Informe laboratorio - Procesamiento de Imágenes Digitales

Histogram Equalization and Matching

Ruben Rodriguez

21 de julio de 2022

Índice

1. Desarrollo

1. Ecualizacion del Histograma

   a) Implementar el metodo para aplicar la ecualizacion del histograma a una imagen.

   La funcion(histogramEqualization) recibe la imagen y su rango de color por ejemplo 256. Esta misma devuelve un mapa de las intensidades procesadas.

   Para aplicar este mapa sobre la imagen utilizamos otro metodo implementado llamado mappingImage. Gracias a que en la funcion histogramEqualization devuelve un mapa ordenado de las intensidades podemos tratar la intesidad de cada pixel en la imagen original como el indice de ese mapa obtenido, facilitando la asignacion de la nueva intensidad.

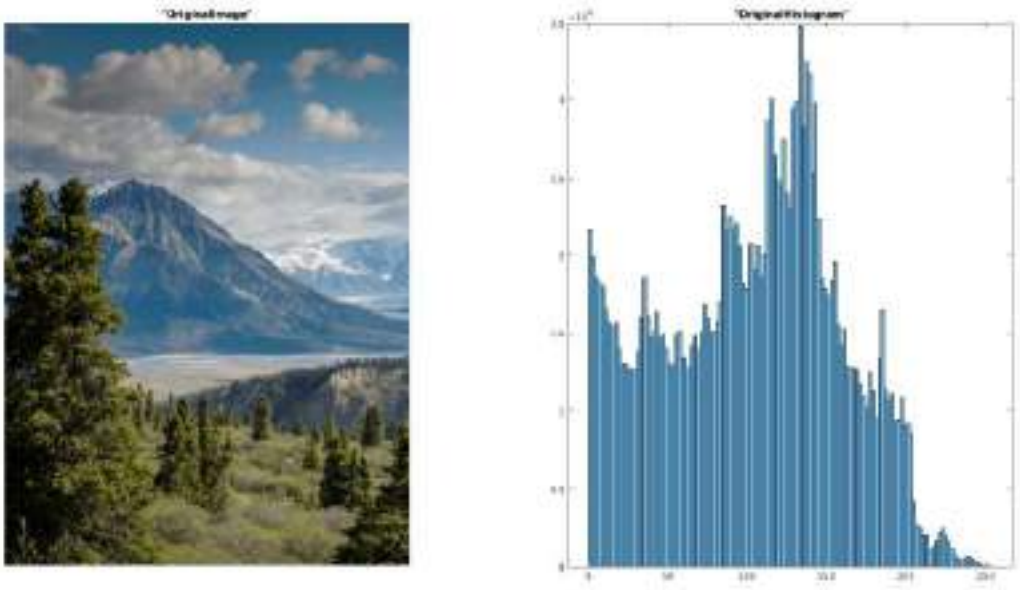

Figura 1: Imagen y Histograma originales.

