

Tarea 4

Procesamiento Morfológico

INFORME

Objetivo de la parte obligatoria: Obtener la frontera del balón en rojo sobre la imagen original.

Para ello, hemos seguido los siguientes pasos:

1. Cambiamos nuestra imagen del espacio RGB al espacio HSI para ver si alguna de las componentes nos sirve como máscara de segmentación.
2. Vemos que la componente H es la mejor para nuestro objetivo que es segmentar el balón. (Anexo 1.1)

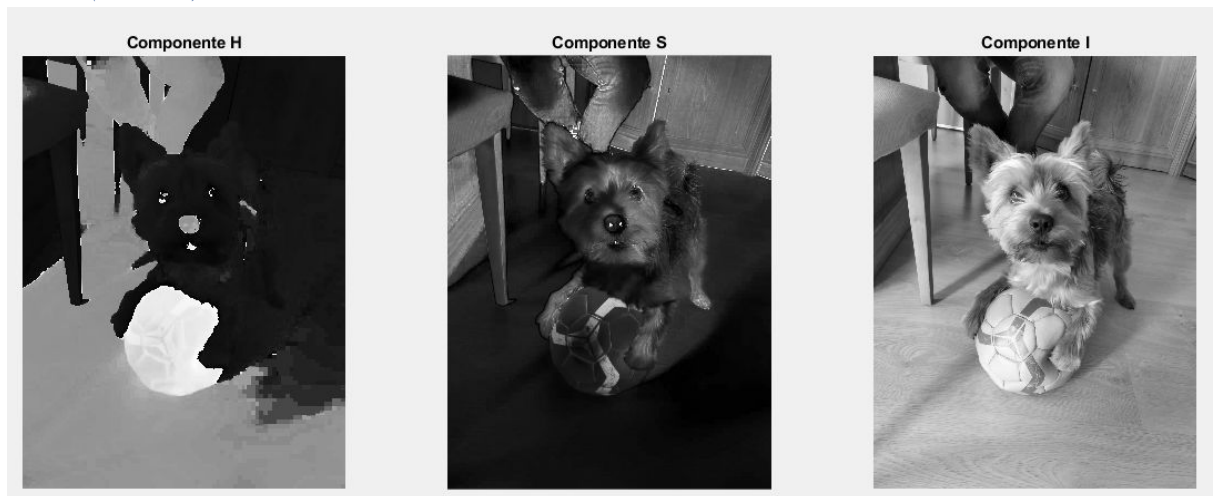


Figura 1. Componentes de la imagen en el espacio HSI

3. Como los valores más altos pertenecen al balón, hacemos un filtrado paso bajo de la componente H, para unificar los niveles similares de intensidad y facilitar la umbralización. (Anexo 1.2)



Figura 2. Componente H suavizada

4. Umbralizamos la imagen de la componente H filtrada paso bajo. Observando el histograma vemos que el umbral más adecuado para segmentar el balón es 0.8. (Anexo

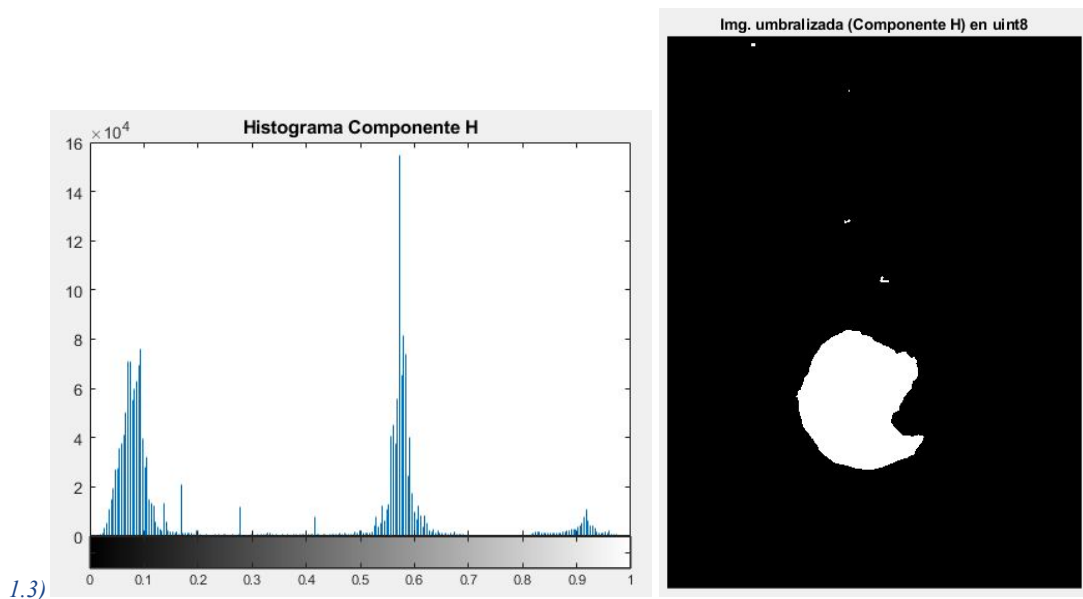


Figura 3. (a)Histograma de la componente H suavizada. (b)Resultado de la umbralización.

