

IEE239 - Procesamiento de Señales e Imágenes Digitales

Laboratorio 04 - Ejercicios Propuestos

(Primer Semestre 2018)

1. Una expedición peruana se encuentra analizando el fondo del oceano, a 5000 m bajo el nivel del mar. A pesar de llevar instrumentos de iluminación y de alta resolución, la oscuridad propia de la región impide la rápida detección de las características de la flora y fauna acuática.

Una de las imágenes adquiridas se muestra en la Figura 1.



Figure 1: Imagen adquirida por la expedición.

- a. La primera acción a realizar es extraer el histograma de la imagen, con la finalidad de entender la distribución de los píxeles y las intensidades predominantes. Realizarlo de dos formas:

- Usando la función `imhist()`, propia de Matlab.
- Clasificar la intensidad de cada píxel dentro de los bins $\{0,1,\dots,255\}$.

Leer la imagen 'sea.jpg'¹ usando `imread()` y graficarla usando `imshow()`. Luego, calcular el histograma mediante las dos aproximaciones indicadas líneas arriba.

Solución: Primero, se lee la imagen en escala de grises y se muestra.

```
1 grayImage = imread('sea.jpg'); %leer imagen
2 [M, N] = size(grayImage); %tamano de la imagen
3 subplot(2, 2, 1);
4 imshow(grayImage, []);
5 title('Original Grayscale Image', 'FontSize', 12);
6
```

¹La imagen está almacenada en la carpeta 'laboratorio/propuestos/'

```

7 % Calculando histograma manualmente:
8 hmax=255;%maximo bin
9 y=double(grayImage(:));%vectorizar la matriz
10 h=zeros(1,hmax+1);%inicializar el vector donde se construira el histograma
11 %colocar cada punto en el bin correspondiente e ir sumando las ...
    coincidencias
12 for i= 1: 255
13 h( i)= sum( y(:)== i);
14 end
15
16 % Calculando el histograma usando imhist()
17 histograma2=imhist(grayImage(:));
18
19 subplot(211),plot(h)
20 subplot(212),plot(histograma2)

```

La primera forma de obtener el histograma es analizar de forma vectorial todos los píxeles de la imagen simultáneamente para cada nivel de intensidad. Dependiendo el valor de intensidad que posea $\{0,1,\dots,255\}$, será colocado en el vector **h**. Asimismo, se va sumando las coincidencias.

La segunda forma consiste en usar la función `imhist()`, que usa como parámetro de entrada la imagen vectorizada ².

- b. A partir de ello, realizar *Histogram Matching*. Este método de ecualización de histograma consiste en modificar la imagen de forma que el histograma resultante sea similar a un histograma de referencia. Realizar este procedimiento considerando un histograma plano como referencia y empleando la función `histeq()`. Graficar la imagen inicial y la transformada usando `subplot()` e incluyendo títulos adecuados. ¿Es posible revertir el proceso de ecualización?

Solución: El código se muestra a continuación.

```

1 intmax=255; %maximo bin
2 htgt=zeros(1,intmax+1); %inicializar el vector del histograma plano
3 a=round( M*N/(intmax+1) ); %numero de elementos en cada bin. Es ...
    producto de repartir los M*N pixeles equitativamente en cada bin.
4 htgt(1:(intmax+1))=a; %construccion del histograma plano
5 imgeq=histeq(grayImage,htgt); %ecualizacion del histograma usando el ...
    histograma plano construido

```

El proceso de ecualización en este caso es sencillo. Del inciso anterior, se puede ver que la distribución de píxeles no es uniforme en la imagen, pues hay un gran porcentaje de ellos con poca intensidad. Un histograma ideal es uno plano, en el que cada bin dispone del mismo número de píxeles en la imagen. Por ello, se debe construir un histograma plano (representado por la variable *htgt*). Mediante la función `histeq()` se logra que la imagen leída adopte la forma del histograma deseado. Como resultado, se obtiene la variable *imgeq*, que representa a la imagen ecualizada. Por último, se grafica ambas imágenes.

No se puede revertir el proceso de ecualización, pues no se trata de una correspondencia unívoca, siendo imposible recuperar la imagen original.

