

La traducción de esta página aún no se ha actualizado a la versión más reciente. Haga clic aquí para ver la última versión en inglés.

Segmentación watershed controlada por marcadores

Este ejemplo muestra cómo utilizar la segmentación watershed para separar objetos adyacentes en una imagen. La transformada de watershed encuentra las "cuencas de captación" o las "líneas de cresta de las cuencas" en una imagen tratándola como una superficie en la que los píxeles claros son elevados y los oscuros son bajos.

Probar este ejemplo

 Copy Command

La segmentación con transformada de watershed funciona mejor si puede identificar, o "marcar", los objetos en primer plano y las ubicaciones de fondo. La segmentación watershed controlada por marcadores sigue este procedimiento básico:

1. Calcula una función de segmentación. En esta imagen, las regiones oscuras son los objetos que desea segmentar.
2. Calcula los marcadores de primer plano. Cada objeto contiene blobs de píxeles conectados.
3. Calcula los marcadores de fondo. Estos píxeles no son parte de ningún objeto.
4. Modifica la función de segmentación, de modo que solo presente mínimos en las ubicaciones de marcadores de primer plano y de fondo.
5. Calcula la transformada de watershed de la función de segmentación modificada.

Paso 1: Leer una imagen en color y convertirla a escala de grises

```
rgb = imread("pears.png");  
I = im2gray(rgb);  
imshow(I)  
  
text(732,501,"Image courtesy of Corel(R)",...  
     "FontSize",7,"HorizontalAlignment","right")
```

 Get ▼



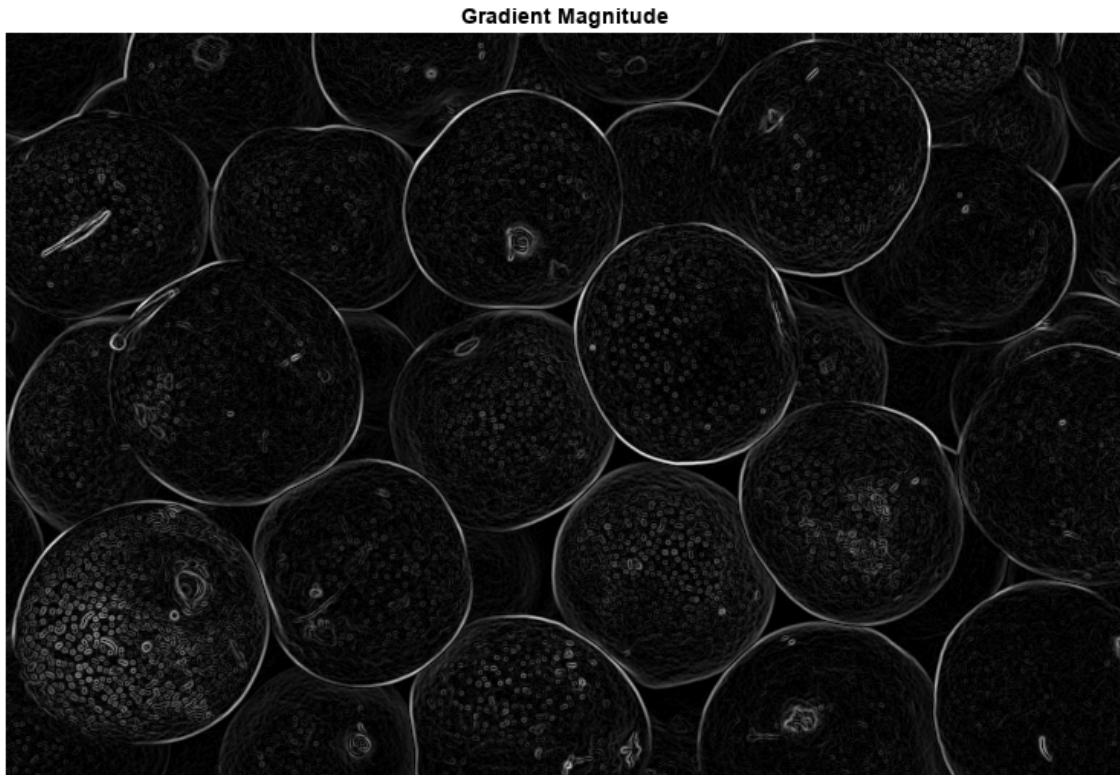
Image courtesy of Corel(R)

Paso 2: Utilizar la magnitud del gradiente como función de segmentación

Calcule la magnitud del gradiente. El gradiente es alto en el borde de los objetos y bajo (en su mayoría) en el interior.

```
gmag = imgradient(I);  
imshow(gmag,[])  
title("Gradient Magnitude")
```