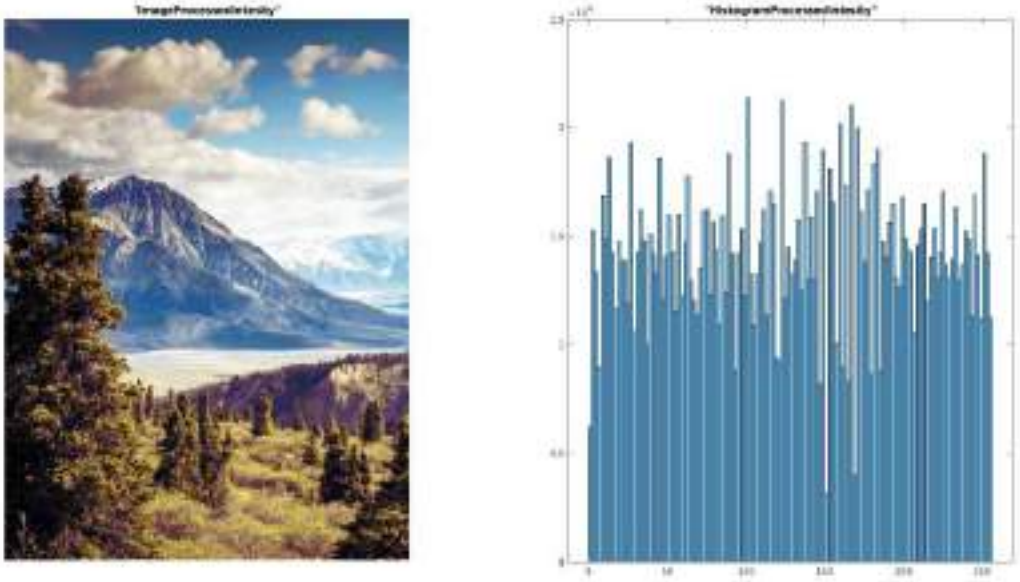

Figura 2: Imagen y Histograma procesados.

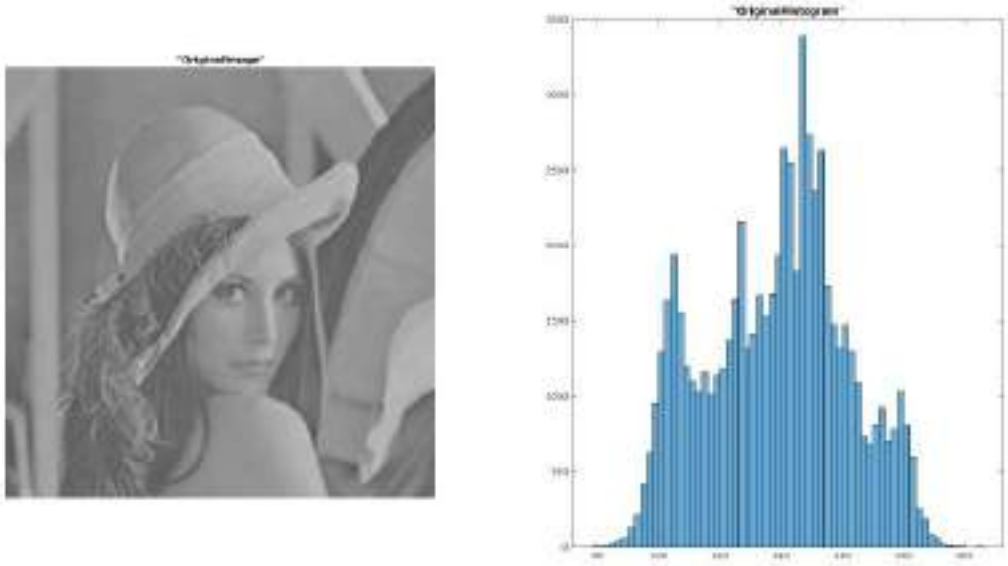

Figura 3: Imagen y Histograma originales.