- c. Realizar una transformación gamma a la imagen 'sea' en escala de grises, de acuerdo a la siguiente expresión:

$$E(x,y) = 255 \cdot \left(\frac{I(x,y)}{255} \right)^\gamma$$

²verificar (:)

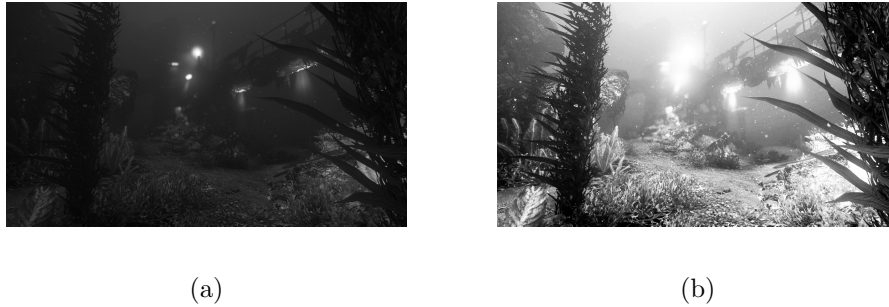


Figure 2: (a) Imagen en escala de grises adquirida por la expedición. (b) Ecualización de la imagen mediante Histogram stretching.

donde $I(x, y)$ corresponde a la imagen de entrada y $E(x, y)$ a la imagen resultante. Considerar $\gamma=2$. Para ello, hacer un cast al tipo double a la imagen 'sea' y aplicar la transformación señalada. Graficar la imagen 'sea' y la imagen resultante en una misma ventana usando `subplot()`.

¿Es posible proponer una transformada que permita revertir el efecto generado en la imagen? Obtener dicha transformación y con ella estimar la imagen original. Luego, verificar si la imagen original y la imagen estimada son similares a partir de `norm`.

Solución:

```

1      in = double(grayImage); %hacer cast a tipo double
2      gamma=2;
3      transformed=255*((in/255).^(gamma)); %aplicar la transformacion gamma
4      subplot(121), imshow(in, [])
5      subplot(122), imshow(transformed, [])
6
7      %la transformada inversa
8      recovery = 255*(transformed/255).^(1/gamma);
9      subplot(121), imshow(transformed, [])
10     subplot(122), imshow(recovery, [])
11
12     a1=norm(in);
13     a2=norm(recovery);

```

Tras la aplicación de la transformación Gamma, la imagen disminuye su brillo, como resultado de una compresión del rango dinámico de los valores mostrados en la imagen. Por otro lado, debido a que la transformación Gamma es una correspondencia unívoca, es posible generar una transformación que permite recuperar las características de la imagen. La norma de la resta de ambas imágenes, sin considerar errores de redondeo, es prácticamente 0.

- d. Como en todo centro de investigación, la gran cantidad de información enviada a la central requiere compresión. Se debe escoger entre i) enviar los bit planes 6 y 7 ii) enviar los bit planes 4, 5, 6, 7. ¿Cuál de ambas opciones permite retener en mayor medida la estructura de la imagen original? Usar las funciones `bitset()` y `bitget()`.³

Solución:

³Si bien es cierto que el bit-plane menos significativo es BP0 (BP0,...,BP7), **Matlab** considera como el menos significativo al BP1 (BP1,...,BP8).

```

1      A=imgeq; %renombrar imagen
2      B=zeros(size(A)); %inicializar variable
3      B=bitset(B,7,bitget(A,7)); %adquirir el bit 7 de A y ubicarlo en ...
      el bit 7 de la nueva variable.
4      B=bitset(B,8,bitget(A,8)); %adquirir el bit 8 de A y ubicarlo en ...
      el bit 8 de la nueva variable. Ahora B tiene el bit 7 y 8 de A.
5      B=uint8(B);
6      figure,subplot(121),imshow(A);
7      subplot(122),imshow(B);
8
9      B2=zeros(size(A)); %inicializar variable
10     B2=bitset(B2,5,bitget(A,5));
11     B2=bitset(B2,6,bitget(A,6));
12     B2=bitset(B2,7,bitget(A,7));
13     B2=bitset(B2,8,bitget(A,8));
14     B2=uint8(B2);
15     figure(2),subplot(121),imshow(A);
16     subplot(122),imshow(B2);

```

El proceso de obtención de bit planes de una imagen es directo. Para el primer caso, se requiere extraer los planos 6 y 7 (7 y 8 en Matlab). Así, se genera una nueva variable B, se extrae el bit plane 6 de A y es ubicado en el bit plane 6 de B. Para el bit plane 7, es extraído de A y ubicado en el bit plane 7 de B. De esta manera, B solo posee de dos bit planes: 6 y 7, a diferencia de A.

