

Topic 3: exercise 06

Region-Based Segmentation

1. Segmentación basada en regiones, watershed

La Transformada Watershed aplica conceptos topológicos para segmentar una imagen, viendo esta, en niveles de gris, como una superficie con embalses (catchment basins), mínimos de la imagen, que se van llenando de agua. La transformada devuelve las presas o diques (watershed lines) que habría que poner para que el agua no pasase de un embalse a otro.

Entender la Transformada Watershed requiere que se piense en una imagen como una superficie. Por ejemplo, considere la imagen de abajo:

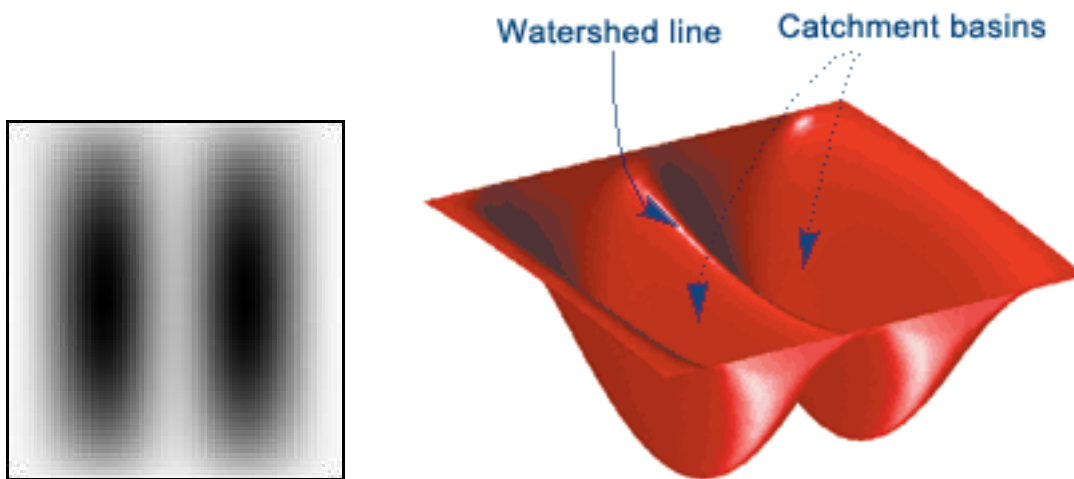


Figura 1. Imagen generada sintéticamente de dos manchas oscuras y su interpretación por parte del algoritmo watershed.

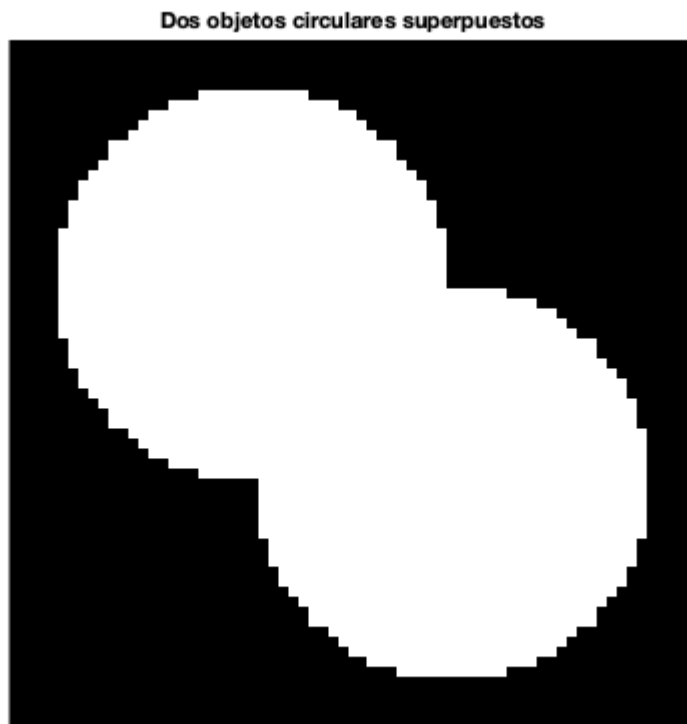
Si imaginas que las áreas brillantes son "altas" y las oscuras "bajas", entonces podría parecerse a la superficie de la derecha. Con las superficies, es natural pensar en términos de embalses (catchment basins) y de diques (watershed lines). La clave para utilizar la función watershed para la segmentación es ésta: Cambie su imagen en otra imagen cuyas embalses (zonas oscuras) sean los objetos que desea identificar.

