

Duarte García Jesús, Nava Vargas Cathia Michelle.

Práctica III. Operaciones Básicas con Imágenes

Resumen—La practica III consistió en tres etapas diferentes, la primera en la realización de operaciones básicas en una imagen digital (inversión de niveles de grises, rotación de noventa grados, reflejo en espejo, recorte de una porción significativa, transformación logarítmica y corrección gamma) con implementación de ciclos y con comandos de MATLAB. La segunda fase consistió en colocar el promedio manualmente de ciertas imágenes ruidosas (promedio elemento a elemento), con la finalidad de que se comparara el resultado de promediar dichas imágenes con cada una de las imágenes con turbulencia, y con la imagen sin turbulencia, de esta manera describimos en qué casos hay mejoras y en cuales el promedio no ayuda en lo absoluto. Por último en la tercera parte trabajamos con el histograma de una imagen con determinadas características. Cabe mencionar que en las tres etapas el programa que se utilizó fue MATLAB.

I. INTRODUCCIÓN

ESTE documento presenta lo realizado, los procedimientos utilizados, resultados obtenidos y conclusiones a las que llegó el equipo, durante la realización de la tercera practica. Algunas cosas de esta práctica que mayormente utilizamos en imágenes médicas son: la inversión de niveles de grises o negativo, lo cual nos ayuda a resaltar aquellos detalles en blanco cuando tenemos como fondo un color negro u oscuro, el aprender a realizar el realce de una imagen ayuda a obtener una imagen mejorada para la aplicación, en este caso medica, al mismo tiempo el logar disminuir u aumentar el brillo, resaltando zonas de nuestro interés.

Esta clase de conceptos que debemos manejar con mayor facilidad son por lo general requisitos comunes en imágenes médicas, por lo tanto para nosotros son de alta utilidad.

A.Ecuaciones

$$S_c = T(r_c) = \sum_{i=1}^L p_i(r_i) = \sum_{i=1}^L \frac{r_i}{L} \quad (1)$$

II. DESARROLLO

La práctica se divide en tres etapas, operaciones básicas, promedio e histograma, para cada una de ellas se usó el programa MATLAB.

En la primera parte “operaciones básicas”, trabajamos con una imagen cuadrada de una capa y ocho bits de intensidad, en esta imagen realizamos en específico seis operaciones (inversión de niveles de grises, rotación de noventa grados,

reflejo en espejo, recorte de una porción significativa, transformación logarítmica y corrección gamma), todas primero manualmente por medio de la implementación de ciclos, y después con comandos de MATLAB, mencionamos que para la operación de corrección gamma tuvimos que usar la imagen muy oscura, y para esta misma y la operación logarítmica normalizamos la imagen antes de realizar las operaciones correspondientes.

En la segunda fase “promedio”, de manera manual obtuvimos el promedio de tres imágenes ruidosas o turbulentas, una vez realizado esto, comparamos este resultado con cada una de las imágenes turbulentas y con una imagen que no presentaba turbulencia, observamos las diferencias o similitudes que pudieran mostrarse en cada una de las respectivas comparaciones.

Por último en la tercera etapa la subdividimos en cuatro puntos a tratar. El primero consistía en obtener manualmente y por medio de comandos de MATLAB, el histograma de una imagen con una capa y ocho bits de intensidad, y observamos la diferencia entre ambos. Nuestro segundo punto consistía en la ecualización del histograma de la imagen, igualmente por medio de comandos de MATLAB y manualmente, nuevamente observamos las diferencias entre ellos. El siguiente punto requería elegir el mejor umbral para binarizar nuestra imagen y observar la zona que se quería resaltar (umbralizar). En el último punto, elegimos un rango de niveles de grises y aumentamos el brillo en ese intervalo, después obtuvimos el nuevo histograma y lo comparamos con el histograma de la imagen original.

III. MÉTODOS Y CÁLCULOS

Para cada etapa de la práctica realizamos un código diferente en MATLAB.