5. Una vez hemos obtenido la imagen umbralizada, probamos distintos elementos estructurantes para ver cual de ellos es más adecuado para la segmentación que queremos realizar. En este caso aplicamos un EE de tipo cuadrado de 35 de lado y vemos que la más adecuada para obtener el balón es la operación Apertura. *(Anexo 1.4)*

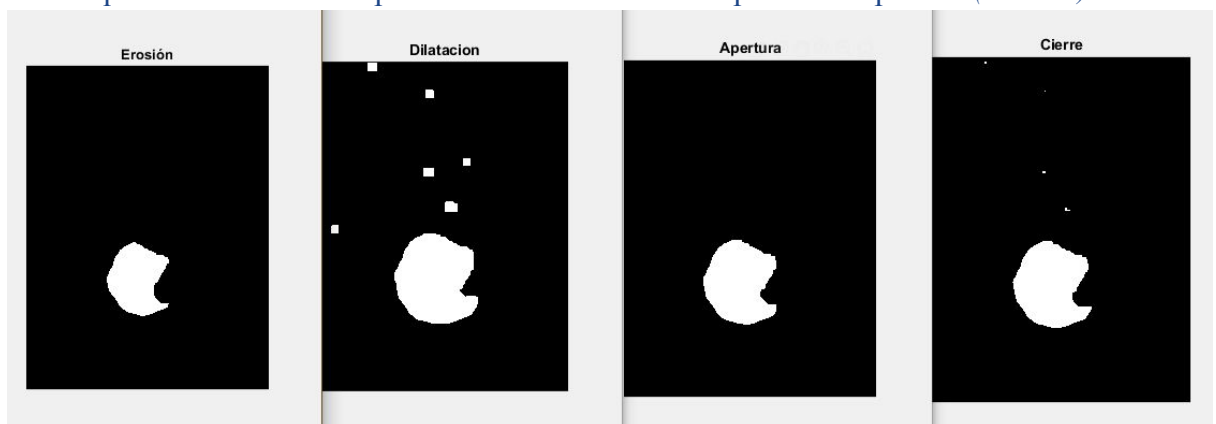


Figura 4. Aplicación de las diferentes operaciones con el EE cuadrado sobre la imagen umbralizada.
(a) Erosión. (b) Dilatación. (c) Apertura. (d) Cierre.

6. Utilizando la imagen obtenida tras aplicar el EE Apertura vamos a obtener la capa de etiquetas para la segmentación. Como está considerando como primer plano el fondo, primero calculamos el negativo de la imagen Apertura, obteniendo el balón como primer plano para a continuación crear la capa de etiquetas. *(Anexo 1.5)*

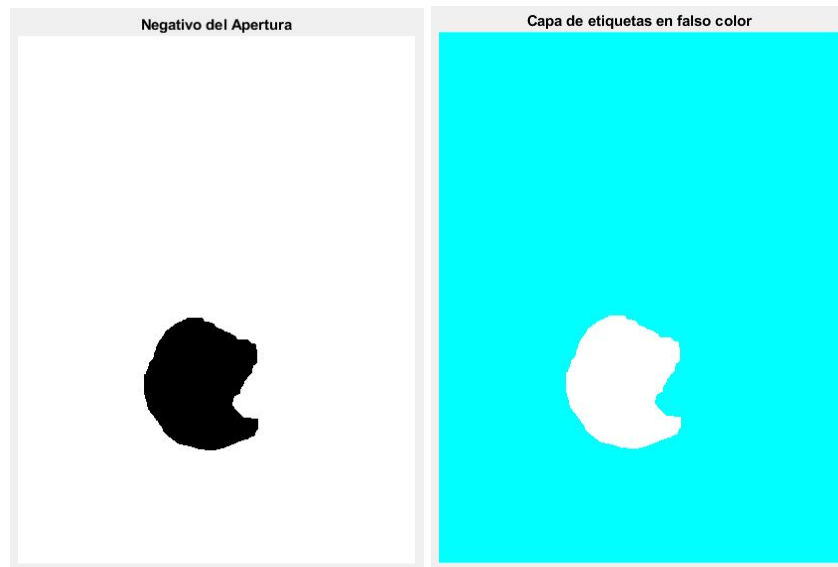


Figura 5. (a) Capa de etiquetas. (b) Capa de etiquetas en falso color.