El proceso para extraer los bit planes 4, 5, 6 y 7 es idéntico. Al final, se encuentra que en este último caso, se pierde menor cantidad de características.

- e. Realizar highboost filtering de la imagen original 'sea.jpg'. Para ello, primero suavizar la imagen usando un filtro mediana de tamaño 7. Considerar $k = 2$.

```

1      I=grayImage;
2      If=medfilt2(I,[7 7]);
3      difference=I-If;
4      k=2;
5      G=I+k*difference;
6      subplot(221),imshow(I)
7      subplot(222),imshow(If)
8      subplot(223),imshow(difference)
9      subplot(224),imshow(G)

```

Con el filtro mediana se consigue una imagen suavizada. Posteriormente, se genera una máscara mediante la diferencia de la imagen suavizada y la original. Por último, la máscara multiplicada por un factor k permite mejorar los bordes de la imagen original.

2. Los dueños del zoológico de los Ángeles, California, reportaron la desaparición de uno de los felinos que tenían en cautiverio. Coincidentemente, la policía recibió diversos reportes sobre el avistamiento de un felino en el bosque de la ciudad. Ante esto, se decide ubicar cámaras para monitorear la ubicación del animal. Solo dos de ellas reportaron su presencia, brindando las imágenes mostradas en la Figura 3.

Para colaborar con la investigación, y confirmar si ambas retrataron al felino fugitivo o se trata de dos ejemplares distintos, se le encomienda las siguientes tareas:

- a. Leer la imagen 'camera1.jpg' y 'camera2.jpg'⁴. Transformar ambas a escala de grises

⁴La imagen está almacenada en la carpeta 'laboratorio/propuestos/'

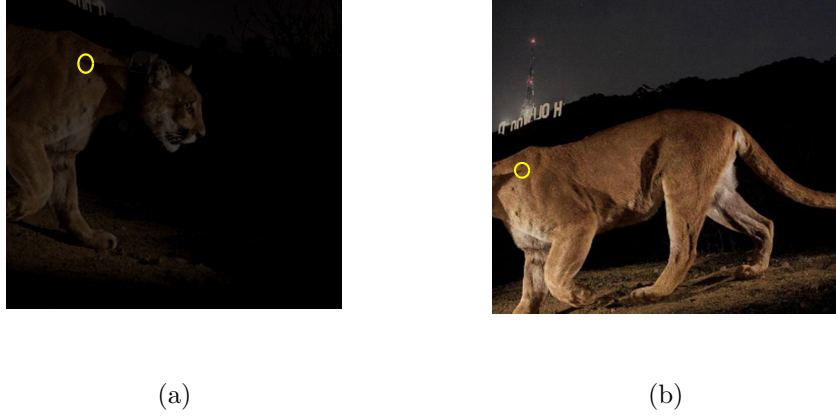


Figure 3: (a) Fotografía adquirida con la cámara 1. (b) Fotografía adquirida con la cámara 2.

empleando la función `rgb2gray()`⁵. Verificar el tamaño de cada una. En base a la relación de tamaño entre ambas, aplicar un cambio de escala a la imagen de menor tamaño por un factor de dos usando interpolación bilineal (`interp2()`) y conseguir que ambas tengan el mismo tamaño.

- b. La imagen adquirida por la cámara 1 presenta poco contraste, probablemente debido a su ubicación. Aplicar contrast stretching. Hacer un cast a la imagen a tipo double usando `double()` y normalizar la intensidad. Aplicar contrast stretching siguiendo la fórmula 1.

$$I_{\text{contrast}}(x, y) = \frac{1}{1 + \left(\frac{m}{I(x, y) + \text{eps}}\right)^E} \quad (1)$$

donde m corresponde al valor promedio de $I(x, y)$ y E controla la pendiente de la transformación aplicada (transición entre blanco y negro). Realizar esto para $E \in \{0.1, 1, 4\}$ y mostrar los resultados en una misma ventana para los tres casos. (usar `subplot`)

- c. Sobre la segunda imagen, aplicar la transformación de intensidad sigmoide, dada por:

$$I_{\text{contrast}}(x, y) = \frac{-\cos(\pi I(x, y)) + 1}{2} \quad (2)$$

donde $I(x, y)$ es una imagen en escala de grises con intensidad normalizada. Graficar la imagen original en escala de grises e I_{contrast} en una misma ventana con los títulos respectivos.