Get ▼

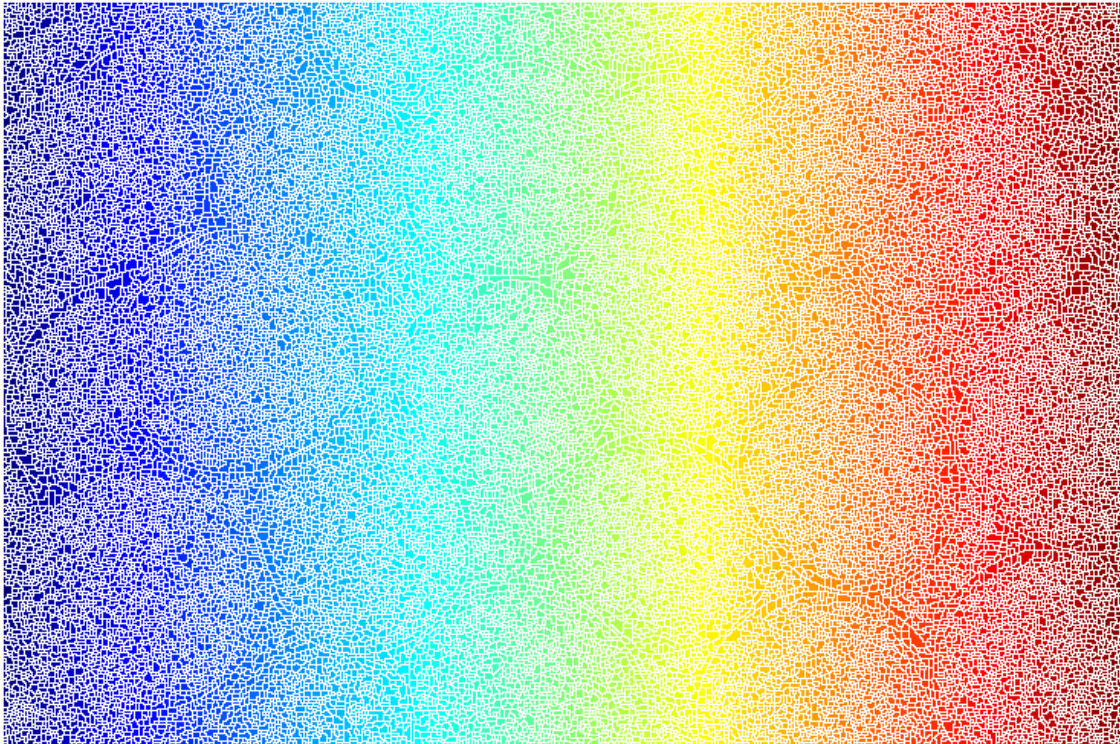


¿Puede segmentar la imagen utilizando la transformada de watershed directamente en la magnitud del gradiente?

```
L = watershed(gmag);  
Lrgb = label2rgb(L);  
imshow(Lrgb)  
title("Watershed Transform of Gradient Magnitude")
```

Get ▼

Watershed Transform of Gradient Magnitude



No. Utilizar la transformada de watershed directamente sin un preprocesamiento adicional como los cálculos de marcadores que aparecen a continuación resulta en una "sobresgmentación".

Paso 3: Marcar los objetos en primer plano

Aquí se puede aplicar una variedad de procedimientos para encontrar los marcadores de primer plano, que deben ser blobs de píxeles ubicados dentro de cada uno de los objetos en primer plano. En este ejemplo, se utilizan técnicas morfológicas llamadas "apertura por reconstrucción" y "cierre por reconstrucción" para "limpiar" la imagen. Estos procedimientos crearán máximos planos en cada objeto que se pueden ubicar utilizando `imregionalmax`.

La apertura es una erosión seguida de una dilatación, y la apertura por reconstrucción es una erosión seguida de una reconstrucción morfológica. Comparémoslas. En primer lugar, calcule la apertura con `imopen`.

```
se = strel("disk",20);  
Io = imopen(I,se);  
imshow(Io)  
title("Opening")
```

 Get ▼

Opening

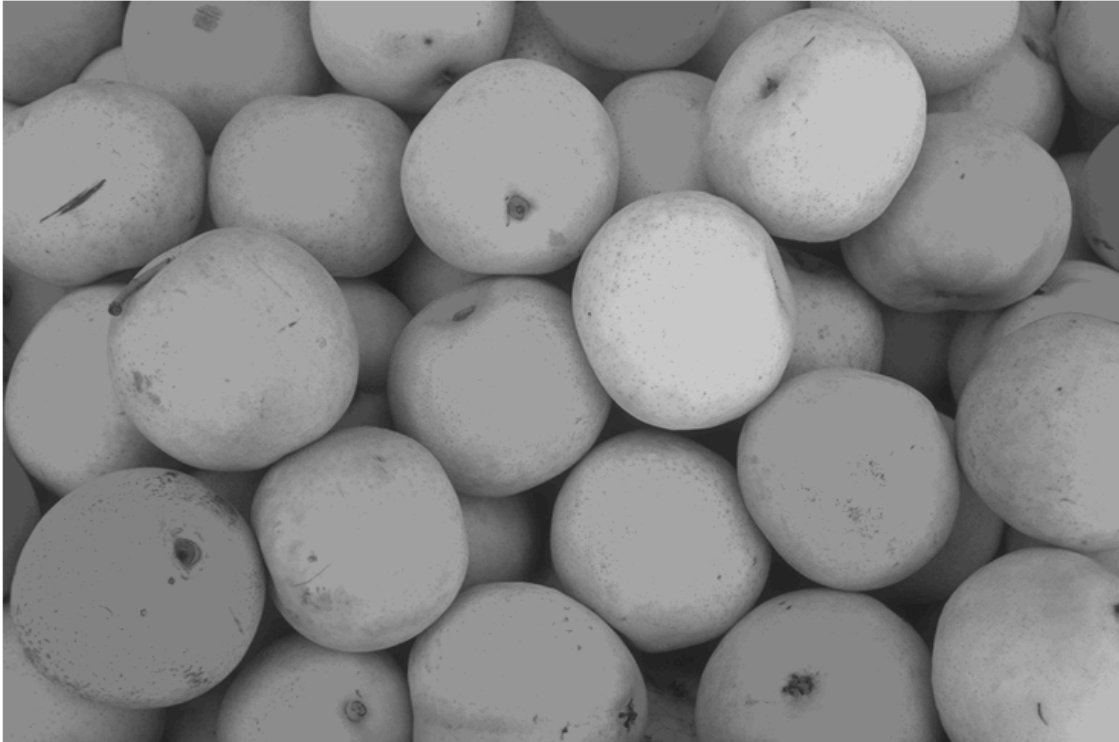


A continuación, calcule la apertura por reconstrucción con `imerode` e `imreconstruct`.

```
Ie = imerode(I,se);  
Iobr = imreconstruct(Ie,I);  
imshow(Iobr)  
title("Opening-by-Reconstruction")
```

