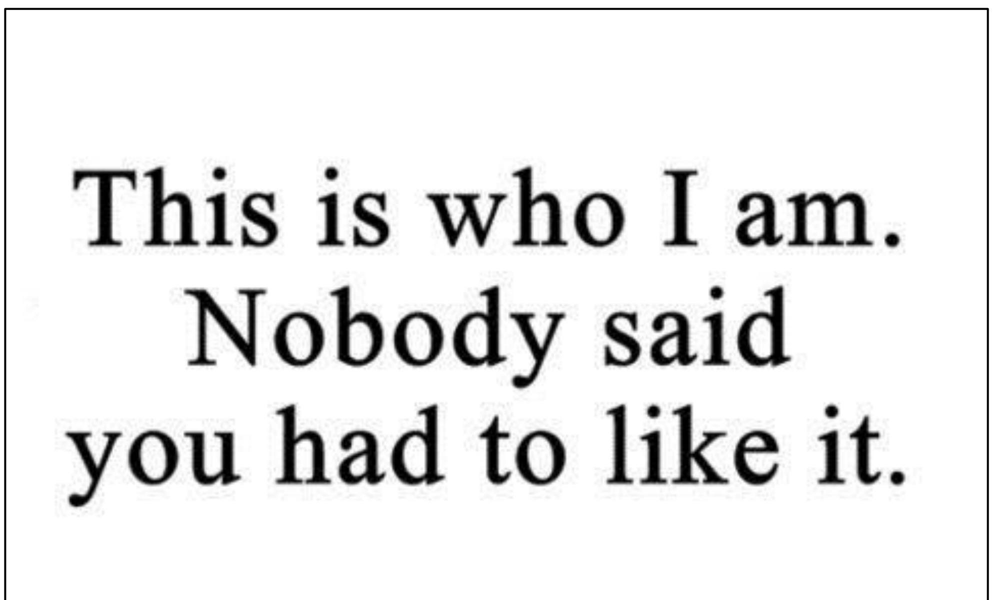
$$\begin{aligned} E(\mathrm{x}) &= \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j} - I_{x_a+i,x_b+j})^2 \\ &= \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j})^2 - 2 \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j} I_{x_a+i,x_b+j}) + \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (I_{x_a+i,x_b+j})^2\,, \end{aligned} \tag{6.30}$$

- First term is a constant and the last term changes slowly so only the middle term needs to be maximized

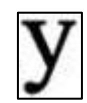
$$Corr_T(\mathrm{x}) = \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j} I_{x_a+i,x_b+j})\,. \tag{6.31}$$

Binary Filtering as Detection

- Filtering (correlation) can be used a simple object detector
  - Mask provides a search template
  - “Matched filter” – kernels look like the effects they are intended to find