7. Obtenemos el número de objetos y calculamos su tamaño. En este caso nos identifica un único objeto, el balón. *(Anexo 1.6)*
8. Obtenemos la frontera del balón. Para ello, utilizamos un EE disco de radio 9 (esto lo decidimos probando distintos radios y nos quedamos con el que nos pareció más adecuado). Una vez definimos aplicamos las operaciones dilatación y erosión sobre la imagen negativo. Calculando la resta de estas dos operaciones (dilatación - erosión) y obtenemos el gradiente, que es la frontera que estábamos buscando. *(Anexo 1.7)*

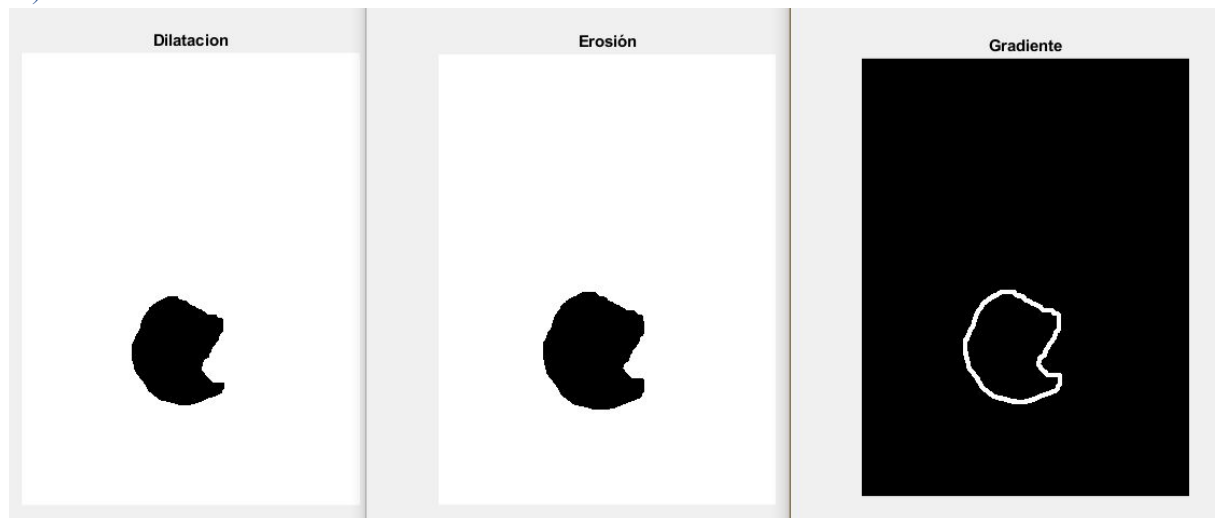


Figura 6. (a) Resultado de la dilatación. (b) Resultado de la erosión. (c) Resultado de la resta: Gradiente.

9. Por último, cambiamos el color de la frontera a rojo y la superponemos en la imagen original, obteniendo el resultado final. *(Anexo 1.8)*

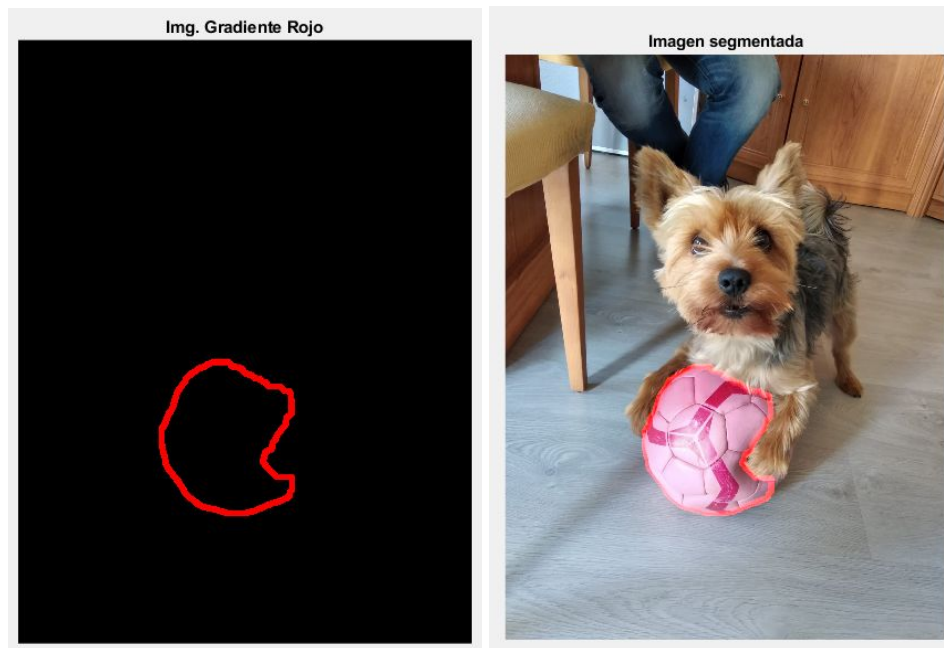


Figura 7. (a) Frontera en color rojo.(b) Resultado final.