 Get ▼

Opening-by-Reconstruction

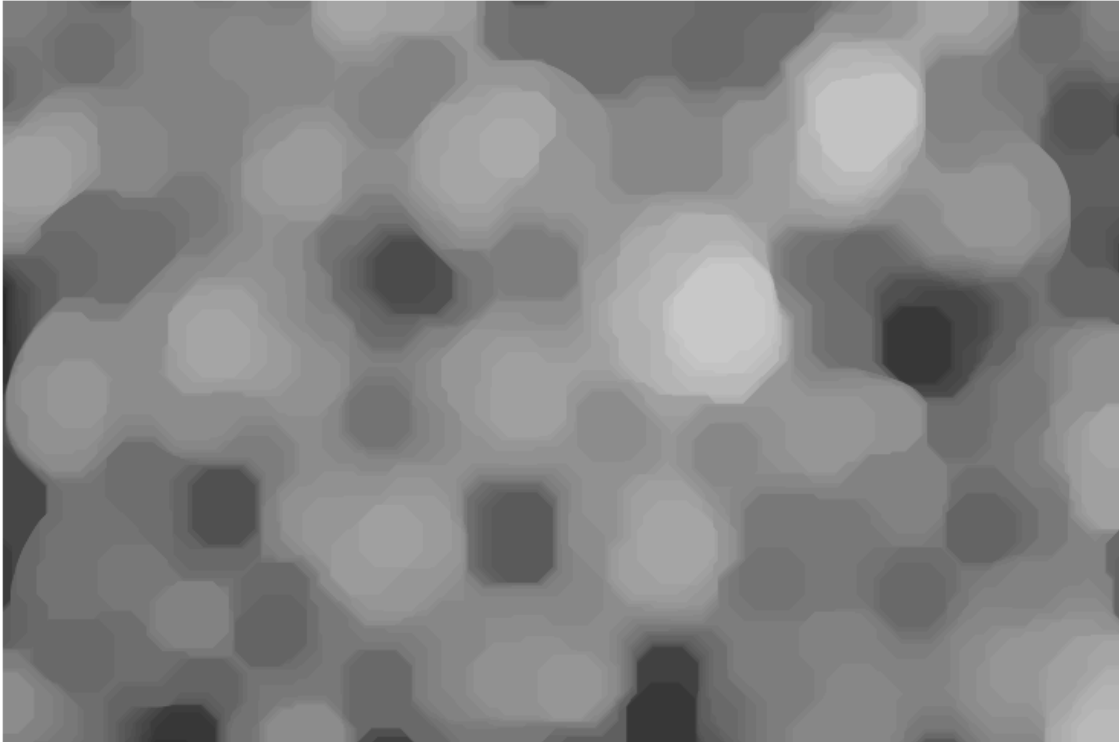


Realizar un cierre después de una apertura puede eliminar los puntos oscuros y las marcas de los tallos. Compare un cierre morfológico normal con un cierre por reconstrucción. En primer lugar, pruebe con `imclose`:

```
Ioc = imclose(Io,se);  
imshow(Ioc)  
title("Opening-Closing")
```

 Get ▼

Opening-Closing

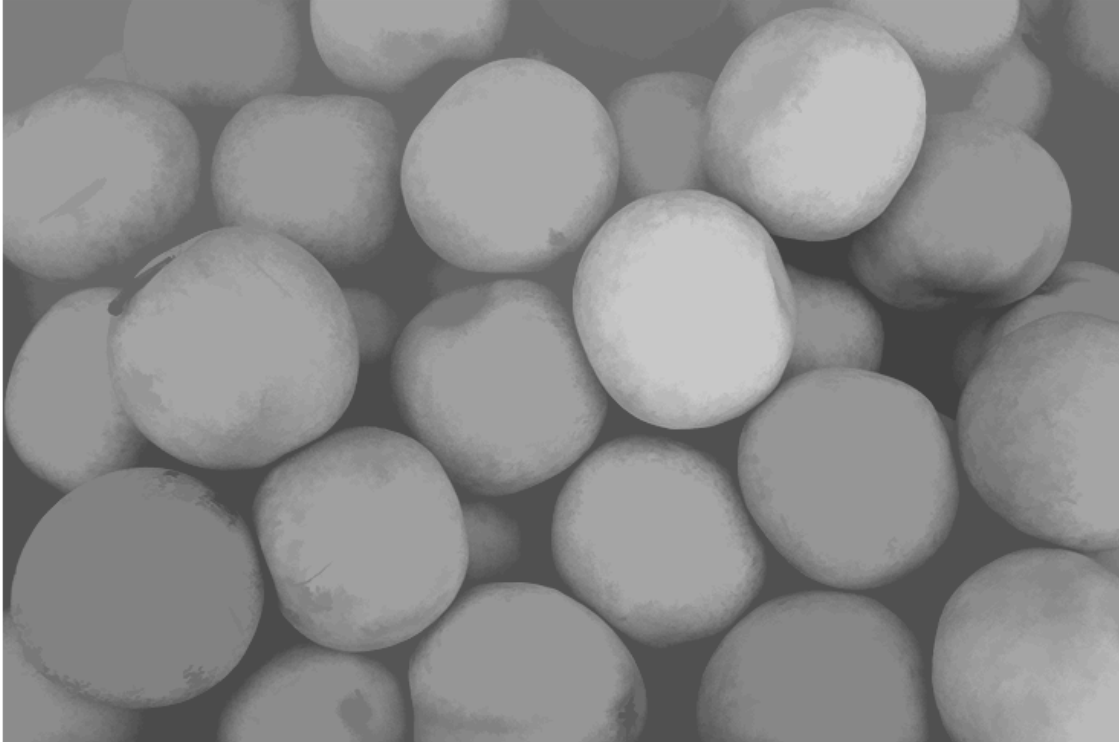


Ahora, utilice `imdilate` y, a continuación, `imreconstruct`. Deberá complementar las entradas de la imagen y la salida de `imreconstruct`.

```
Iobrd = imdilate(Iobr,se);  
Iobrcbr = imreconstruct(imcomplement(Iobrd),imcomplement(Iobr));  
Iobrcbr = imcomplement(Iobrcbr);  
imshow(Iobrcbr)  
title("Opening-Closing by Reconstruction")
```