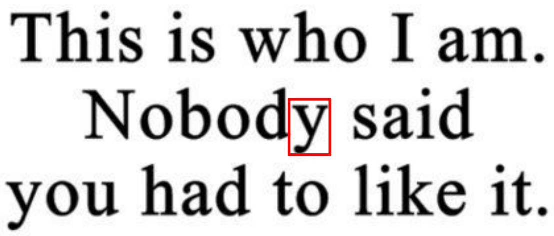
image



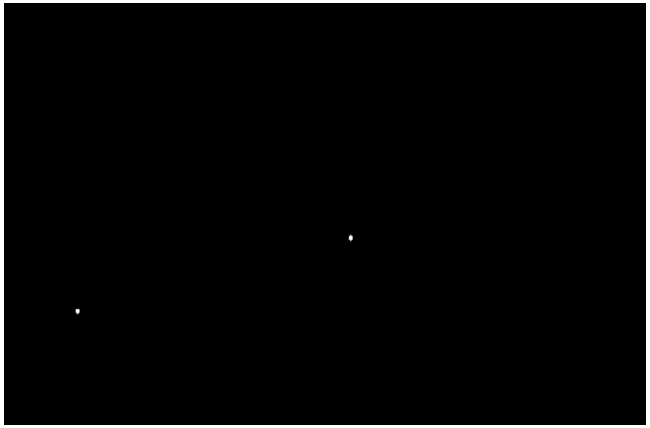
template

Correlation Masking

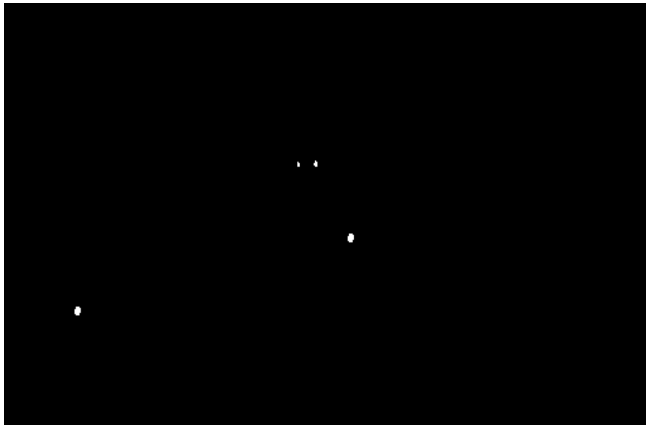
correlation



detected letter



0.9 max threshold

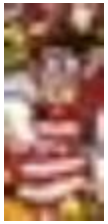


0.5 max threshold

Normalized Cross-Correlation

- Extension to intensity values
  - Handle variation in template and image brightness

scene

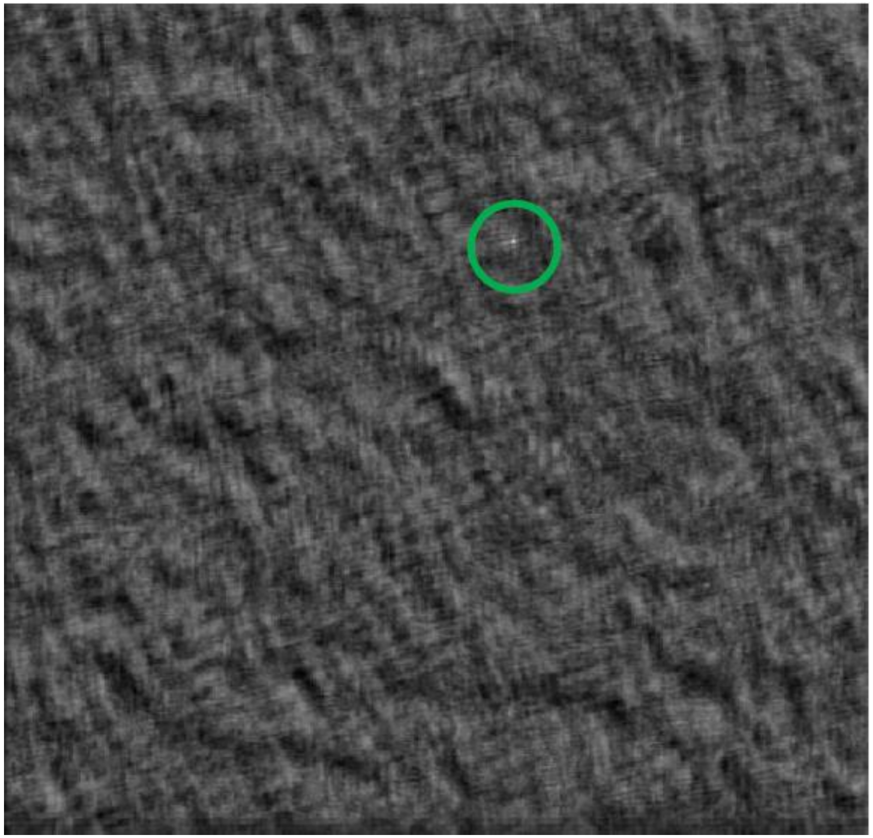


template

Adapted from
http://kurser.iha.dk/ee-ict-master/ticovi/

Where's Waldo

Detected template



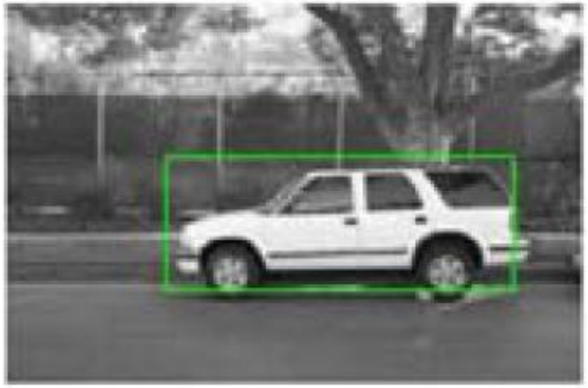
correlation map

Adapted from
http://kurser.iha.dk/ee-ict-master/ticovi/

Detection of Similar Objects

- Previous examples are detecting exactly what we want to find
  - Give the perfect template
- What happens with similar objects