Este algoritmo está implementado en Matlab mediante la función **watershed**, que se aplicaría a una imagen cuyos objetos a segmentar serían los valores mínimos de la imagen.

```
close all;
clear all;
clc;
```

Se crea una imagen binaria que contenga dos objetos circulares superpuestos y se muestran

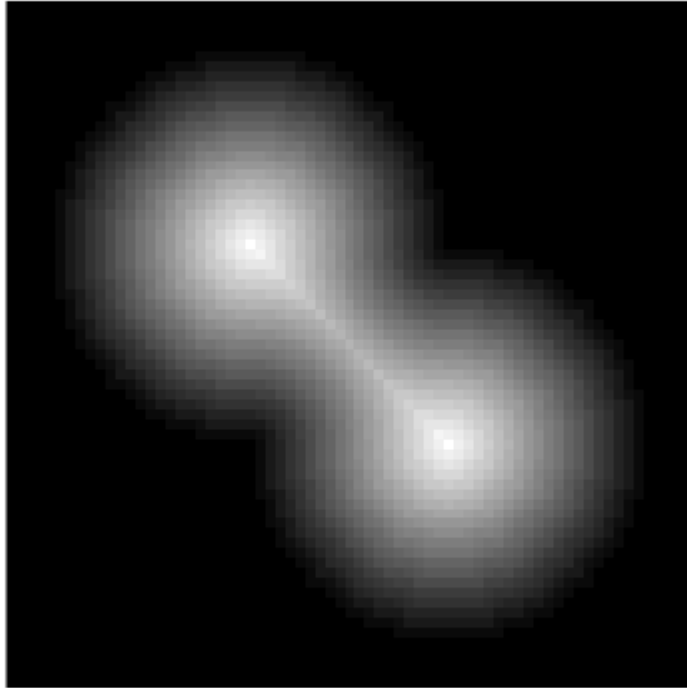
```
center1 = -10;
center2 = -center1;
dist = sqrt(2*(2*center1)^2);
radius = dist/2 * 1.4;
lims = [floor(center1-1.2*radius) ceil(center2+1.2*radius)];
[x,y] = meshgrid(lims(1):lims(2));
bw1 = sqrt((x-center1).^2 + (y-center1).^2) <= radius;
bw2 = sqrt((x-center2).^2 + (y-center2).^2) <= radius;
bw = bw1 | bw2;
figure
imshow(bw, 'InitialMagnification','fit'),
title('Dos objetos circulares superpuestos')
```



Se calcula la transformación de distancia de la imagen binaria complementada.

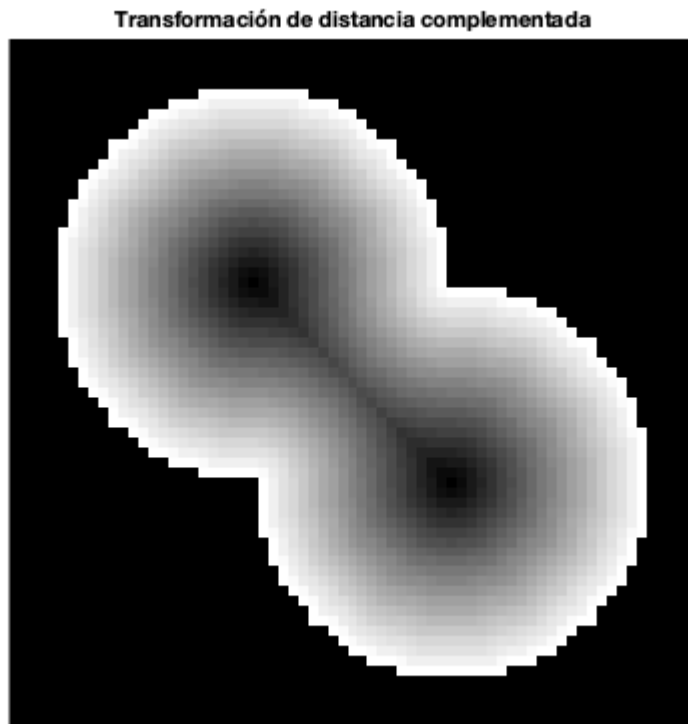
```
nbw = ~bw;
D = bwdist(nbw);
figure
imshow(D, [], 'InitialMagnification','fit')
title('Transformación de distancia de la imagen binaria complementada')
```