Para la primera fase, el código lo subdividimos dependiendo de la operación a realizar. Primero en MATLAB leímos y obtuvimos el tamaño de la imagen con la que íbamos a trabajar y que cumplía con las características requeridas, hicimos uso de tres matrices, una era la de nuestra imagen original, la segunda era la imagen con comandos y la tercera fue la imagen manual. Teniendo en cuenta esto, explicaremos que fue lo que utilizamos para cada operación.

En la inversión de grises se utilizó una anidación de dos ciclos for, uno controlaba los renglones y el segundo controlaba las columnas, de ambas matrices (la matriz original y la matriz de la imagen manual), ambos tenían como condición de paro las columnas y renglones de la matriz inicial, para ir llenando nuestra tercera matriz, se aplicó la resta de 255 menos el nivel de gris que había en la posición de la matriz original, estipulada por los ciclos for, obviamente

esto se repetía hasta que terminaran los ciclos. El comando utilizado en la segunda matriz fue el de “imcomplement”.

Con respecto a nuestra segunda operación, rotación de 90 grados de nuestra imagen, utilizamos el comando “rot90”, y manualmente el procedimiento fue similar al de la primera operación, sin embargo lo que cambiaba era la operación para llenar nuestra tercera matriz, esta vez realizábamos una resta de renglones.

En la operación de reflejo en espejo, manualmente utilizamos nuevamente una anidación de ciclos for, sin embargo en uno utilizamos inicialmente el tamaño de la segunda matriz obtenida con el comando “fliplr”, el ciclo fue decreciendo de uno en uno y su condición de paro sería hasta llegar a uno, mientras que nuestro segundo ciclo sería igual que los demás utilizados hasta ahora. Hicimos uso de una variable que iría cambiando conforme los ciclos su valor, y este valor se obtendría modificando el valor de la columna; por lo tanto, los renglones y columnas de la matriz original se mantuvieron de acuerdo a lo que estipulaban los ciclos, sin embargo las columnas de la tercera matriz tomaron el valor de nuestra variable, se hizo la asignación del valor de la posición de la primera matriz a la posición de la tercera matriz.

Para realizar el recorte de la imagen (cuarta operación), usamos el comando “imcrop”, el cual requería a diferencia de los demás comandos utilizados hasta ahora, las coordenadas donde inicia, cuanto se va a recorrer y el tamaño de la imagen. Para realizar el recorte manualmente hicimos uso nuevamente de una anidación de ciclos for para el control de renglones y columnas de la tercera y primera matriz, sin embargo la posición de la primera matriz se vio modificada a causa de una sumatoria a las variables de columnas y renglones, se le sumaban los valores de dos variables con nombre “x” y “y”, con las que se trabajaron previamente a los ciclos.

Ya casi para finalizar en la operación de logaritmo no se encontró un comando como tal, por lo tanto trabajamos la tercera y segunda imagen con procedimientos similares, para la segunda imagen primero se dividió entre 255, después a la imagen dos se le signo el resultado de multiplicar 1.9 por el logaritmo de 1 mas el resultado de la división. De manera manual se realizó algo similar, solo que se utilizó la anidación que hasta ahora veníamos manejando de dos ciclos for para el control de renglones y columnas, de la primera matriz, que hizo sustitución de la segunda en las operaciones mencionadas anteriormente, y de la tercera matriz.

Por último en la corrección gamma, cambiamos la imagen, utilizamos una imagen a color y con ayuda de la función “rgb2gray”, la pasamos a escala de grises, no encontramos un comando directo que realizara la operación, entonces nuevamente se trabajaron las imágenes dos y tres con un método similar, lo único que varío entre una y otra, fue que en la imagen dos se aplicó directo mientras que en la imagen tres se tuvo que utilizar la anidación de ciclos y se sustituyó la imagen dos por la imagen original. Cabe mencionar que tanto en esta operación como en la logarítmica para la normalización de la imagen se dividió entre 255.