- d. Datos posteriores indicaron que la cámara 2 está ubicada al lado opuesto de la cámara 1. Con este dato, para poder interpretar mejor las imágenes, se debe reflejar la imagen 'camera2.jpg' horizontalmente, según:

$$I^*(x, y) = I(-x, y)$$

- e. Considerando que el punto señalado en la Figura 3 (círculos amarillos) para 'camera1' está ubicado en el punto $\mathbf{x}_1 = (135, 110)$ y para 'camera2.jpg' está en $\mathbf{x}_2 = (78; 462)$, realizar la unión de ambas imágenes en una sola (generar una matriz inicial de ceros de tamaño 600

⁵Esta función convierte las imágenes RGB a escala de grises mediante la eliminación de la información de Hue y Saturación, conservando la información de luminosidad. La luminosidad es calculada a partir de la suma ponderada de las capas RGB: $Y(x, y) = 0.2126R(x, y) + 0.7152G(x, y) + 0.0722B(x, y)$.

× 1800, unir ambas imágenes según los puntos guía dados mediante el desplazamiento de una de ellas y ubicarlos en la matriz de ceros, formando así la imagen completa). Mostrar en una misma ventana a las figuras iniciales y la imagen final obtenida.

- f. Sobre la imagen de salida del ítem anterior, añadir ruido impulsivo usando `imnoise()` con una densidad de 0.02. Realizar el filtrado espacial de la imagen mediante el uso del siguiente kernel:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Luego, aplicar un filtro mediana sobre la imagen con ruido impulsivo usando `medfilt2()`. ¿Con cuál de ellos se logró remover el ruido de mejor manera? ¿Qué consecuencias adicionales produjo el filtrado sobre la imagen (difuminación)?

3. El Departamento de Policía de Los Ángeles recibe un reporte de robo vehicular así como una imagen instantánea aérea del vehículo en cuestión. Asimismo, recibe información de la calle por la que se sospecha que transita el vehículo robado, por lo que se le encomienda dirigirse a la oficina de videovigilancia para capturar una instantánea de la calle en cuestión para su posterior análisis.

- a. Al llegar a la oficina de videovigilancia, lo contacta la banda de ladrones (a quienes usted reporta como infiltrado en el departamento de policía), exigiéndole que altere la instantánea antes de enviarla al recinto policial, para hacer el vehículo indistinguible. Para ello, cargar y normalizar la imagen instantánea `'cuadro.jpg'`⁶, y aplicarle una transformación gamma de valor 4, dada por $T\{f(x,y)\} = (f(x,y))^4$ (donde la potencia se aplica punto a punto). Graficar en una sola figura la imagen original y la imagen resultante tras la transformación, así como sus respectivos histogramas. ¿Cuál es el efecto de la transformación en la imagen? Explicar.

Rutinas a emplear: `subplot()`, `imhist()`, `imshow()`.

Solución:

```
1 % Carga de imagen
2 Image = double(rgb2gray(imread('Cuadro.jpg')))/255;
3 Gamma = 4;
4 %Transformacion Gamma y graficas correspondientes
5 I2 = Image.^Gamma; %Cabe recalcar que debido a que los pixeles de la ...
    imagen original se encuentran en el rango [0, 1], no es necesario ...
    un factor de escalamiento en la transformacion, pues el resultado ...
    seguira teniendo el mismo rango de valores
6 figure; hold on;
7 subplot(221); imshow(Image); title('Original');
8 subplot(222); imshow(I2); title('Imagen resultante');
9 subplot(223); imhist(Image); title('Histograma original');
10 subplot(224); imhist(I2); title('Histograma imagen resultante');
```

Como puede apreciarse en el resultado gráfico, la transformación Gamma con potencia superior a la unidad reduce el contraste en baja intensidad, generando sombras más oscuras. Este efecto también puede observarse en el histograma, en el cual la distribución de valores se acumula en rangos cercanos a cero.