 Get ▼

Opening-Closing by Reconstruction

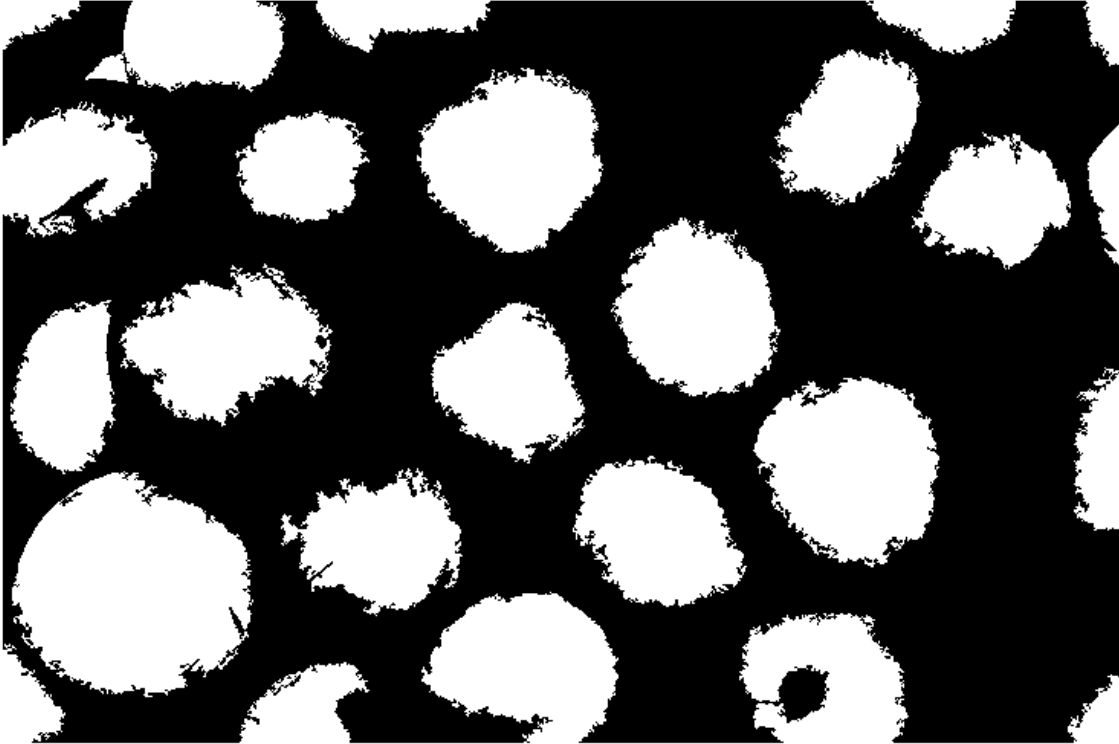


Como puede ver comparando `Iobrcbr` con `Ioc`, la apertura y el cierre por reconstrucción son más efectivos que la apertura y el cierre estándar en la eliminación de pequeñas imperfecciones sin afectar a las formas generales de los objetos. Calcule los máximos regionales de `Iobrcbr` para obtener unos buenos marcadores de primer plano.

```
fgm = imregionalmax(Iobrcbr);  
imshow(fgm)  
title("Regional Maxima of Opening-Closing by Reconstruction")
```

