Lecture : Watershed Segmentation

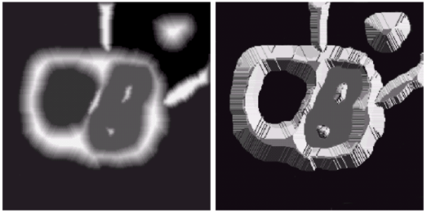
Dr. V Masilamani
masila@iiitdm.ac.in

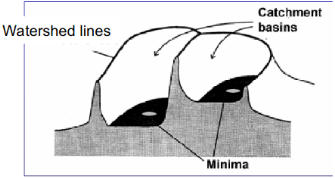
Department of Computer Science and Engineering
IIITDM Kancheepuram
Chennai-127

Watershed segmentation

- In watershed segmentation algorithm, the process of constructing dam(to stop merging two water bodies) is mimicked on images
- Consider a landscape, and visualize the topogrphy of the landscape
- In such a topographic view, there are 3 types of points when rain water drop falls:
  - Water drop falls at regional minima(topgraphycally) point P
  - When a drop of water fall at a point P, the water drop flows to a single minimum. (The catchment point)
  - When a drop of water fall at a point P, the water drop would be equally likely to flow to more than one minimum. (The divide line point or watershed line point or dam point.)
- The set of all dam points is called as dam or watershed line or divide line

Watershed segmentation (cont.)

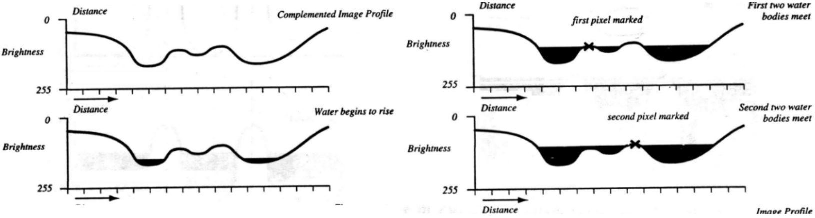




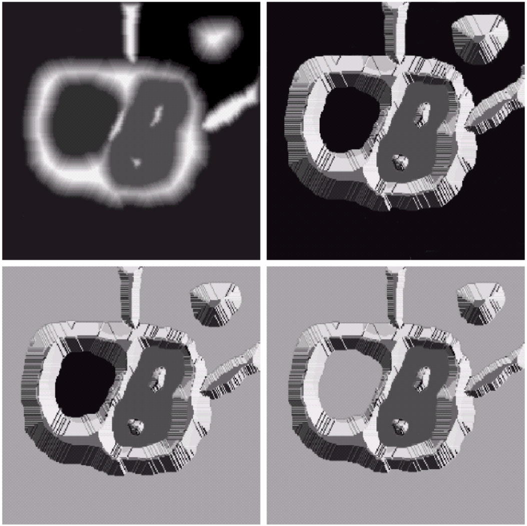
Watershed segmentation (cont.)

How to find watershed lines, given the topography?

- The idea is simple:
  - Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate.
  - When the rising water in distinct catchment basins is about to merge, a dam is built to prevent merging.
- 1D-profile of the topography along with water filling and dam construction are shown below



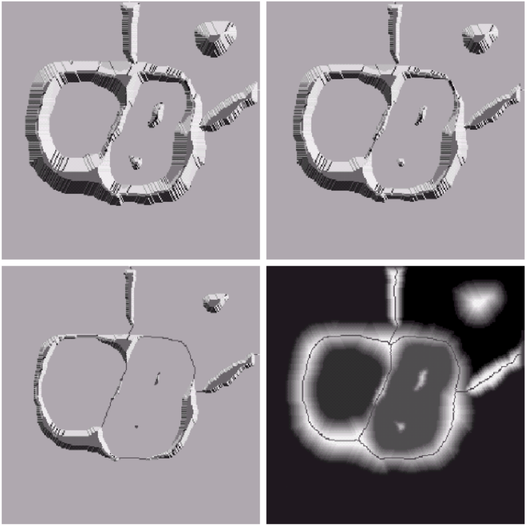
Watershed segmentation (cont.)



a b
c d

FIGURE 10.44 (a) Original image. (b) Topographic view. (c)–(d) Two stages of flooding.

Watershed segmentation (cont.)



e f
g h

FIGURE 10.44 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed segmentation (cont.)

To mimic the process of building dams as explained above for segmenting images

- Visualize an image in 3D: pixel location(2D) and pixel value(third dimension)

Watershed Segmentation Algorithm

- Find pixels with local minima's
- For each of the local minima, assign distinct labels
- For each intensity level k=0 to Max
    - For each unlabelled neighbour p of each of the labelled pixels
        - If p is adjacent to pixel with a single label l, then assign label l to p
        - Else(adjacent to at least two pixels with different labels), Do not label p, and call that p as a point in the watershed line or dam point.
- Output all pixels their corresponding labels, and also the set all of dam points

Watershed segmentation (cont.)

- Can we use watershed segmentation algorithm directly on the input gray scale image
  - No, It deos not produce desired results
- Watershed algorithm can be used on the gradient image instead of the original image.
  - As per the underlying assumption each object corresponds to one catchment area
  - The gradient at the bottom of catchment area is zero

Watershed segmentation (cont.)

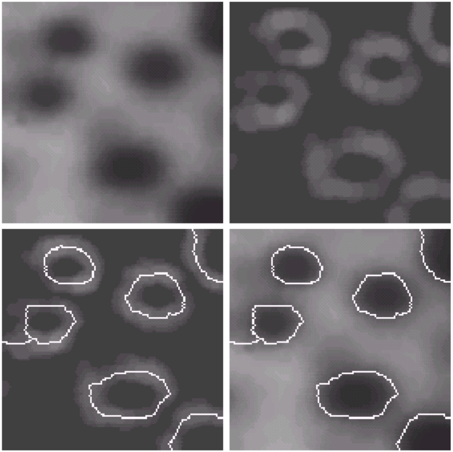


FIGURE 10.46 (a) Image of blobs. (b) Image gradient. (c) Watershed lines. (d) Watershed lines superimposed on original image. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed segmentation (cont.)