Objetivo de la parte creativa: Segmentar los ojos, la nariz y la boca del perro mediante watershed

Nuestro objetivo es obtener las líneas de watershed sobre los 2 ojos, la boca y la nariz. Para ello, hemos seguido los siguientes pasos:

1. Primero procesamos la imagen, pasamos la imagen a escala de grises y la recortamos para obtener un mejor resultado.(Anexo 2.1)



Figura 8. Recorte de la imagen original en escala de grises.

2. Tras esto, aplicamos un filtrado secuencial, primero hacemos apertura y luego cierre, considerando un elemento estructurante de tipo disco con distintos radios hasta obtener un resultado más o menos preciso. (Anexo 2.2)



Figura 9. Resultados del filtrado secuencial.

3. Después, hacemos la segmentación por watershed. Para ello, primero hay que obtener los marcadores interiores (de las zonas de interés). Para obtenerlos seguimos los siguientes pasos (*Anexo 2.3*):
 - a. Primero, hago el negativo del filtrado para quedarme con los elementos de primer plano.



Figura 10. Negativo de la imagen filtrada

- b. Después, erosionamos la imagen resultante (negativo) con un elemento estructurante de tipo disco con radio 9.
 - c. Y con el resultado anterior, hacemos una reconstrucción morfológica para obtener los objetos de interés.

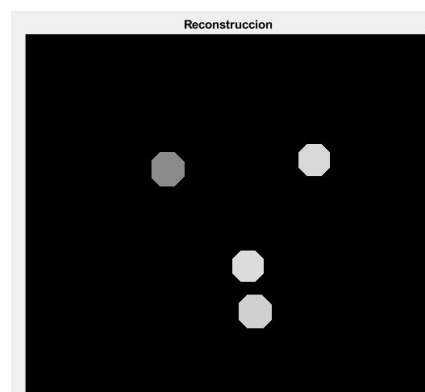


Figura 11. Imagen reconstruida con operadores morfológicos de los objetos de interés.

- d. Con la imagen reconstruida, obtenemos los máximos regionales con el comando `'imregionalmax()'`. Tras esto, nos damos cuenta de que se obtienen máximos distintos a los objetos de interés. Para eliminar los marcadores de los bordes usamos el comando `'imclearborder()'`.

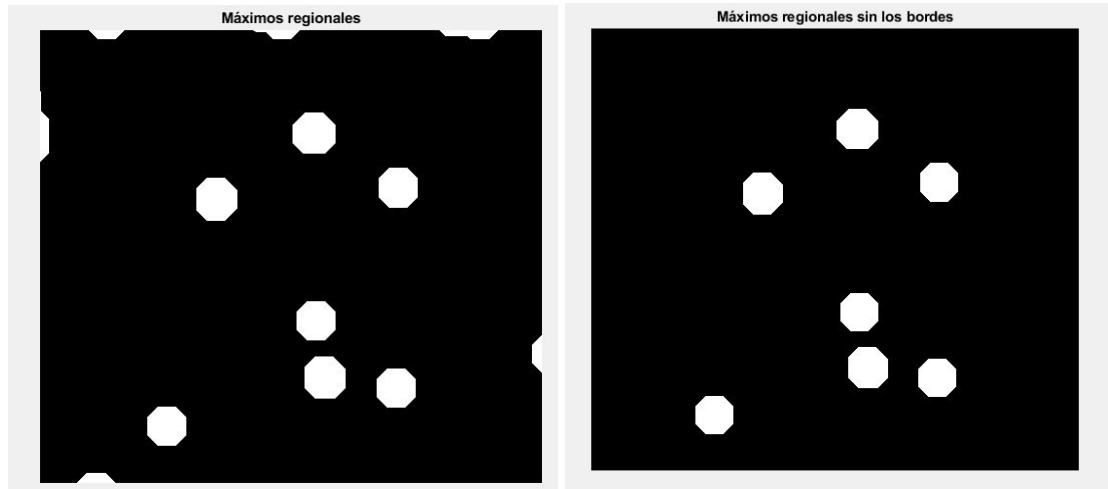


Figura 12. (a) Imagen de todos máximos regionales. (b) Imagen de los máximos regionales una vez eliminados aquellos que se encontraban en los bordes.

- e. El resultado anterior, aún contiene máximos regionales de objetos de no interés, por lo que vamos a quedarnos con los de interés haciendo uso de las propiedades, este caso del 'área'. Tras crear nuestro vector de interés, recorreremos la imagen poniendo a primer plano sólo los píxeles que estén asociados etiquetas de interés. Obteniendo así, la capa de etiquetas deseada, y con ello nuestros marcadores internos.