 Get ▼

Regional Maxima of Opening-Closing by Reconstruction

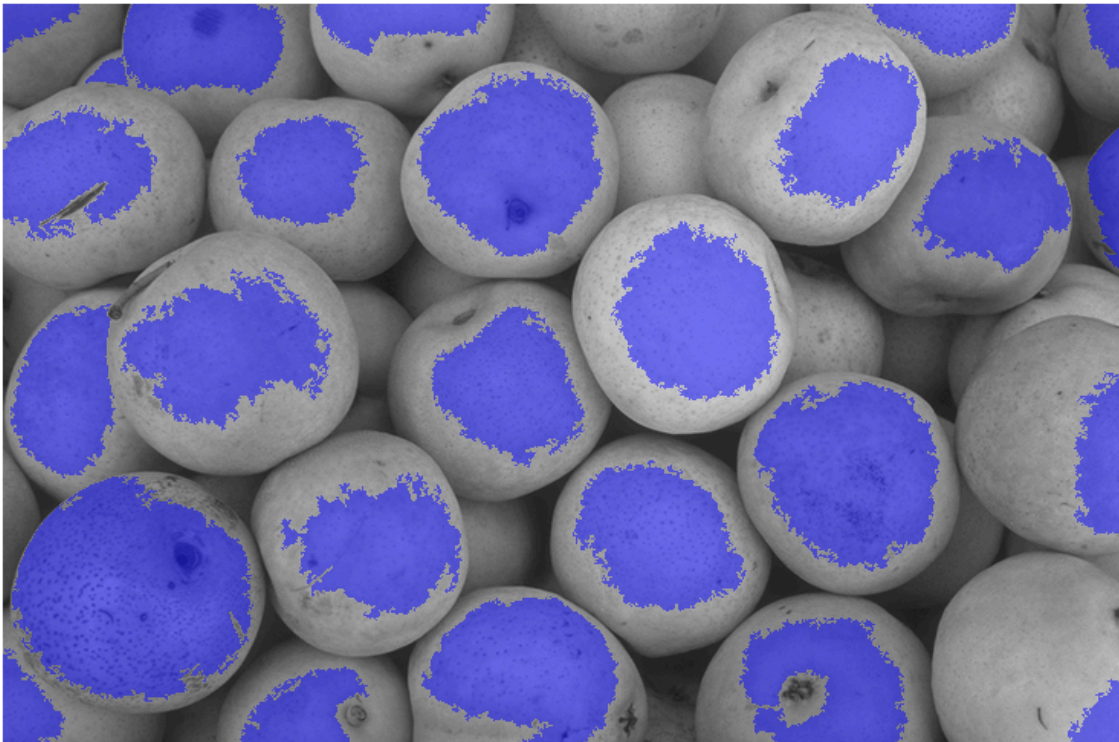


Para interpretar el resultado, superponga la imagen del marcador de primer plano a la imagen original.

```
I2 = labeloverlay(I,fgm);  
imshow(I2)  
title("Regional Maxima Superimposed on Original Image")
```

 Get ▼

Regional Maxima Superimposed on Original Image



Observe que algunos de los objetos más oscuros u ocultos no se marcarán y, por lo tanto, no se segmentarán correctamente en el resultado final. Además, los marcadores de fondo en algunos objetos se sitúan en el borde del objeto. Eso significa que debe limpiar los bordes de los blobs de marcadores y reducirlos un poco. Puede hacerlo con un cierre seguido de una erosión.

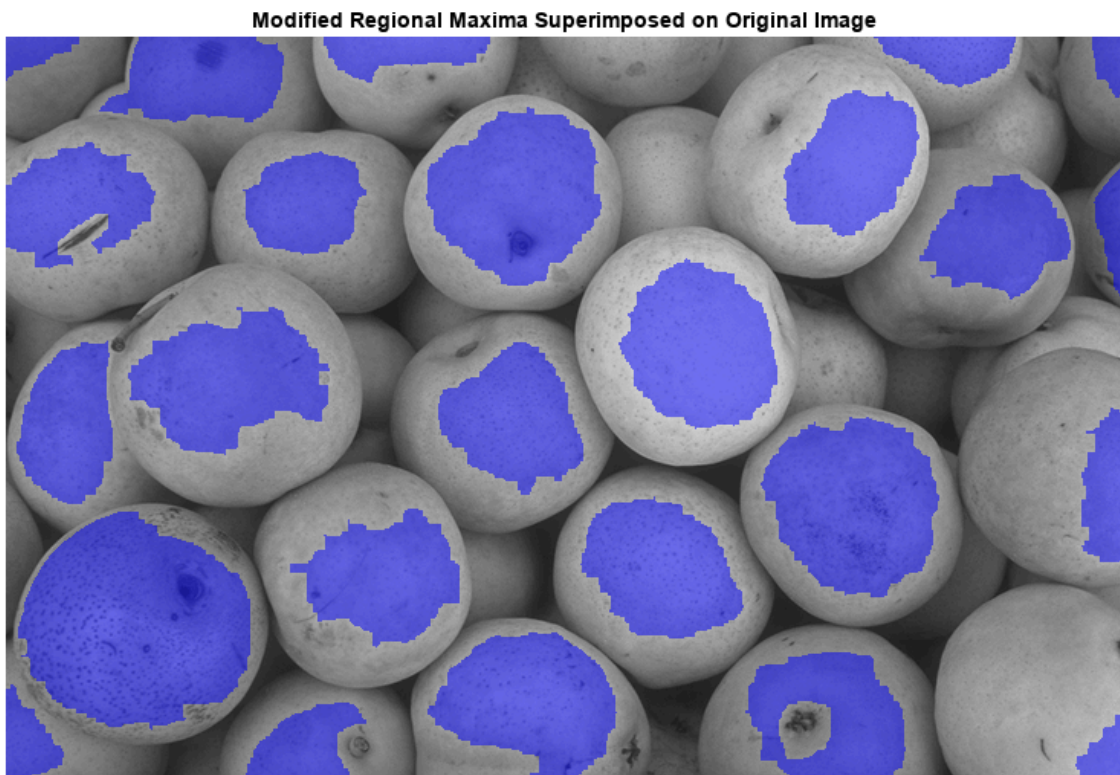
```
se2 = strel(ones(5,5));  
fgm2 = imclose(fgm,se2);  
fgm3 = imerode(fgm2,se2);
```

 Get ▼

Este procedimiento suele dejar píxeles aislados que deberán eliminarse. Puede hacerlo con `bwareaopen`, que elimina todos los blobs que contienen un número de píxeles menor que el especificado.

```
fgm4 = bwareaopen(fgm3,20);  
I3 = labeloverlay(I,fgm4);  
imshow(I3)  
title("Modified Regional Maxima Superimposed on Original Image")
```