Template



Detected template

- Works fine when scale, orientation, and general orientation are matched

- What to do with different sized objects, new scenes

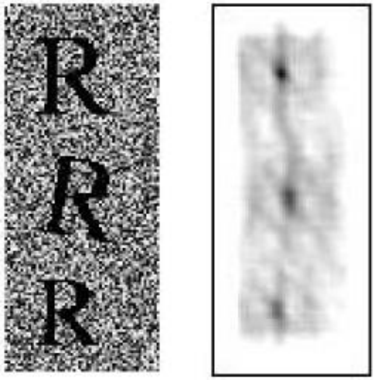


Figure 6.48: Template matching: A template of the letter **R** is sought in an image that has itself, a slightly rotated version, and a smaller version. The correlation response (contrast stretched for display) illustrates the diffuse response seen for even small adjustments to the original. © *Cengage Learning 2015.*

Adapted from K. Grauman

Template Matching Strategies

- Detection of parts
  - Full “pixel perfect” match may not exist, but smaller subparts may be matched
  - Connect subparts through elastic links
- Search at scale
  - Pattern matching is highly correlated in space
    - Neighborhoods around match have similar response
  - Search at low resolution first and go to higher resolution for refinement
    - Less comparisons, much faster
- Quit sure mismatches quickly
  - Do not compute full correlation when error is too large
  - Matches are rare so only spend time on heavy computation when required (cascade classifier later)

Evaluating Segmentations

- Need to know what is the "right" segmentation and then measure how close and algorithm matches
- Supervised approaches
  - Use "expert" opinions to specify segmentation
  - Evaluate by:
    - Mutual overlap
    - Border position errors (Hausdorff set distance)
- Unsupervised approaches
  - No direct knowledge of true segmentation
    - Avoid label ambiguity
  - Define criterion to evaluate region similarity and inter-region dissimilarity

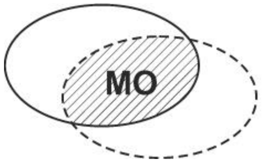


Figure 6.52: Mutual overlap: machine segmented region in solid, ground truth in dashed. © *Cengage Learning 2015.*

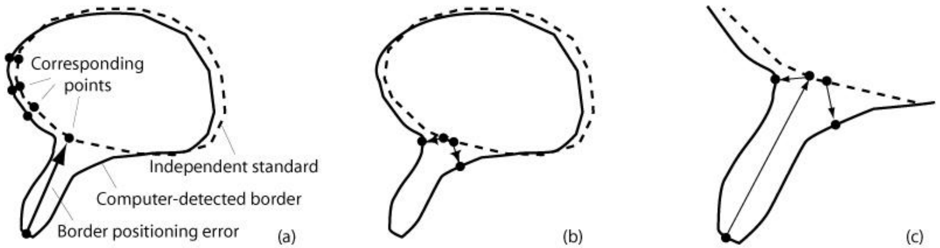


Figure 6.53: Border positioning errors. (a) Border positioning errors are computed as directed distances between the computer-determined and correct borders. (b) If errors are calculated in the opposite direction (from ground truth to the computer-determined border), a substantially different answer may result. (c) Zoomed area showing the difference in calculating directional errors. © *Cengage Learning 2015.*

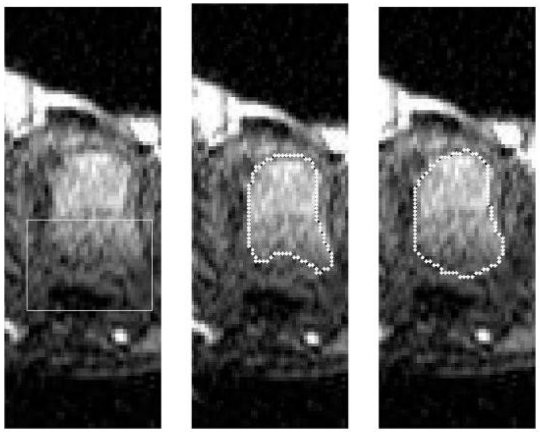


Figure 6.51: A region from a dynamically enhanced MRI study with partly ambiguous boundary. Two different experts have overlaid their judgments. *Courtesy of O. Kubassova, S. Tanner, University of Leeds.*