Transformación de distancia de la imagen binaria complementada



Se complementa la transformación de distancia y se fuerzan los píxeles que no pertenezcan a los objetos a valor **Inf**

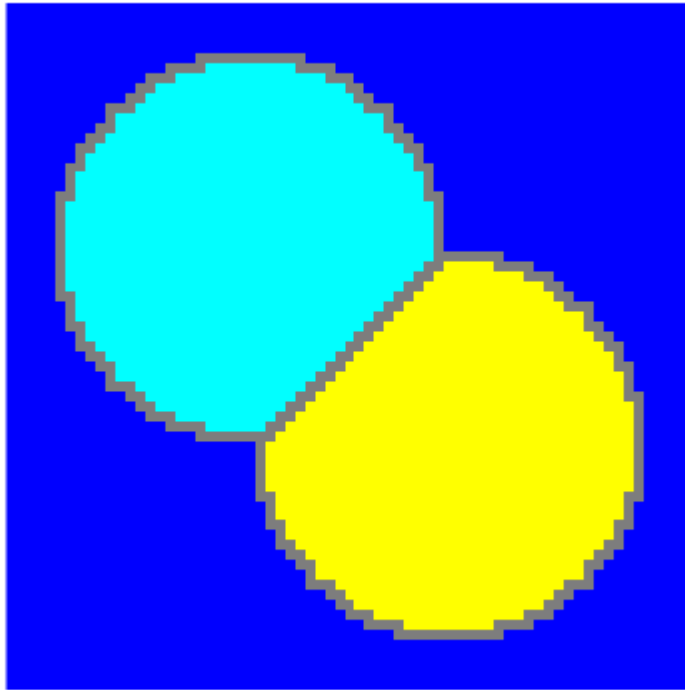
```
D = -D;  
D(nbw) = -Inf;  
imshow(D, [], 'InitialMagnification', 'fit')  
title('Transformación de distancia complementada');
```



Se calcula la transformación de Watershed y se visualiza la matriz de etiquetas resultante como una imagen RGB

```
L = watershed(D);  
rgb = label2rgb(L, 'jet', [.5 .5 .5]);  
figure  
imshow(rgb, 'InitialMagnification', 'fit')  
title(sprintf('Transformada Watershed. Objetos segmentados: %d', max(max(L, [], 1), [], 2)))
```

Transformada Watershed. Objetos segmentados: 3



Se pide:

1. ¿qué ocurriría si en lugar de utilizar la imagen binaria complementada ($\sim \mathbf{bw}$) se utilizase (\mathbf{bw})? realice el cambio y comente los resultados.
2. Cargue la imagen '`rice.png`' umbralízela y realice una segmentación utilizando la transformación Watershed, tal y como se ha hecho en el ejemplo dado. ¿Existe sobre-segmentación, es decir, se han segmentados muchos más objetos de los que hay en la imagen?
3. Si se ha producido sobre-segmentación realice un rellenado de los objetos segmentados en el apartado anterior con valor 0 (`BWrellenada=bwfill(L==0, 'holes')`) y vuelva a hacer el etiquetado a partir de la imagen `BWrellenada`. Finalmente compruebe si el número de objetos segmentados es menor.