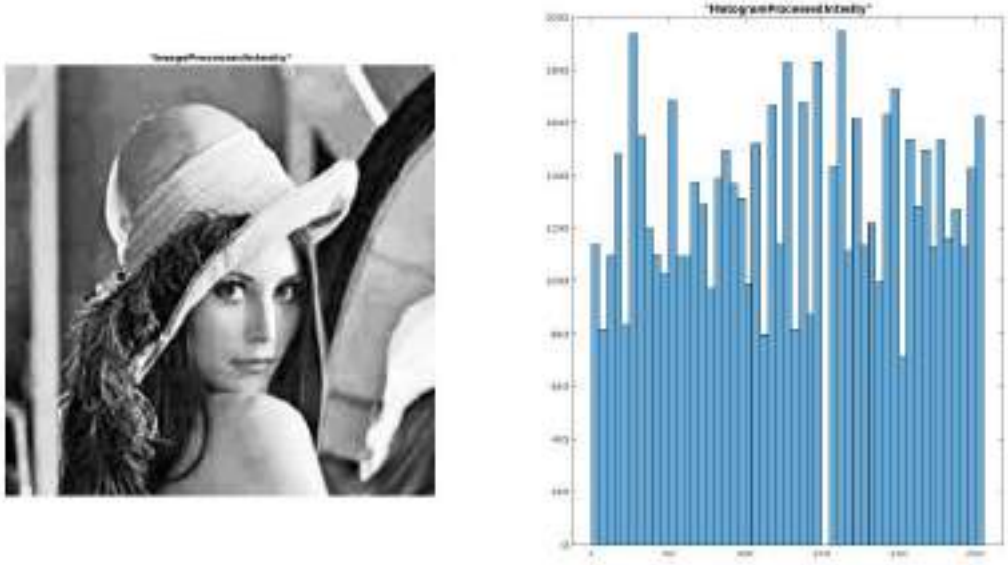

Figura 4: Imagen y Histograma procesados.

2. Histogram Matching

   a) Implementar el metodo para el Histogram Matching.

   Primeramente necesitamos un histograma dado, en mi caso utilice el histograma de una imagen de referencia. A este imagen de referencia le aplicamos la funcion histogramEqualization para obtener el histograma ya ecualizado de referencia.

   Ahora utilizamos la funcion histogramMatching que recibe dos mapas procesados el de la imagen orignal y el de referencia, y esta nos retorna un nuevo mapa. Este mapa obtenido lo podemos pasar por la funcion mappingImage para aplicarlo a la imagen.