Una vez terminada la primera etapa, nos dirigimos a trabajar en la segunda parte, la cual consistía en obtener un promedio de tres imágenes turbulentas, mostrar esa imagen resultante y compararla con las tres previas y una imagen sin turbulencia, para realizar esto, primero en el programa de MATLAB

leímos las tres imágenes con turbulencia que ya nos proporcionaban, las transformamos a matrices, utilizamos una variable de nombre “res”, en la cual almacenaríamos primero la sumatoria de las tres matrices, y luego la división de esta sumatoria entre tres, una vez hecho esto la pasamos nuevamente a una imagen que pudiéramos visualizar, realizamos las comparaciones correspondientes y fue todo.

Para nuestra última parte la trabajamos como cuatro puntos diferentes, todos con la misma imagen, en el primer punto obtuvimos dos histogramas de dicha imagen, uno con el comando “imhist”, y otro con una anidación de ciclos for, que iban controlando la posición de la matriz de la imagen, la cual proporcionaba un valor que utilizaríamos en la operación dentro del ciclo for. El segundo punto implementamos una ecualización al histograma obtenido en el punto previo, por medio del comando “histeq” y manualmente con una anidación de ciclos for, además del uso de (1). Para el tercer punto elegimos como umbral en nuestro histograma, para ello, optamos por el valor más alto y el más bajo, y los promediamos, y esto tomamos como valor umbral, a partir de este valor con una anidación de ciclos for para ir controlando columnas y renglones, fuimos comparando posición por posición de la matriz con el valor promedio, dependiendo si era más bajo o más alto se asignaba a esa posición el valor de 255, 0 o nuestro valor promedio, de esta manera logramos binarizar nuestro histograma a partir del umbral elegido. Para el último punto de esta parte, elegimos a partir de nuestro umbral previo un rango de niveles de grises para resaltarlos, con resaltarlos nos referimos a aumentar el nivel de gris en 90 mas, primeramente con niveles mayores a el nivel umbral (70 más), después saltando niveles (uno si, uno no), y por ultimo tomando niveles arriba (70 más) y niveles abajo (30 menos) del valor umbral de nuestra elección. Para esto utilizamos igualmente anidaciones de ciclos for y algunas evaluaciones para ir aumentando en el nivel deseado.

IV.

RESULTADOS

A. Operaciones Básicas

La Fig. 1 muestra la imagen que utilizamos para realizar casi todas las operaciones requeridas en el reporte. El resto de las imágenes hasta la Fig. 14, exceptuando la Fig. 12, muestran los resultados de dichas operaciones por ambos métodos (comandos y manual).



Fig. 1. Imagen de una resonancia magnética de cabeza, con una capa y ocho bits de intensidad.



Fig. 3. Imagen resultante de la operación "Inversión de Grises", obtenida con comandos.



Fig. 2. Imagen resultante de la operación "Inversión de Grises", obtenida manualmente.