⁶La imagen está almacenada en la carpeta `'laboratorio/propuestos/'`

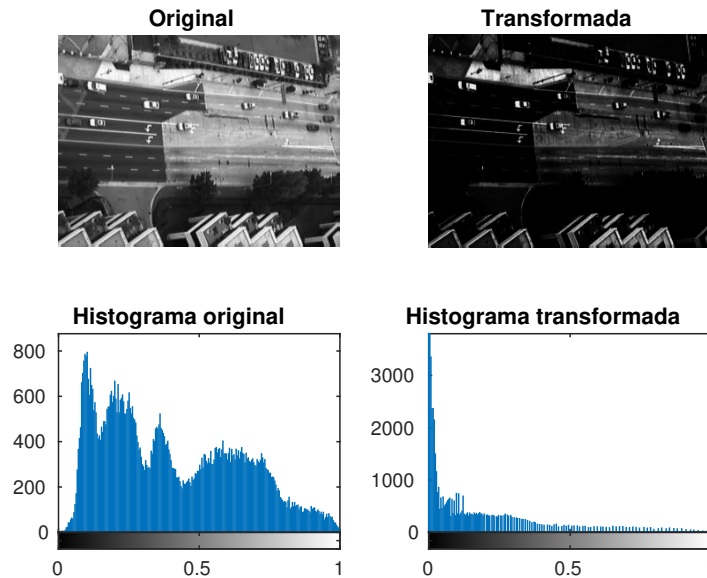


Figure 4: Resultado obtenido mediante transformación gamma.

- b. Sin embargo, los ladrones desconocen que usted es en realidad un doble agente encubierto, y que empleó su astucia y conocimiento del procesamiento de imágenes para alterar la imagen de forma reversible. Determinar la transformación gamma inversa necesaria y aplicarla a la imagen obtenida en a.. Verificar que el resultado corresponde a la imagen original usando la distancia euclidiana, dada por:

$$d(x, y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$$

donde x e y corresponden a los vectores (en este caso, imágenes) a comparar (use para ello el comando **norm()**). ¿Por qué fue posible reconstruir la imagen original?

Solución:

```
1 %Factor inverso y transformacion respectiva
2 Gamma_inv = 1/Gamma;
3 Irec = I2.^Gamma_inv;
4 %Verificacion de la imagen recuperada
5 dif = norm(Image(:)-Irec(:));
```

Puesto que la transformación Gamma tiene una correspondencia unívoca para cada valor de intensidad, su efecto es fácilmente reversible mediante un factor inverso. A manera de prueba, se puede observar que la máxima diferencia entre los píxeles de la imagen original y la recuperada es cero (evaluar expresión *dif*).

- c. Para localizar el vehículo, es necesario detectar primero los objetos en movimiento calculando el valor absoluto de la diferencia entre el cuadro reconstruido en b. y el fondo estático⁷ contenido en la imagen 'fondo.jpg'⁸. Graficar la imagen diferencia obtenida, así como su histograma.

⁷Imagen que contiene información de los píxeles correspondientes a objetos estáticos

⁸La imagen está almacenada en la carpeta 'laboratorio/propuestos/'

Rutinas a emplear: `subplot()`, `imhist()`, `imshow()`, `abs()`.

Solución:

```
1 %Cargando imagen en cuestion
2 backgd_img = double(rgb2gray(imread('fondo.jpg')))/255;
3 Calculando diferencia y graficado resultado
4 moving = abs(Image - backgd_img);
5 figure; hold on
6 subplot(121); imshow(moving); title('Imagen diferencia');
7 subplot(122); imhist(moving); title('Histograma');
```

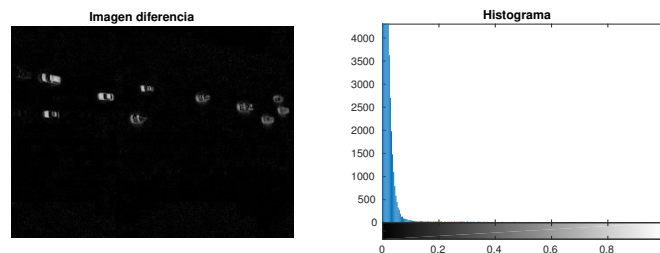


Figure 5: Diferencia entre imagen y fondo estático.

La sustracción del fondo estático es un proceso común en el análisis de secuencias de imágenes. En este caso, se puede observar que el resultado contiene principalmente los vehículos en movimiento. En la distribución de valores en el histograma, puede observar que la mayoría de píxeles tienen valores cercanos a cero, ello debido a que un gran porcentaje de la imagen está constituido por el fondo (más adelante en el curso se verá que este histograma corresponde a una distribución unimodal, casi como los métodos de segmentación automática apropiados).