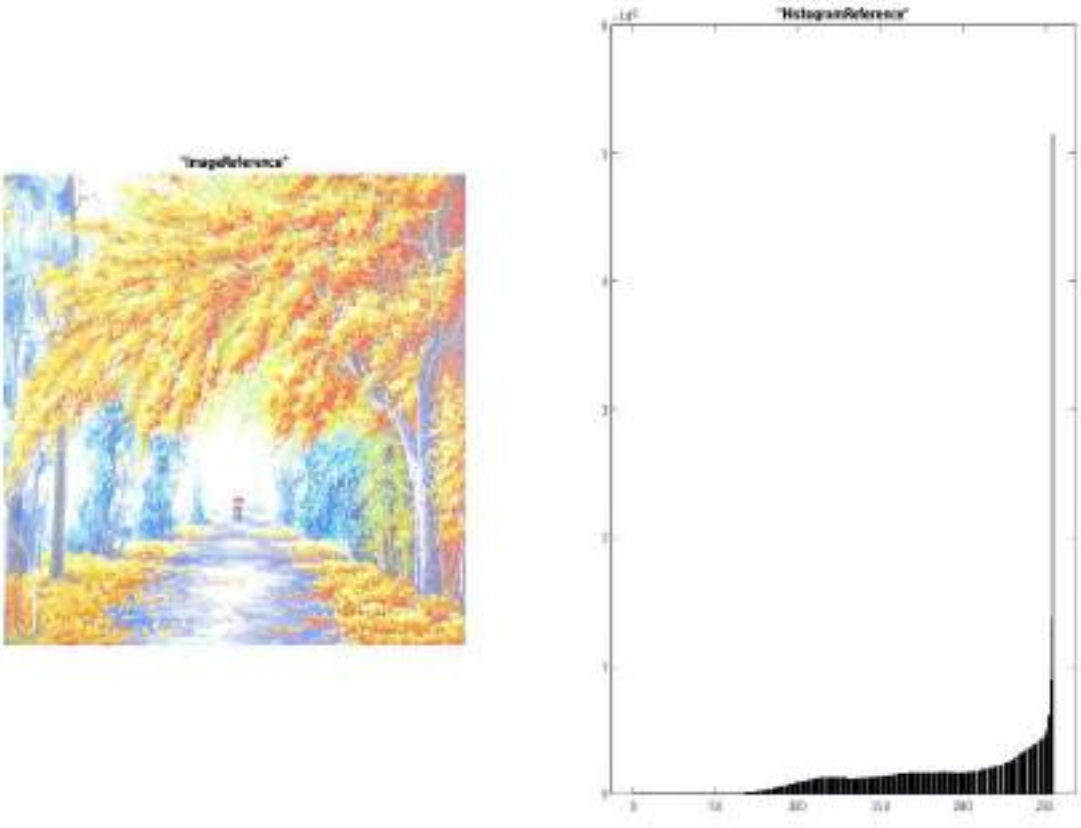

Figura 5: Imagen e Histograma de referencia.