 Get ▼



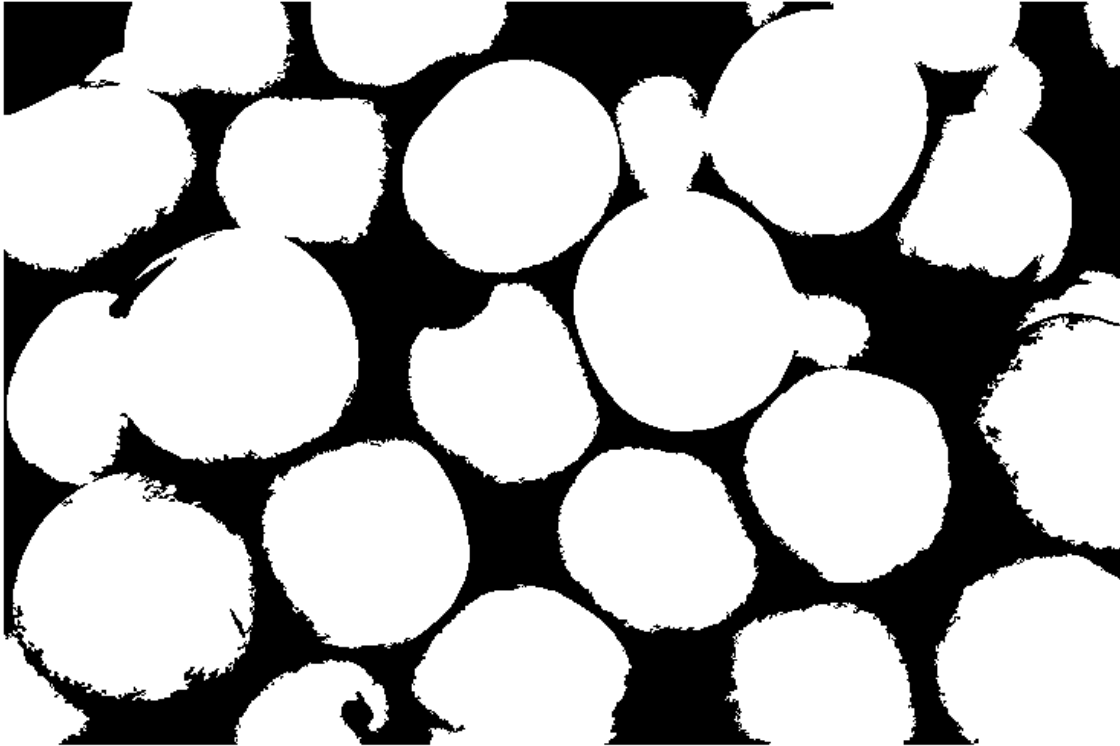
Paso 4: Calcular los marcadores de fondo

Ahora, es necesario que marque el fondo. En la imagen limpia, `Iobrcbr`, los píxeles oscuros pertenecen al fondo, de manera que puede empezar por una operación de detección de umbral.

```
bw = imbinarize(Iobrcbr);  
imshow(bw)  
title("Thresholded Opening-Closing by Reconstruction")
```

 Get ▼

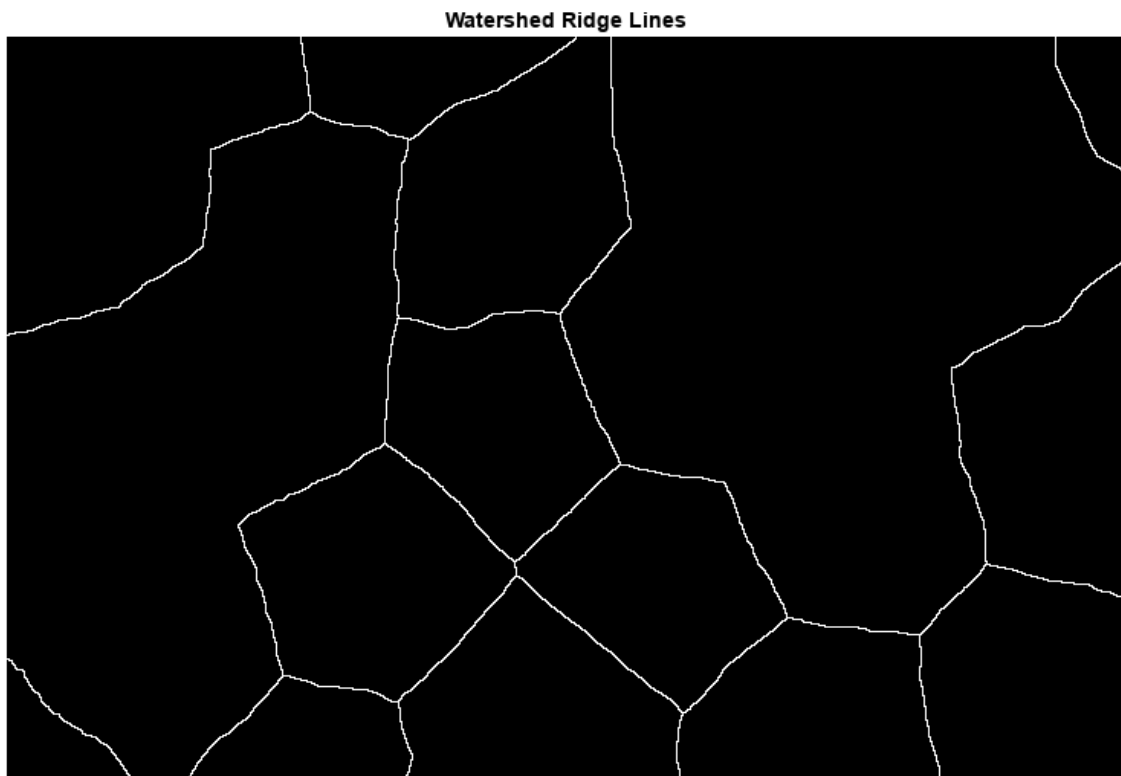
Thresholded Opening-Closing by Reconstruction