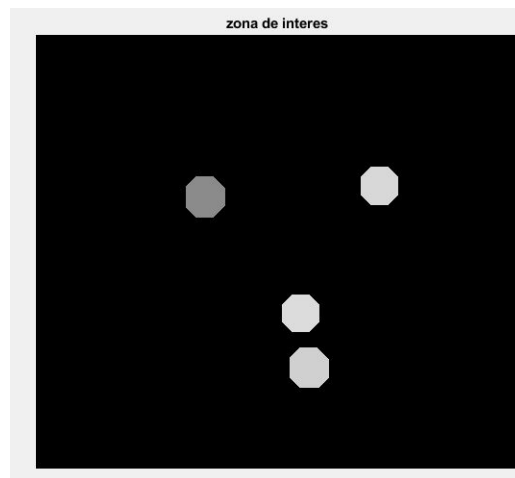


Figura 13. Zonas de interés para la segmentación de nuestro objetivo (ojos, nariz y boca del perro)

4. A continuación, calculamos el marcador externo. Para ello, utilizamos como imagen de referencia la imagen reconstruida, y le aplicamos el operador morfológico 'dilatación' con elemento estructurante de tipo disco de radio 7 para que los marcadores no correspondan a zonas de interés. (Anexo 2.4)

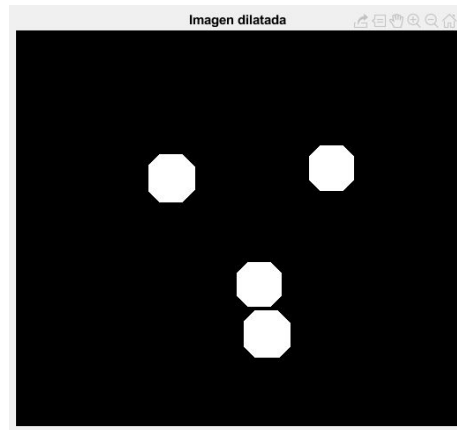


Figura 14. Resultado de la dilatación.

5. Después, lo que queremos hacer es obtener el ‘marcador externo’ usando las líneas de watershed.(*Anexo 2.4*)
 - a. Aplicamos dilatación con EE de disco y tamaño 7 (6 y 5 quedaría más juntas). Me dilata la imagen anterior.
 - b. Nos calcula la distancia euclídea de la imagen binaria, que va a ser la distancia de ese píxel a un píxel igual a 0 más cercano.
 - c. Calculamos watershed sobre la distancia anterior.
 - d. Se aplica las líneas de watershed.
 - e. Finalmente, aplico las líneas de watershed sobre la imagen a nivel de gris recortada.

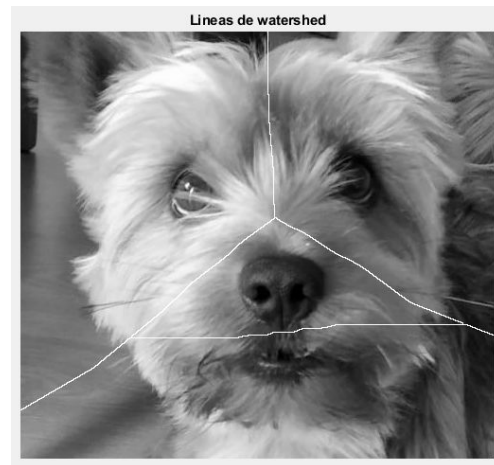


Figura 15. Líneas de watershed obtenidas tras la primera aplicación.

6. Teniendo ya los marcadores internos y el marcador externo hacemos una combinación de ambos.(*Anexo 2.5*)

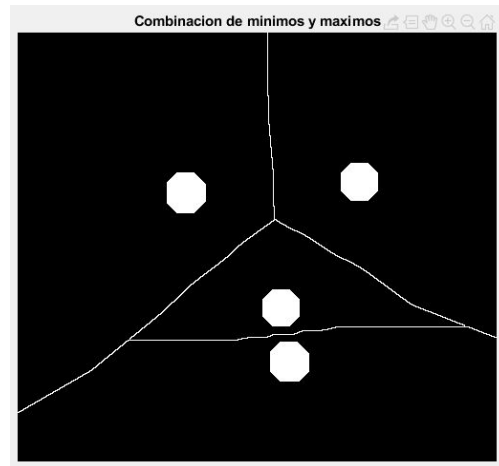


Figura 16. Combinación de los máximos regionales y las líneas de watershed.

7. Después, aplicaremos watershed sobre la imagen a nivel de gris recortada, para ello usaremos el módulo del gradiente. Para usarle, nos ayudaremos del filtro de 'sobel', para ello necesito que la imagen sea tipo double. (Anexo 2.6)



Figura 17. Resultado del módulo del gradiente.