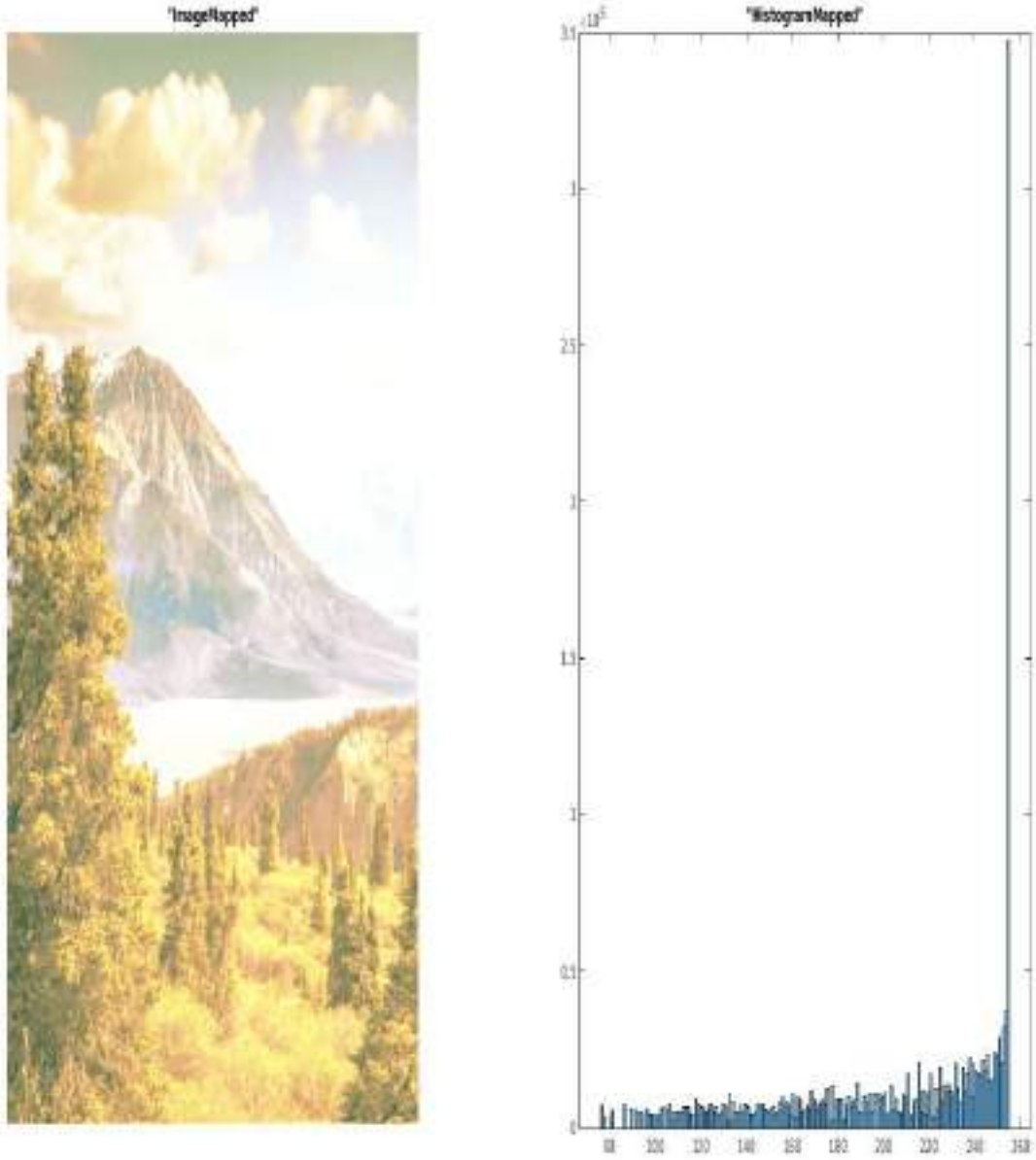

Figura 6: Imagen y nuevo Histograma de la imagen original aplicando histogram matching.

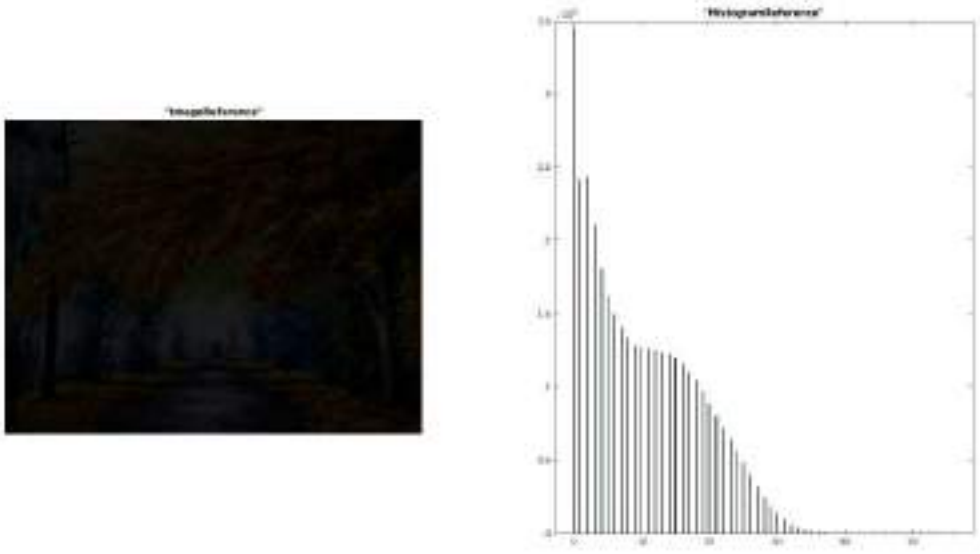

Figura 7: Imagen y Histograma referencia.