Due to noise and other local irregularities of the gradient, oversegmentation might occur.

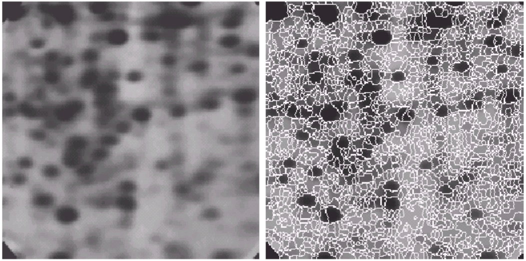
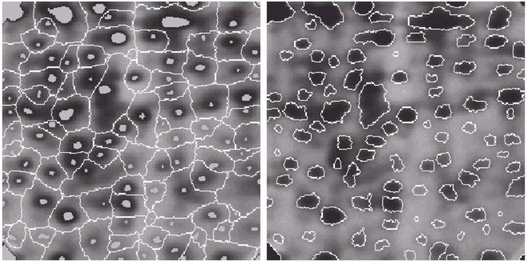


FIGURE 10.47 (a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed segmentation (cont.)

- How to avoid oversegmentation
  - Limit number of local minimas
  - One way to limit number of local minimas is to use **markers**
    - Marker is a connected component with similar(close) values in the gradient image
  - Consider all points in the same marker as the same local minima (they all need to be labelled with the same label)

Watershed segmentation (cont.)



a b

FIGURE 10.48 (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed Segmentation without using gradient

Step 1: Convert to binary image such that the foreground is in white, and the background is in black

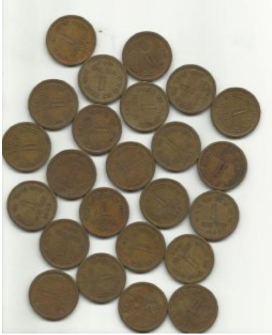


Figure 1: Input Image

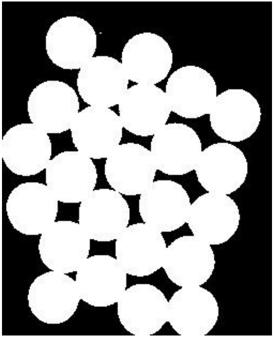


Figure 2: Binary Image with foreground in white

Watershed Segmentation without using gradient (cont.)

Step 2: Remove noise using opening operation

Step 3: Find sure-Foreground(Fg) and sure-Background(Bg) on output of step 2

- To find sure-background, do dilation(black pixels are sure Bg)
- To find sure-Fg
  do erosion(white pixels are sure Fg) (works well when objects are not touching)
  or
  Thresholded distance transform output(works well when objects are touching), where distance transform: min d((x,y), Bg) for each pixel (x,y)

Watershed Segmentation without using gradient (cont.)

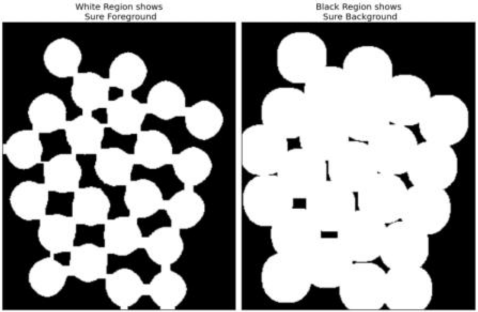


Figure 3: Left: Eroded; Right: Dilated

Watershed Segmentation without using gradient (cont.)

Observation after step3:

- There are three types of pixels
  - Sure Fg pixels
  - Sure Bg pixels
  - Non-labeled pixels
- The Non-labeled pixels will be classified to Fg or Bg by the watershed segmentation algorithm

Watershed Segmentation without using gradient (cont.)

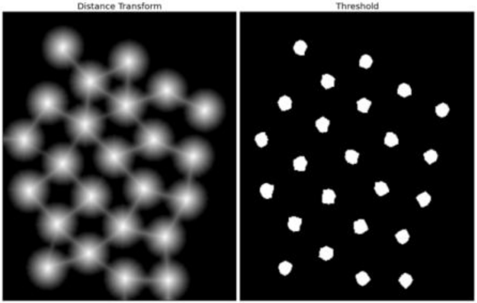


Figure 4: Left: Distance Transform; Right: Thresholded Distance Transform

Watershed Segmentation without using gradient (cont.)

Step 4: Find the connected components in the sure-Fg image

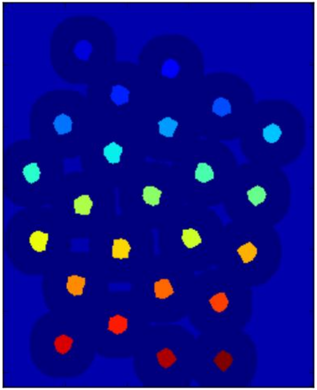


Figure 5: Each connected component is shown in different colour

Watershed Segmentation without using gradient (cont.)

Step 5: Apply watershed algorithm, keeping each marker as local minima

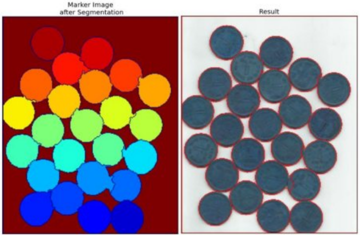


Figure 6: Each connected component is shown in different colour