8. Tras el resultado obtenido, nos damos cuenta de que tenemos 'sobresgmentación' debido a watershed. Para ello sobre la imagen obtenida en el gradiente, le imponemos los mínimos extraídos anteriormente (combinación de marcadores interiores y marcador exterior). Para hacer esta imposición hacemos uso del comando 'imimposemin()'. (Anexo 2.7)

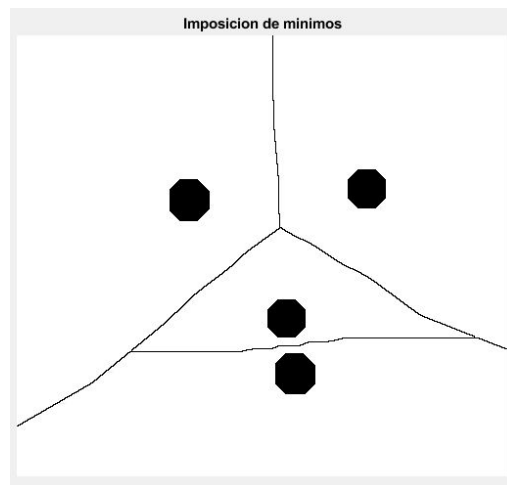


Figura 18. Resultado de la imposición de los mínimos regionales.

9. Ahora a la nueva imagen obtenida, 'L_frontera' la aplicamos watershed de nuevo y la extraemos las líneas de watershed. *(Anexo 2.8)*

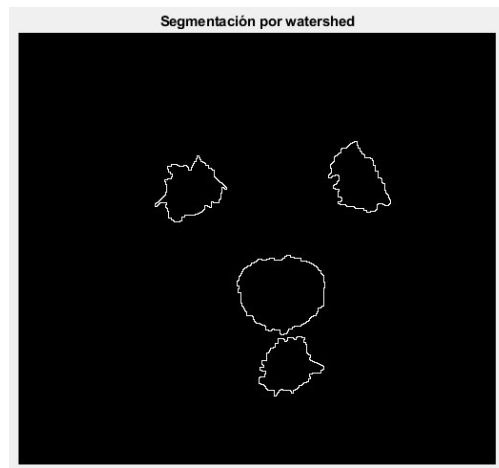


Figura 19. Fronteras obtenidas de la segmentación por watershed.

10. Y finalmente superponemos la imagen resultando 'L_frontera' sobre la imagen de escala de grises recortada, obteniendo el resultado final. *(Anexo 8)*



Figura 20. Resultado final.

ANEXO

Anexo 1.1

```
% Cambiamos la imagen al espacio HSI.
[H,S,I] = rgb2hsi(Img);
figure, subplot(1,3,1), imshow(H), title('Componente H')
subplot(1,3,2), imshow(S), title('Componente S')
subplot(1,3,3), imshow(I), title('Componente I')

% Rango de variación -> lo sabemos visualizando el histograma de cada una.
figure, imhist(H), axis('auto'), title('Histograma Componente H')
figure, imhist(S), axis('auto'), title('Histograma Componente S')
figure, imhist(I), axis('auto'), title('Histograma Componente I')

% Vemos que el rango dinamico esta comprendido entre 0 y 1.

% Escogemos la componente H, ya que destaca más el balón
Componente = H;

% Comprobamos que los valores mas altos, en su mayoría pertenecen al balón.
```

Anexo 1.2

```
% Definimos la máscara y sus coeficientes.
h = 1/49*ones(7,7);
% Aplicamos el filtro.
H_FPB = imfilter(Componente,h);
figure, imshow(H_FPB), title('Componente H suavizada')
figure, imhist(H_FPB), title('Histograma H suavizada')
figure, mesh(H_FPB), title('H suavizada')
```

Anexo 1.3

```
% II. Umbralización y filtrado.

% Visualizamos el histograma de la componente H.
figure, imhist(H_FPB), axis('auto'), title('Histograma Componente H')

% Primero, pasamos a la imagen de la componente H a blanco y negro usando
% segmentación binaria por umbralización.
H_U = im2bw(H_FPB,0.8);
H_U = uint8(H_U*255);
figure, imshow(H_U), title('Img. umbralizada (Componente H) en uint8')
```

Anexo 1.4

```
% Creamos el elemento estructurante.
EE_cuadrado = strel('square', 35);

% Erosión.
Erosion = imerode(H_U, EE_cuadrado);
figure, imshow(Erosion), title('Erosión')

% Dilatación.
Dilatacion = imdilate(H_U, EE_cuadrado);
figure, imshow(Dilatacion), title('Dilatacion')

% Apertura.
Apertura = imopen(H_U, EE_cuadrado);
figure, imshow(Apertura), title('Apertura')

% Cierre.
Cierre = imclose(H_U, EE_cuadrado);
figure, imshow(Cierre), title('Cierre')
```