- d. Para suavizar la imagen obtenida en c. antes de detectar los vehículos, aplicarle un filtro promedio de tamaño 3×3 mediante el comando `conv2()`. Graficar la imagen e histograma resultantes, y comentar el efecto del filtro sobre el resultado.

Rutinas a emplear: `subplot()`, `imhist()`, `imshow()`.

Solución:

```
1 H=ones(3,3)/9;
2 B=conv2(moving, H, 'valid'); %Permite uniformizar las regiones con ...
    valores cercanos en la imagen
3 subplot(121); imshow(B); title('Imagen filtrada');
4 subplot(122); imhist(B); title('Histograma');
```

El uso de un filtro promedio permite suavizar la imagen diferencia, así como eliminar pequeños artefactos producto de la resta. Si bien la diferencia es visualmente difícil de apreciar debido a los bajos valores de intensidad de la imagen diferencia, su beneficio se hace evidente en el siguiente acápite.

- e. Para poder detectar los objetos es necesario primero binarizar la imagen. Para ello, use la herramienta **Data cursor** en la imagen e histograma obtenidos en d. para encontrar un valor umbral x_{umbral} que permita separar los píxeles de fondo y aquellos que contienen objetos en movimiento. Una vez escogido el umbral, utilizar operaciones de indexado para binarizar la imagen diferencia obtenida en c. (sin filtrar) y la imagen obtenida en d. (filtrada). ¿Qué diferencia observa?

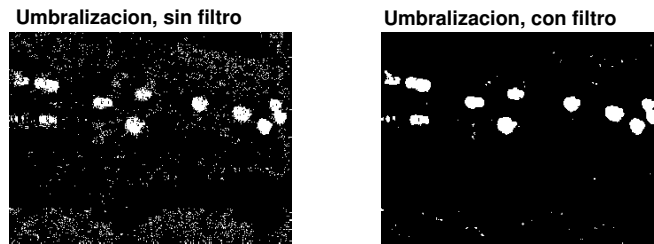


Figure 6: Resultado del proceso de umbralización en las dos imágenes evaluadas

Solución:

```

1 %Declaracion del umbral (a partir de inspeccion del histograma mediante ...
  el Data cursor) y binarizacion de las imagenes
2 x_umbral = 0.0392;
3 mask = (B > x_umbral);
4 mask2 = (moving > x_umbral);
5 %Graficando resultados
6 figure; hold on;
7 subplot(121); imshow(mask2); title('Umbralizacion, sin filtro');
8 subplot(122); imshow(mask); title('Umbralizacion, con filtro');

```

Dado el histograma de la imagen diferencia, se escoge el valor 0.0392 para denotar el límite de los valores de fondo (puesto que este valor se escogió heurísticamente, puede probar cambiarlo por otro y observar el resultado) y se binariza tanto la imagen filtrada como la diferencia original. Se puede observar que la imagen binarizada a partir de la original contiene un mayor número de artefactos (debido a diferencias numéricas menores entre el fondo proporcionado y el fondo real de la imagen) que pueden ser interpretados como ruido, el cual ha sido significativamente reducido en la imagen filtrada.

- f. Detectar los objetos binarios y sus respectivas propiedades en la imagen haciendo un análisis de componentes conectados, empleando las funciones **bwconncomp()** y **regionprops()** (la cual recibe como entrada una imagen binaria y analiza propiedades de las regiones conectadas como área, centroide, etc.). ¿Cuántos objetos contiene la imagen? Si todos los autos ocupan más de doscientos píxeles en la imagen, ¿Cuántos autos hay en la escena?

Solución:

```

1 %Análisis de componentes conectados
2 CC = bwconncomp(mask);
3 ST=regionprops(mask, 'basic');
4 %Almacenamiento de valores de area de cada objeto
5 Area = zeros(CC.NumObjects,1);
6 for l=1:CC.NumObjects
7     Area(l)=ST(l).Area;
8 end
9 %Conteo de autos
10 numAutos = sum(Area > 200)

```

Como se puede observar en la imagen ilustrativa, la función **regionprops()** proporciona tres propiedades útiles de las regiones conectadas (con 8 adyacencia) de la imagen binaria:

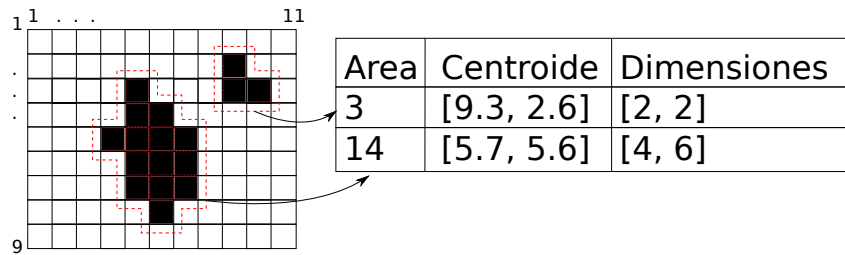


Figure 7: Imagen ilustrativa del análisis de componentes conectados (función **regionprops()**)

area, centroide y 'bounding box' (limites cartesianos que ocupa cada objeto en la imagen). En base a ello, se puede iterar directamente sobre las áreas para estimar cuantos autos hay en la escena (en este caso, 7).

- g. Utilizando la información obtenida en los campos '**BoundingBox**' de los objetos detectados, realizar la búsqueda del auto de interés. De forma iterativa para cada objeto, calcular el histograma asociado a su respectivo 'bounding box' (convertir las coordenadas a valores enteros para poder indexar, usando la función **fix()**), y comparar con el histograma del auto de referencia '**imref.jpg**'. Realizar la comparación empleando la distancia euclidiana (cuya expresión analítica es proporcionada en b.) mediante el comando **norm()**. Graficar el objeto cuya distancia a la imagen de referencia sea menor. ¿Coincide con el auto buscado?

Solución:

```

1  %Imagen a color
2  imrgb = double((imread(img_name)))/255;
3  %Imagen 'bounding box' de referencia
4  boxref = double((imread('imref.jpg')))/255;
5  %Histogramas RGB de referencia para el calculo de la distancia
6  cref=zeros(3,255);
7  [cref(1,:) ~] = imhist(boxref(:,:,1));
8  [cref(2,:) ~] = imhist(boxref(:,:,2));
9  [cref(3,:) ~] = imhist(boxref(:,:,3));
10 %Inicializando distancia minima
11 min_dist=inf;
12 for i=1:28
13     %Convirtiendo las coordenadas y dimensiones del 'bounding box' a ...
        valores enteros
14     BB = fix(ST(i).BoundingBox);
15     box = imrgb(BB(2)+1:BB(2)+BB(4),BB(1)+1:BB(1)+BB(3),:);
16     %Calculando la distancia euclidiana en cada capa RGB, y evaluando ...
        la distancia total
17     for k =1:3
18         [c,x] = imhist(box(:,:,k));
19         d(k) = norm(c-cref(i,:));
20     end
21     dist=norm(d);
22     %Evaluando si la distancia obtenida es la menor. De ser asi, se ...
        almacena el indice actual
23     if dist < min_dist
24         min_dist = dist;
25         idx = i;
26     end
27 end
28 %Mostrando objeto obtenido

```

```

29 BB = fix(ST(idx).BoundingBox);
30 box = imrgb(BB(2)+1:BB(2)+BB(4), BB(1)+1:BB(1)+BB(3), :);
31 figure; imshow(box)

```

Como se puede observar, a partir de un histograma de referencia es bastante directo iterar a través de un 'banco' de imágenes (en este caso, el conjunto de objetos detectados por **regionprops()**) y evaluar la similitud (mediante distancia euclidiana en este caso) entre cada objeto y la referencia. Una vez localizado el objeto, se puede corroborar de forma gráfica que el resultado coincide con el vehículo de interés,

4. Un equipo de médicos solicita a una firma de ingenieros un método que permita mejorar mediante software las imágenes obtenidas mediante baciloscopia por fluorescencia, entregándoles una imagen de referencia en la que los bacilos aparecen de color amarillo brillante, y el fondo en color negro. Se le pide evaluar distintas alternativas:



Figure 8: Referencia de baciloscopia por fluorescencia.

- a. Convertir la imagen del espacio de colores RGB a escala de grises (función **rgb2gray()**), normalizar y observar la distribución de valores en la imagen mediante su histograma (función **imhist()**). ¿Es posible distinguir el rango de valores correspondiente a los bacilos y el fondo a partir del histograma? Seleccionar un valor x_{umbral} que separe ambos rangos, y utilizar la función **im2bw()** para umbralizar la imagen con dicho valor. ¿Se logra mantener los bacilos en la imagen binaria?
- b. Localizar el rango de valores $[s_0, s_1]$ en el que se encuentran los bacilos, y aplicar las siguientes transformaciones de intensity-level slicing:
 - i. Asignar el valor 1 a todos los píxeles en el rango de interés, y el valor 0 a todos los que estén fuera de dicho rango. Esta transformación está denotada por la expresión:

$$T\{f(x, y)\} = \begin{cases} 1, & f(x, y) \in [s_0, s_1] \\ 0, & \text{otros casos} \end{cases}$$

Graficar el resultado obtenido.

