

Procesamiento de imágenes



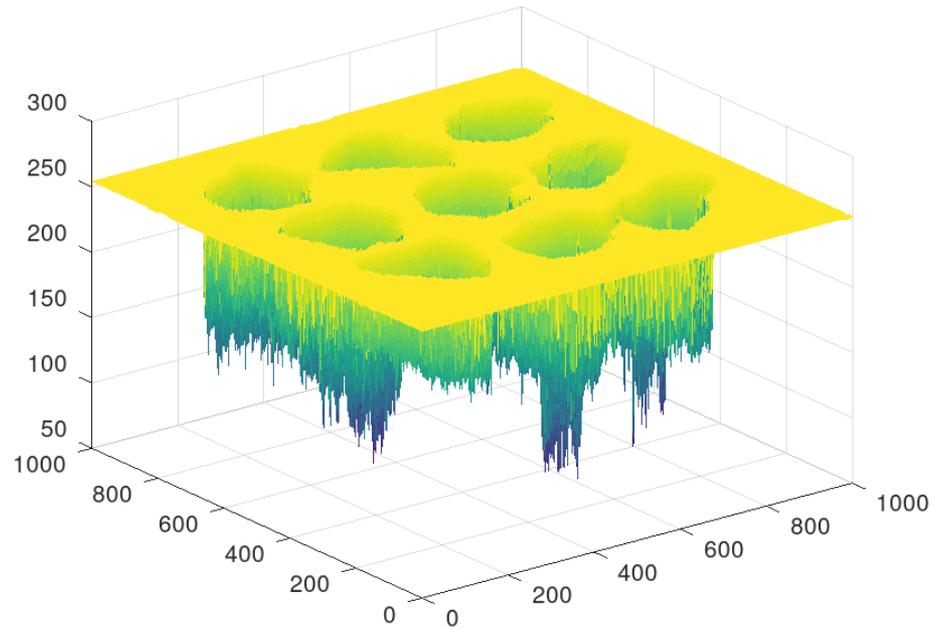
Algoritmo Watershed

Idea básica

- Una imagen en escala de grises se puede ver como una superficie topográfica.
- Zonas claras indican picos o montañas.
- Zonas oscuras indican valles.
- Se llenan los valles con agua de distintos colores (etiquetas).
- Cuando el agua de distintas zonas comienza a tocarse se construyen barreras.



Imagen de piedras



Superficie topológica

-
- Existen varios métodos para calcular la segmentación por watershed...
 - Vamos a utilizar el método basado en la transformada de distancia.
 - La transformada de distancia en una imagen binaria computa la distancia de cada pixel hasta el pixel más cercano no nulo.

Transformada de distancia

- En Octave:
- `D=bwdist(b, method)`
- Donde `b` es una imagen binaria y `method` es la manera de calcular la distancia. Por default es la distancia euclídea. Se necesitan los objetos en negro.

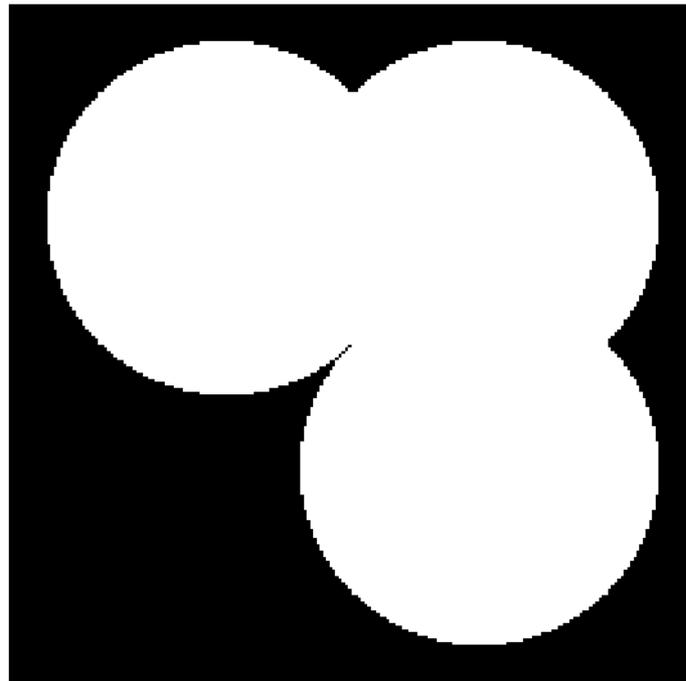
Transformada watershed

- En Octave:
- `L=watershed(~D)`
- Calcula la transformada watershed. Devuelve una matriz etiquetada. Necesita el complemento de la transformada de distancia. Las fronteras tienen valor cero.