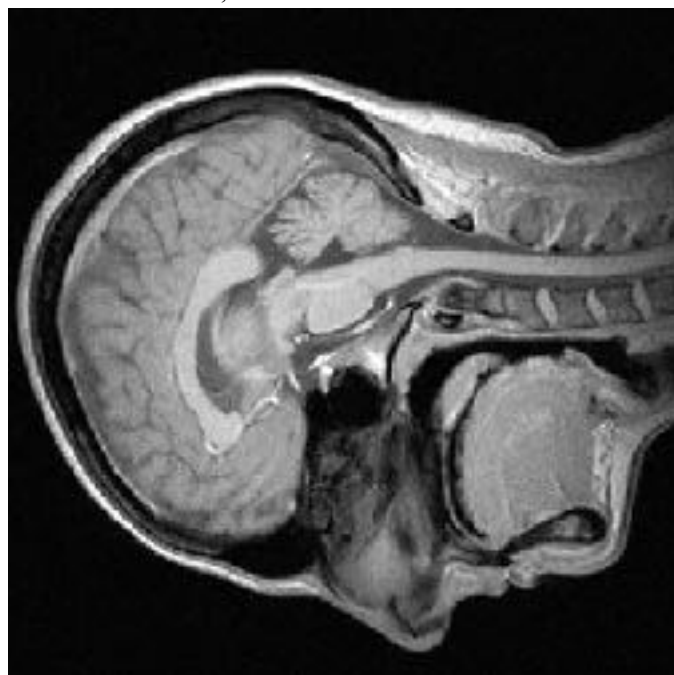


Fig. 4. Imagen resultante de la operación "Rotación 90 Grados" hacia la derecha, obtenida manualmente.

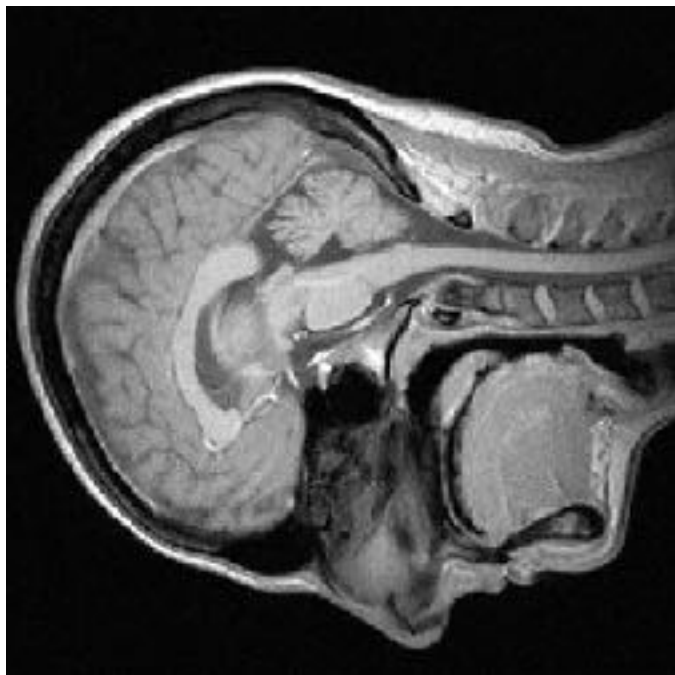


Fig. 5. Imagen resultante de la operación “Rotación 90 Grados” hacia la derecha, obtenida con comandos.



Fig. 7. Imagen resultante de la operación “Reflejo en Espejo”, obtenida con comandos.



Fig. 6. Imagen resultante de la operación “Reflejo en Espejo”, obtenida manualmente.



Fig. 8. Imagen resultante de la operación “Recorte”, obtenida manualmente.



Fig. 9. Imagen resultante de la operación “Recorte”, obtenida con comandos.



Fig. 10. Imagen resultante de la operación “Transformación Logarítmica”, obtenida manualmente.



Fig. 11. Imagen resultante de la operación “Transformación Logarítmica”, obtenida aplicando el proceso directo a la segunda matriz.



Fig. 12. Imagen a color utilizada para la corrección gamma.



Fig. 13. Imagen resultante de la operación “Corrección Gamma”, obtenida manualmente.



Fig. 14. Imagen resultante de la operación "Corrección Gamma", obtenida aplicando el proceso directo a la segunda matriz.

B.Promedio

Al realizar las respectivas comparaciones entre las imágenes 15, 16, 17, y 18, con la Fig. 19, se nota lo siguiente:

En la Fig. 15 definitivamente el promedio no ayuda en mejorar la calidad de la imagen, al contrario, la empeoraría, ya que ante la percepción del ojo humano, se ve más borrosa.

En la Fig. 16, el efecto que tiene el promedio no es en mayor medida, pero ayuda ligeramente en la mejora de la resolución de la imagen ante la percepción de la vista humana.