- ii. Asignar el valor s_1 a todos los píxeles en el rango de interés, y matener el valor de todos los píxeles fuera de dicho rango. Esta transformación está denotada por la expresión:

$$T\{f(x, y)\} = \begin{cases} s_1, & f(x, y) \in [s_0, s_1] \\ f(x, y), & \text{otros casos} \end{cases}$$

Graficar el resultado obtenido.

¿En qué circunstancia(s) considera que cada una de estas transformaciones sería preferible a la otra? Considere que en ocasiones el contexto visual del objeto de interés también puede ser relevante para el análisis.

- c. Evaluar las siguientes alternativas para mejorar el contraste de la imagen:
 - i. Ecualizar el histograma de la imagen mediante la función **histeq()**, y graficar el nuevo histograma. ¿Qué cambios observa respecto al histograma original?
 - ii. Utilizando el comando **imadjust()**, realizar una transformación que permita realzar el contraste en la imagen (revisar la transformación que la función realiza por defecto en su documentación). ¿Qué diferencia observa respecto a lo obtenido en i.?
 - iii. Cambiar el espacio de colores a HSV (modelo de color alternativo al RGB, cuyas siglas corresponden a Hue, Saturation, Brightness – Matiz, Saturación, Brillo respectivamente) mediante el comando **rgb2hsv()**. Aplicar las técnicas realizadas en i. y ii. en la capa V (tercera capa) de la imagen. Graficar el resultado. ¿Tiene algún beneficio mejorar el contraste en este espacio de color? Explicar claramente su respuesta.
- d. Al intentar tomar una nueva imagen, una falla de hardware distorsiona el resultado ocasionando un shear en el eje horizontal y otro en el eje vertical, dando como resultado la imagen '**distorted.jpg**'. Para recuperar la imagen original se le pide lo siguiente:

- i. Realizar un shear vertical de factor -0.5, considerando para ello la siguiente transformación de coordenadas:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

donde las coordenadas (v, w) corresponden a la imagen original (distorsionada), y (x, y) corresponden a la imagen tras la transformación (recuperada). Realizar la transformación empleando la función **imtransform()** (cuyo argumento de entrada requiere el uso de la función **maketform()**), y considerando interpolación bilineal (revisar documentación de **imtransform()**). Graficar la imagen obtenida.

- ii. Realizar un shear horizontal de factor -0.5 sobre la imagen obtenida en i., considerando para ello la siguiente transformación de coordenadas:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

empleando para ello las mismas funciones que en i.. Realice la transformación considerando tres tipos de interpolación: bilineal, vecinos más cercanos y bicúbica (revisar help de **imtransform()**). Graficar las imágenes obtenidas en cada caso ¿Qué interpolación proporciona mejor resultado?

- e. Tras realizar una reparación 'casera' al hardware del microscopio para evitar el problema de distorsión, se toma una nueva instantánea y se descubre que el cambio ocasionó que se acoplara ruido blanco aditivo a la imagen generada, como se puede observar en la imagen '**noisy.jpg**'. Para solucionar el problema se le pide evaluar las siguientes alternativas:
 - i. Generar un filtro promedio de tamaño 3×3 (utilizando la función **fspecial()**), y convolucionarlo con la imagen ruidosa mediante el comando **conv2()**. Graficar la imagen ruidosa y la filtrada. ¿Logra reducirse el ruido presente en la imagen? Explicar.
 - ii. Generar un filtro laplaciano de tamaño 3×3 (utilizando la función **fspecial()**) considerando un valor $\alpha = 0.3$, y convolucionarlo con la imagen ruidosa mediante el

comando **conv2()**. Graficar la imagen ruidosa y la filtrada. ¿Logra reducirse el ruido presente en la imagen? ¿Cual es el efecto del filtro laplaciano sobre las discontinuidades en la imagen? Explicar.

- f. Utilizar el comando **imcomplement()** para obtener la imagen negativa y graficar su histograma. ¿A qué se debe el efecto de 'espejo' respecto al histograma original?