Los píxeles de fondo aparecen en negro, pero no conviene que los marcadores de fondo estén demasiado cerca de los bordes de los objetos que queremos segmentar. Estrecharemos el fondo calculando el "esqueleto por zonas de influencia" (o SKIZ) del fondo de bw. Puede hacerlo calculando la transformada de watershed de la transformada de distancia de bw y, a continuación, obteniendo las cuencas de captación ($DL == 0$) del resultado.

```
D = bwdist(bw);  
DL = watershed(D);  
bgm = DL == 0;  
imshow(bgm)  
title("Watershed Ridge Lines")
```

 Get ▼



Paso 5: Calcular la transformada de watershed de la función de segmentación

La función `imimposemin` se puede utilizar para modificar una imagen de modo que solo presente mínimos regionales en determinadas ubicaciones deseadas. Aquí puede utilizar `imimposemin` para modificar la magnitud del gradiente de modo que solo presente mínimos regionales en los píxeles de los marcadores de primer plano y de fondo.

```
gmag2 = imimposemin(gmag, bgm | fgm4);
```

 Get ▼

Por último, calcule la segmentación basada en watershed.

```
L = watershed(gmag2);
```

 Get ▼

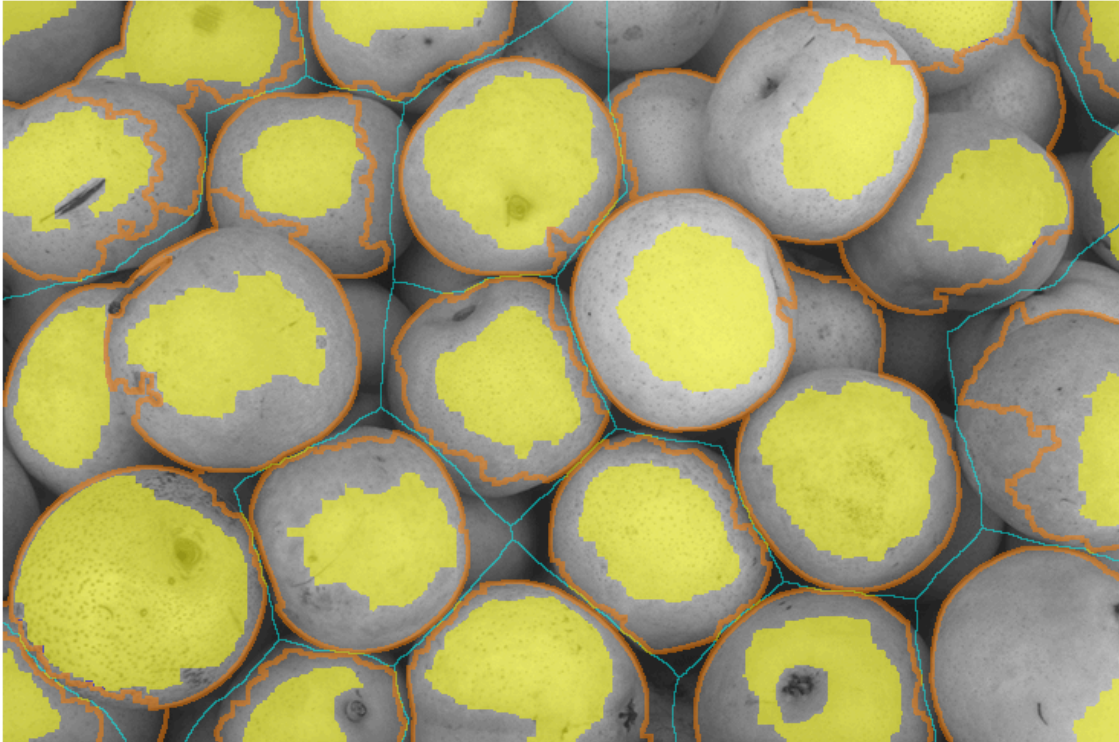
Paso 6: Visualizar el resultado

Una técnica de visualización es superponer marcadores de primer plano, marcadores de fondo y límites de objetos segmentados en la imagen original. Puede utilizar la dilatación según sea necesario para hacer que determinados aspectos, como los límites de los objetos, sean más visibles. Los límites de los objetos se encuentran donde $L == 0$. Los marcadores de primer plano y de fondo binarios se escalan a diferentes valores enteros para que se les asignen distintas etiquetas.

```
labels = imdilate(L==0,ones(3,3)) + 2*bgm + 3*fgm4;  
I4 = labeloverlay(I,labels);  
imshow(I4)  
title("Markers and Object Boundaries Superimposed on Original Image")
```

