

Desafío 5

**Extracción de contornos de la imagen de
piedras y de la imagen hyperspectral**

Iyán Álvarez



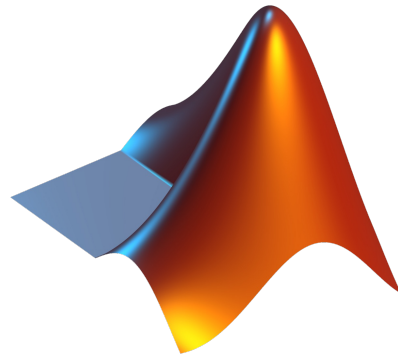
Trabajo realizado




Dos scripts y una función auxiliar en Matlab.

El script “ExtraccionContornosPiedras_main” carga los datos, realiza un proceso de binarización y filtrado de componentes conexos por área y calcula el error.

El script “ExtraccionContornosHyperspectral_main” carga los datos, realiza un proceso de suavizado mediante un filtro gaussiano, ejecuta la detección de bordes mediante proceso de sobel, filtrado de componentes conexos por área y calcula el error.

La función auxiliar “error_imagenes_binarias.m” calcula el error entre la imagen binaria dada en el material y la obtenida mediante el script.



 error_imagenes_binarias.m
 ExtraccionContornosHyperspectral_main.m
 ExtraccionContornosPiedras_main.m

Función “error_imágenes_binarias”

Compara dos imágenes binarias y calcula el error.

1. Calcula el número de elementos diferentes entre las dos imágenes.
2. Calcula la tasa de error

La tasa de error obtenida se devuelve como salida.

0.047658473

Extracción contornos imagen piedras

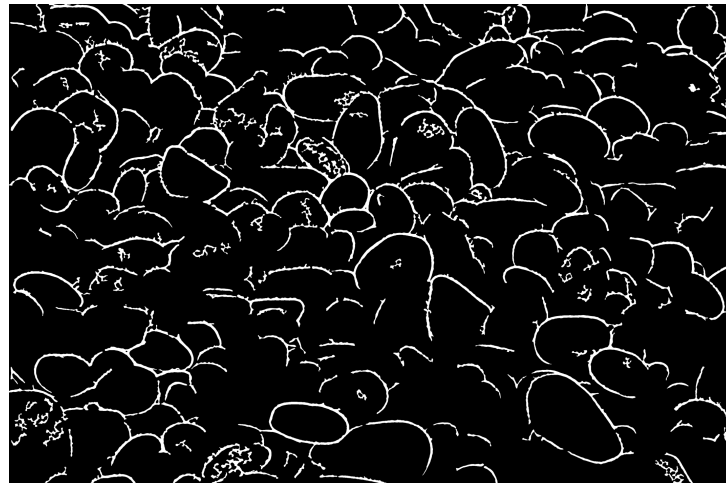
Se ha realizado un proceso de binarización y filtrado de componentes conexos por área.

```
img_binarized = imbinarize(img, umbral_binarizacion);  
  
labels = bwlabel(img_binarized);  
  
areas = regionprops(labels, 'Area');  
img_filtered = ismember(labels, find([areas.Area] > umbral_area));
```

Contornos objetivo



Contornos obtenidos



Umbral binarización: 0.2
Umbral área: 1000
Error: 8.084760e-02

Extracción contornos imagen hyperspectral

Se ha realizado un proceso de suavizado mediante un filtro gaussiano, detección de bordes mediante proceso de sobel y filtrado de componentes conexos por área.

```
h_gaussian = fspecial('gaussian', [5, 5], 2.5);

T1_smoothed = zeros(size(T1));
for i = 1:size(T1_smoothed, 3)
    T1_smoothed(:, :, i) = imfilter(T1(:, :, i), h_gaussian);
end

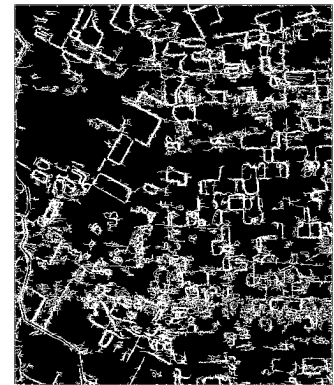
T1_edges = zeros(size(T1, 1), size(T1, 2));
for i = 1:size(T1_smoothed, 3)
    actual_edges = edge(T1_smoothed(:, :, i), 'sobel');
    T1_edges = T1_edges | actual_edges;
end

areas = regionprops(labels, 'Area');
img_filtered = ismember(labels, find([areas.Area] >
                                     umbral_area));
```

Contornos objetivo



Contornos obtenidos



Error: 1.818533e-01