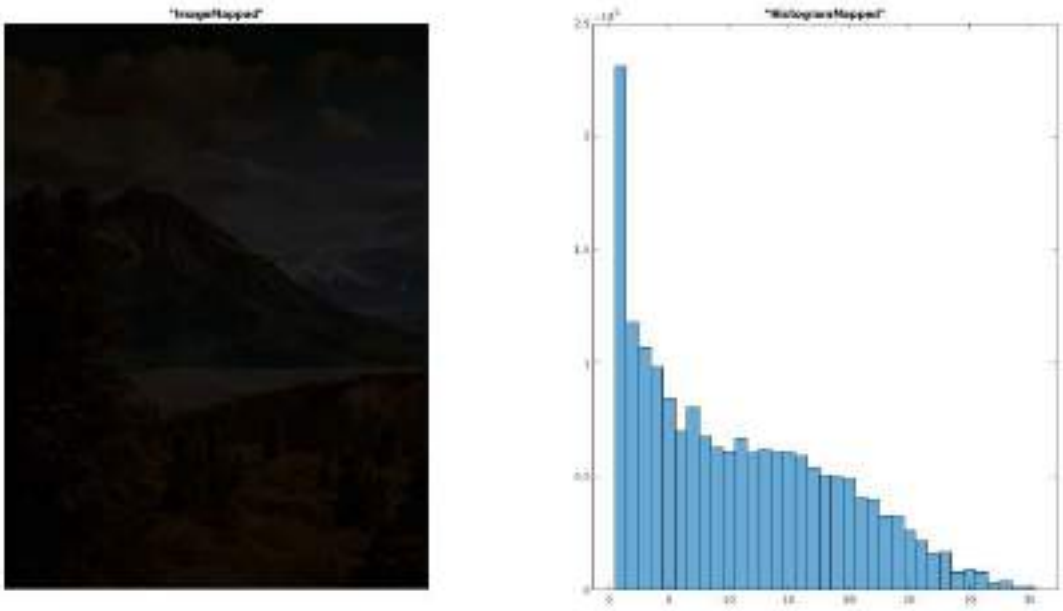

Figura 8: Imagen y nuevo Histograma de la imagen original aplicando histogram matching.

2. Anexo (CODIGO)

Código de matlab.

```
% ORIGINAL IMAGE
I = imread("airplanes.png");
figure
subplot(1,2,1); imshow(I); title "OriginalImage"
subplot(1,2,2); histogram(I); title "OriginalHistogram"

% NEW IMAGE WITH EQUALIZED HISTOGRAM
mapProcessedIntesity = histogramEqualization(I,256);
imageProcessedIntesity = mappingImage(I,mapProcessedIntesity);

figure
subplot(1,2,1); imshow(imageProcessedIntesity); title "ImageProcessedIntesity"
subplot(1,2,2); histogram(imageProcessedIntesity); title
    "HistogramProcessedIntesity"

% LOAD IMAGE REFERENCE
Iref = imread("claro.png");

figure
subplot(1,2,1); imshow(Iref); title "ImageReference"
subplot(1,2,2); histogram(Iref); title "HistogramReference"

% GENERATE EQUALIZED HISTOGRAM OF REFERENCE/OBJETIVE
mapRef = histogramEqualization(Iref,256);

% FUNCTION THAT MAPPED EQUALIZED MAPS
newmap = hitogramMatching(mapProcessedIntesity,mapRef);

% FUNCTION THAT ASSIGN NEW PIXEL VALUE TO THE IMAGE
newimg = mappingImage(I,newmap);

figure
subplot(1,2,1); imshow(newimg); title "ImageMapped"
subplot(1,2,2); histogram(newimg); title "HistogramMapped"

-
function [mapProcessedIntesity] = histogramEqualization(image, imageColorRange)

```

```
38      [sizeY, sizeX, sizeZ] = size(image);
39
40      mapProcessedIntesity = zeros(imageColorRange,sizeZ);
41
42      for i=1:sizeZ
43
44          band = image(:,:,i);
45          [count, ~] = imhist(band,imageColorRange);
46          processedIntesity = 0;
47
48          for j=1:imageColorRange
49              processedIntesity = processedIntesity + ((imageColorRange -
                    1)/(sizeY*sizeX))*count(j);
50              mapProcessedIntesity(j,i) = round(processedIntesity);
51          end
52
53      end
54
55  end
56  -
57  function [img] = mappingImage(image,map)
58
59      [sizeY, sizeX, sizeZ] = size(image);
60
61      for z=1:sizeZ
62          for y=1:sizeY
63              for x=1:sizeX
64                  image(y,x,z) = map(image(y,x,z) + 1,z);
65              end
66          end
67      end
68
69      img = image;
70
71  end
72  -
73  function [map] = hitogramMatching(mapImage,mapRef)
74
75      [sizeY,sizeX] = size(mapRef);
76
77      for i=1:sizeX
78          for j=1:sizeY
```

```
79              sk = mapImage(j,i);
80              [~,index] = min(abs(sk - mapRef(:,i)));
81              mapImage(j,i) = index;
82          end
83      end
84
85      map = mapImage;
86  end
```