En la Fig. 17 tiene una efectividad mucho mayor a comparación de las dos imágenes previas, se observa mucho más clara, pudiendo percibir algunos detalles que son irreconocibles en dicha Fig.

Por último en la Fig. 18 por lógica se obtuvo un efecto negativo, el promedio no es una opción para mejorar la resolución de la imagen, todo lo contrario sería una mala decisión aplicar el promedio en esta imagen, ya que la claridad se pierde en gran medida, haciéndola de mala calidad.



Fig. 15. Primera imagen con presencia de turbulencia.



Fig. 16. Segunda imagen con presencia de turbulencia.



Fig. 17. Tercera imagen con presencia de turbulencia.



Fig. 19. Imagen resultante del promedio de las tres imágenes con presencia de turbulencia.



Fig. 18. Imagen sin presencia de turbulencia.

C.Histograma

Podemos observar en los primeros dos histogramas, que realmente hay una diferencia entre lo que se obtuvo con un comando que realiza el procedimiento de manera directa y sin errores, y uno obtenido de manera manual a través de ciclos y operaciones.

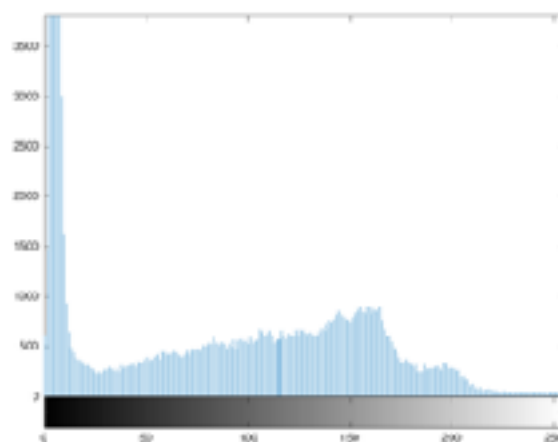


Fig. 20. Histograma obtenido por medio de comandos.

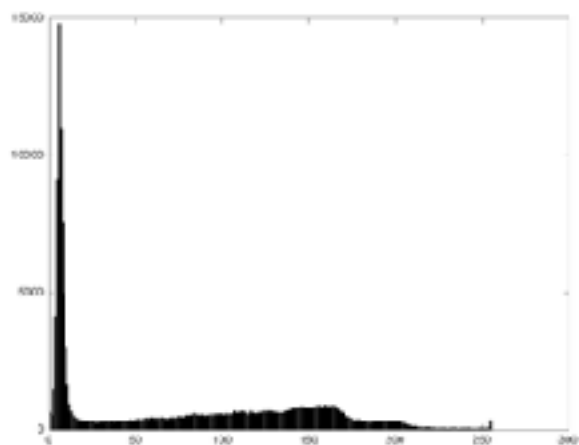


Fig. 21. Histograma obtenido de manera manual.

Los siguientes dos histogramas nos muestran la ecualización del histograma de la Fig. 20, igualmente por medio de comandos y manualmente, podemos observar una notable diferencia entre ambos.

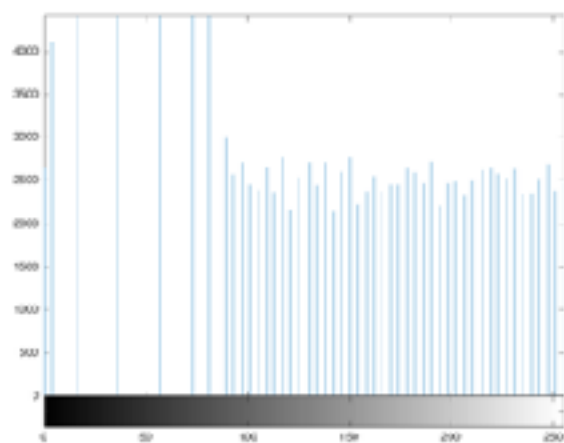


Fig. 22. Histograma ecualizado por medio de comandos.