Ejemplo: objetos que se tocan

`imshow (bw)`

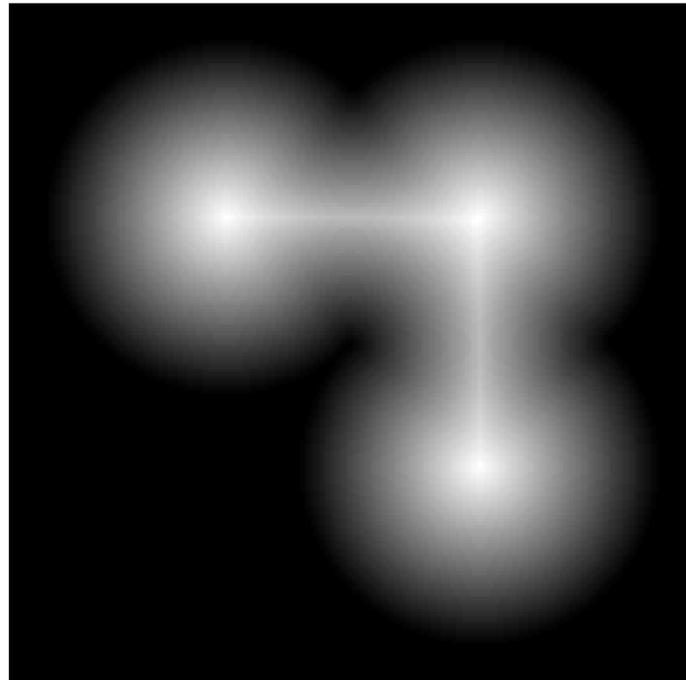
imagen binaria con 3 objetos



Ejemplo: objetos que se tocan

```
D = bwdist(~bw);  
figure, imshow(D, [])
```

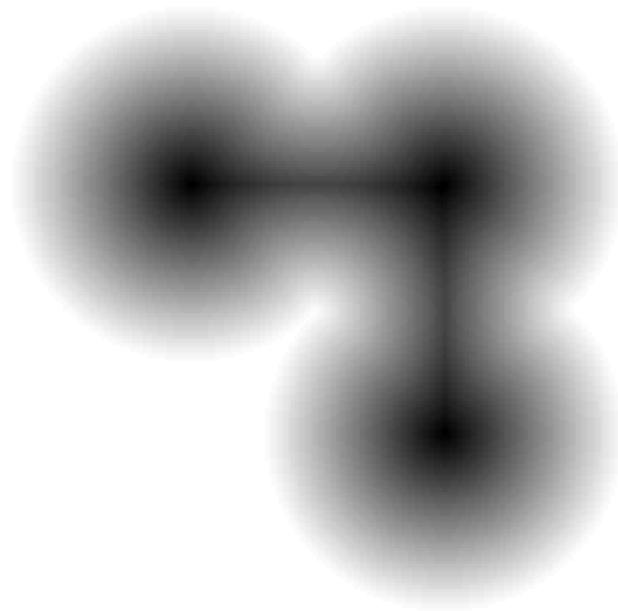
transformada de distancia del complemento



Ejemplo: objetos que se tocan

```
D = -D;  
figure, imshow(D, [])
```

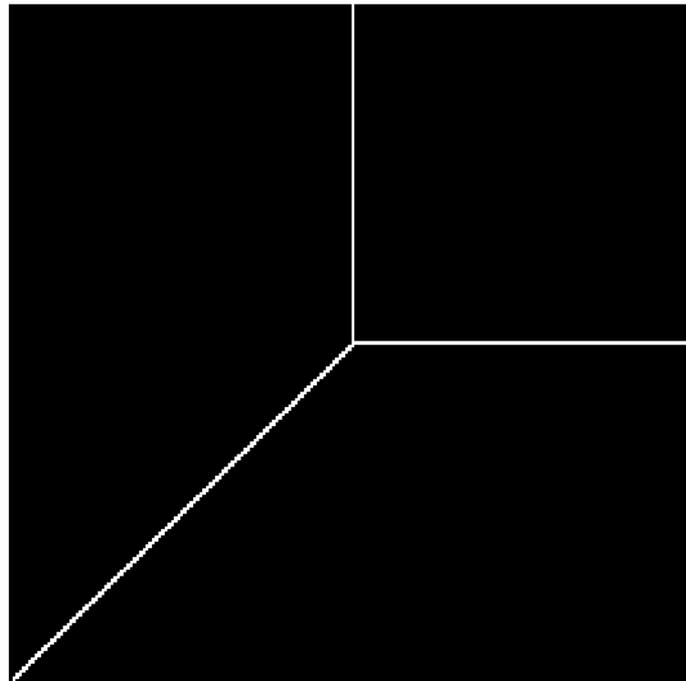
complemento de la transformada de distancia



Ejemplo: objetos que se tocan

```
L = watershed(D);  
L = L==0;  
figure, imshow(L)
```

matriz de fronteras



Ejemplo: objetos que se tocan

```
L(~bw) = 0;  
rgb = label2rgb(L, 'jet', [.5 .5 .5]);  
figure, imshow(rgb)
```

segmentación por watershed