Anexo 1.5

```
% Utilizaremos la APERTURA para hacer la segmentación, ya que es donde mejor
% se observa el balón

% Para poder quedarnos el balón en primer plano, hacemos el negativo
% de la imagen apertura.
Negativo_Apertura = 255 - Apertura;
figure, imshow(Negativo_Apertura), title('Negativo del Apertura')

% Vamos a obtener la capa de etiquetas ( bwlable utiliza vecindad a 8)
[IM_Seg,n] = bwlable(Negativo_Apertura);
figure, imshow(IM_Seg), title('Capa de etiquetas')

% Pasamos la capa de etiquetas a RGB y la pintamos.
RGB_Segment = label2rgb(IM_Seg);
figure, imshow(RGB_Segment), title('Capa de etiquetas en falso color')
% Ahora si obtenemos el balón y no coge el fondo.
```

Anexo 1.6

```
% Obtenemos el número de objetos.
Num_objetos = max(IM_Seg(:));

% Calculamos el tamaño de los objetos.
imtool(IM_Seg, [])
Props = regionprops(IM_Seg, 'Eccentricity');
% Ver el valor del primer objeto.
Props(1).Eccentricity
```


Anexo 1.7

```
% Obtenemos las fronteras
% Primero definimos el EE(forma y tamaño).
EE_disco = strel('disk', 9); % Mas grande el tamaño, más se marca

% Aplicamos Dilatacion con el EE.
Dilatacion = imdilate(Negativo_Apertura, EE_disco);
figure, imshow(Dilatacion), title('Dilatacion')

% Aplicamos Erosion con el EE
Erosion = imerode(Negativo_Apertura, EE_disco);
figure, imshow(Erosion), title('Erosión')

% Calculamos el gradiente.
Gradiente = Dilatacion - Erosion;
figure, imshow(Gradiente), title('Gradiente')
```

Anexo 1.8

```
% Mapa del gradiente.
Gradiente_map = [0 0 0; 255 0 0];
Gradiente_Rojo = ind2rgb(Gradiente, Gradiente_map);
figure, imshow(Gradiente_Rojo), title('Img. Gradiente Rojo')

% Vemos la marcación del balon sobre la imagen original sumando ambas
% imagenes.
I_Gradiente = uint8(Gradiente_Rojo) + Img;
figure, subplot(1, 2, 1),imshow(Img), title('Imagen original')
subplot(1, 2, 2),imshow(I_Gradiente), title('Imagen segmentada')
```

Anexo 2.1

```
%Cargo la imagen
Img = imread('G15.jpeg');
figure, imshow(Img), title('Imagen Original');
figure, imhist(Img), title('Imagen Original');

% 1.Preprocesado
% Pasamos a nivel de gris
Img = rgb2gray(Img);
figure, imshow(Img), title('Imagen gris');
figure, imhist(Img), title('Imagen gris');

%Recortamos la imagen
Img = Img(350:800, 350:850, :);
figure, imshow(Img), title('Imagen recortada')
```

Anexo 2.2

```
% Hacemos el filtro con radio 1
% Creamos el elemento estructurante.
EE1 = strel('disk', 1);

% Apertura.
Apertura1 = imopen(Img, EE1);
figure, imshow(Apertura1), title('Apertura')

% Cierre.
I_ASF1 = imclose(Apertura1, EE1);
figure, imshow(I_ASF1), title('ASF1')

% Hacemos el filtro con radio 2
EE2 = strel('disk', 2);

% Apertura.
Apertura2 = imopen(I_ASF1, EE2);
figure, imshow(Apertura2), title('Apertura')

% Cierre.
I_ASF2 = imclose(Apertura2, EE2);
figure, imshow(I_ASF2), title('ASF2')

% Hacemos el filtro con radio 20
EE3 = strel('disk', 20);

% Apertura.
Apertura3 = imopen(I_ASF2, EE3);
figure, imshow(Apertura3), title('Apertura')

% Cierre.
I_ASF3 = imclose(Apertura3, EE3);
figure, imshow(I_ASF3), title('ASF3')
```