 Get ▼

Markers and Object Boundaries Superimposed on Original Image

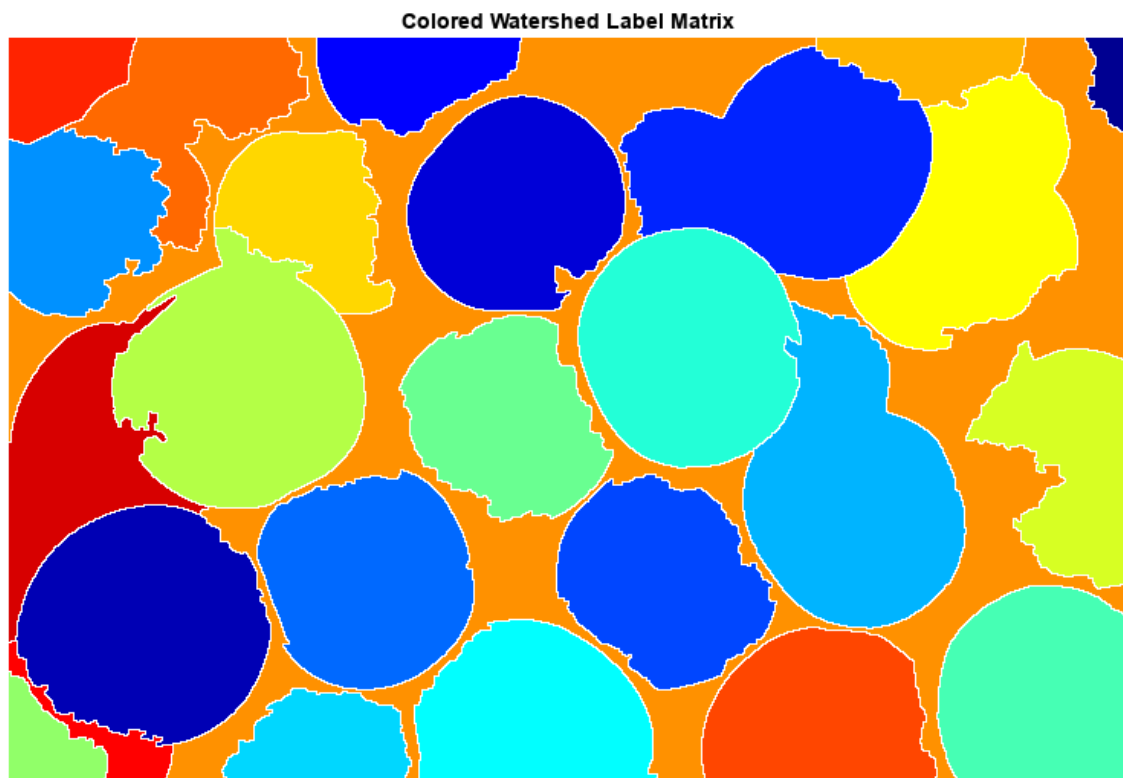


Esta visualización ilustra cómo las ubicaciones de los marcadores de primer plano y de fondo afectan al resultado. En un par de ubicaciones, algunos objetos parcialmente ocultos y oscuros se han mezclado con los objetos adyacentes más claros porque no disponían de marcadores de primer plano.

Otra técnica de visualización útil es mostrar la matriz de etiquetas como una imagen en color. Las matrices de etiquetas, como las producidas por `watershed` y `bwlabel`, se pueden convertir a imágenes en color verdadero con fines de visualización utilizando `label2rgb`.

```
Lrgb = label2rgb(L,"jet","w","shuffle");  
imshow(Lrgb)  
title("Colored Watershed Label Matrix")
```

 Get ▼

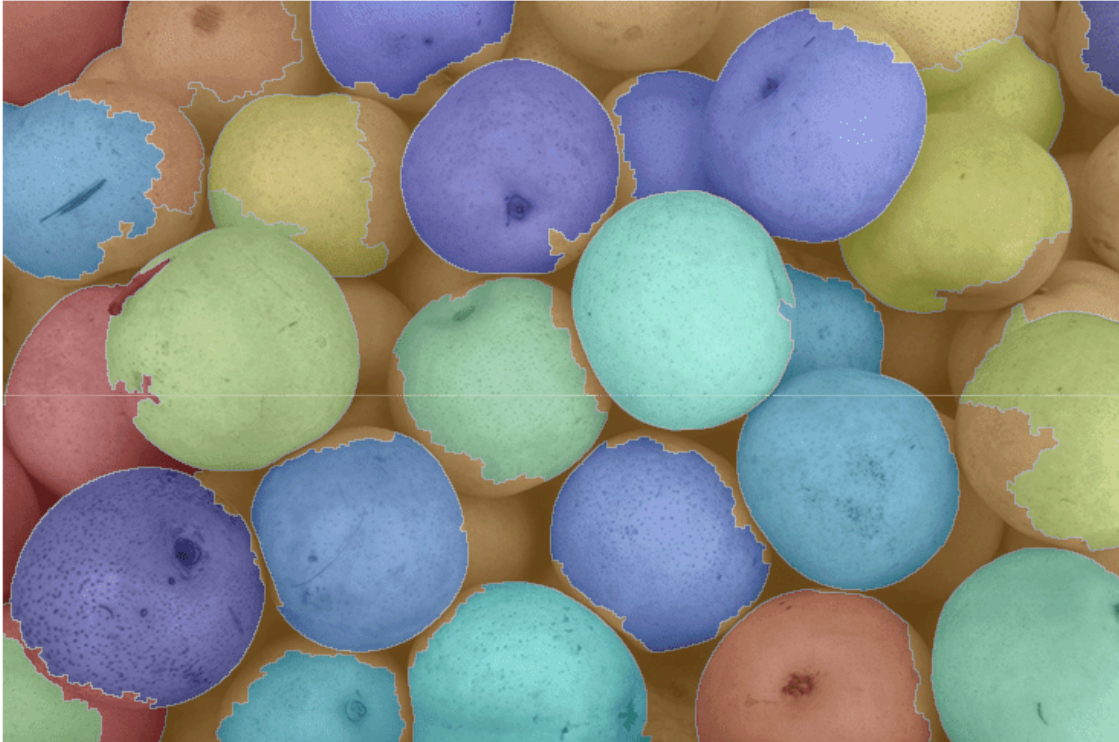


Puede utilizar la transparencia para superponer esta matriz de etiquetas en pseudocolor a la imagen de intensidad original.

```
figure
imshow(I)
hold on
himage = imshow(Lrgb);
himage.AlphaData = 0.3;
title("Colored Labels Superimposed Transparently on Original Image")
```

 Get ▼

Colored Labels Superimposed Transparently on Original Image



Consulte también

[watershed](#) | [imopen](#) | [imreconstruct](#) | [imclose](#) | [imdilate](#) | [imregionalmax](#) | [imerode](#) | [bwareaopen](#) | [bwdist](#) | [label2rgb](#) | [imcomplement](#) | [labeloverlay](#) | [imgradient](#)