Anexo 2.3

```
% 2.Segmentación por watershed
% a. Obtención de los marcadores de las zonas de interés

% Hago el negativo
I_neg = imcomplement(I_ASF3);
figure, imshow(I_neg), title('Negativo del filtro')

%Erosionamos el negativo con un un EE_plano con disco 9
% Creamos el elemento estructurante.
EE_disco = strel('disk', 9);

% Erosión.
I_marker = imerode(I_neg, EE_disco);
figure, imshow(I_marker), title('I_marker')

%Reconstruccion morfologica
I_rec = imreconstruct(I_marker, I_neg);
figure, imshow(I_rec), title('Reconstruccion')

%Imagen binaria con los maximos regionales
I_max_reg = imregionalmax(I_rec);
figure, imshow(I_max_reg), title('Máximos regionales');

%Eliminamos los máximos que no zonas de interés
I_max_reg2 = imclearborder(I_max_reg);
figure, imshow(I_max_reg2), title('Máximos regionales sin los bordes');
figure, imtool(I_max_reg2), title('tool maximos regionales');

% Visualizamos la capa de etiquetas en falso color. Blanco en matlab es la etiqueta 0 que
% veíamos en negro en los apuntes.
RGB_I_max_reg2= label2rgb(I_max_reg2);
imtool(RGB_I_max_reg2), title('Capa de etiquetas en falso color')

% Calculamos el tamaño de los objetos.
[I_max_reg3, Nobjetos] = bwlabel(I_max_reg2);
Props = regionprops(I_max_reg3, 'Area');
V_Area = [];
for ind_obj = 1:Nobjetos
    V_Area = [V_Area Props(ind_obj).Area];
end
figure, stem(V_Area)
xlabel('Número de objeto'), ylabel('Tamaño')

% Identificamos que la etiquetas de interes son
V_Interes = [2, 4, 5, 7];

% Bucle que filtra de la imagen binaria I_U las regiones de no interés.
[n_filas, n_cols] = size(I_rec); % nos dice el numero de filas y columnas que tiene la imagen.
```



```

% recorre todas las filas de la imagen
for ind_nfila=1:n_filas
    % recorre todas las columnas de la imagen
    for ind_ncol=1:n_cols
        % Accedemos a la posición de la matriz de la imagen B/N si en la posición indicada tenemos un pixel de
        if I_rec(ind_nfila, ind_ncol)
            % busco si la etiqueta de ese pixel es una de las etiquetas de interes que hemos definido antes.
            numero_et = I_max_reg3(ind_nfila, ind_ncol);
            % si el pixel pertenece a la region de interes, ismember nos
            % devuelve un valor mayor > 0. Concretamente nos devuelve el
            % valor en binario del numero de la etiqueta que pertenece a mis V_Interes.
            if sum(ismember(V_Interes, numero_et)) == 0
                % Si se cumple la condición, borramos el pixel en cuestión,
                % y lo convertimos en un pixel de fondo. Es decir,
                % eliminamos la etiqueta que identifica ese pixel como un
                % objeto y le asignamos la etiqueta de fondo.
                I_rec(ind_nfila, ind_ncol) = 0;
            end
        end
    end
end
end
figure, imshow(I_rec), title('zona de interes')

```

Anexo 2.4

```

% b. Obtención del marcador externo
I_max_reg3 = I_rec;
%Aquí con dilatacion, intentamos que los marcadores externos no
%correspondan a zonas de interés.
% Me dilata la imagen anterior con un EE de disco de radio 7
I_dilate = imdilate(logical(I_max_reg3),strel('disk',7));
%Nos calcula la distancia euclidea de la imagen binaria.
%Distancia de ese pixel a un pixel = 0 mas cercano
D = bwdist(I_dilate);
DL = watershed(D); % Calculamos watershed
bgm = (DL == 0); % Nos va a aplicar las lineas de watershed

%Visualizamos la dilatacion (segun bajo el disco me quedan mas juntas
figure, imshow(I_dilate), title('Imagen dilatada');
figure, mesh(I_dilate), title('Imagen dilatada');
%Visualizamos watershed
figure, imshow(DL), title('Watershed');
figure, mesh(DL), title('Watershed');
%Visualizamos las lineas de watershed
figure, imshow(imadd(255*uint8(bgm),Img)), title('Lineas de watershed');

```

Anexo 2.5

```

% c. Combinacion de marcadores internos con externos

%Combinamos ambos marcadores
I_minimos = bgm | I_max_reg3;
figure, imshow(I_minimos), title('Combinacion de minimos y maximos');

```

Anexo 2.6

```
% d. Modulo del gradiente de la imagen

%Creo el filtro de sobel
H = fspecial('sobel');
Htranspose = H.'; % Hago la transpuesta
FilterH = imfilter(double(Img),H); %Hago el filtrado
FilterHtranspose =imfilter(double(Img),Htranspose);
%Calculamos el modulo del gradiente
Img_grad=sqrt(FilterH.^2+FilterHtranspose.^2);
figure, imshow(Img_grad, []), title('Modulo del gradiente');
```

Anexo 2.7

```
% e. Evitar la sobresegmentación

% Nos va a ayudar a imponer minimos
Img_grad_mrk = imimposemin(Img_grad, I_minimos);
figure, imshow(Img_grad_mrk), title('Imposicion de minimos');
```

Anexo 2.8

```
% f. Aplicar watershed

L_frontera = watershed(Img_grad_mrk);
bgm = (L_frontera == 0);
figure, imshow(bgm), title('Segmentación por watershed');

% g. Imagen frontera

%Visualizamos las lineas de watershed
figure, subplot(1,2,1), imshow(Img), title('Imagen Original');
subplot(1,2,2), imshow(imadd(255*uint8(bgm),Img)), title('RESULTADO FINAL');
```