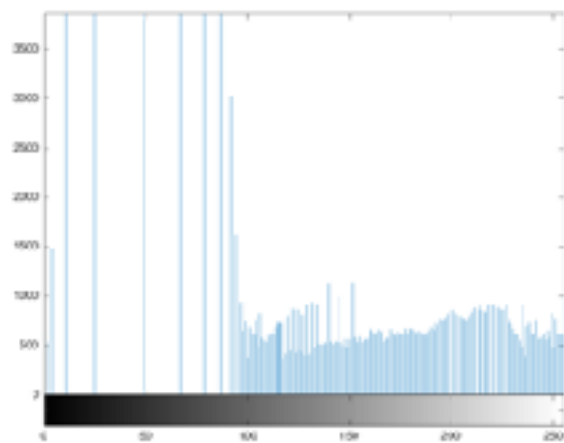


Fig. 23. Histograma ecualizado manualmente.

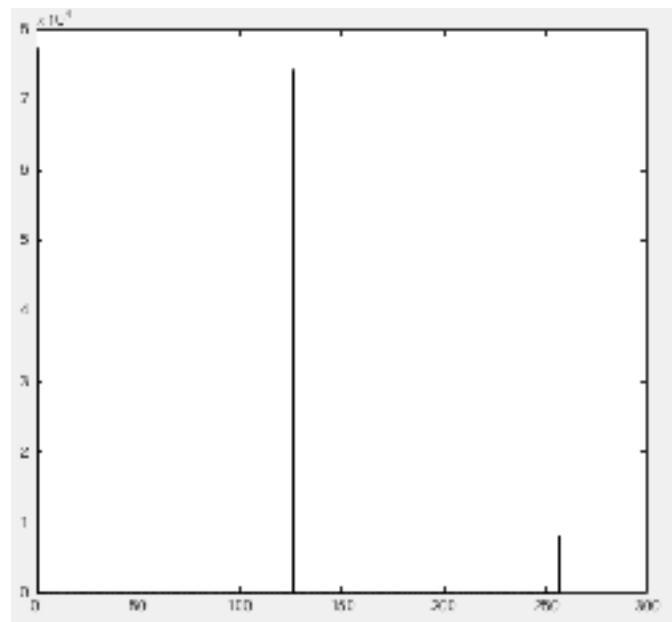


Fig. 24. Histograma binarizado, a partir de un valor umbral (línea de en medio).

En los siguientes histogramas observamos lo que le sucede a la imagen cuando se le da cierto realce determinados valores, a partir de un valor umbral tanto arriba como debajo de este, además de las modificaciones que sufre el histograma con esto.

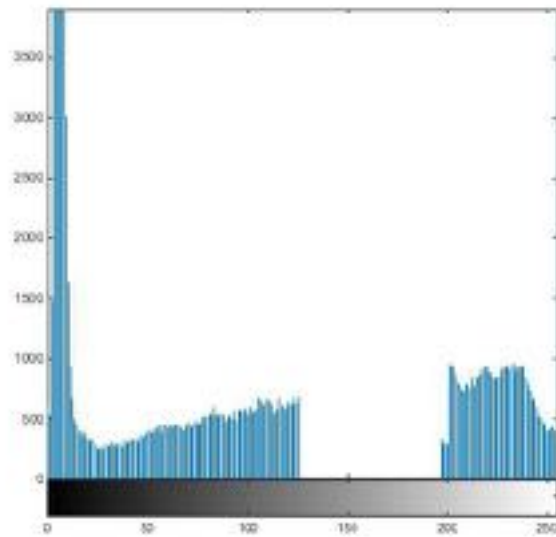


Fig. 25. Histograma con realce en un rango de niveles de grises, tomado a partir de nuestro valor umbral y finalizado 70 niveles arriba de este valor.

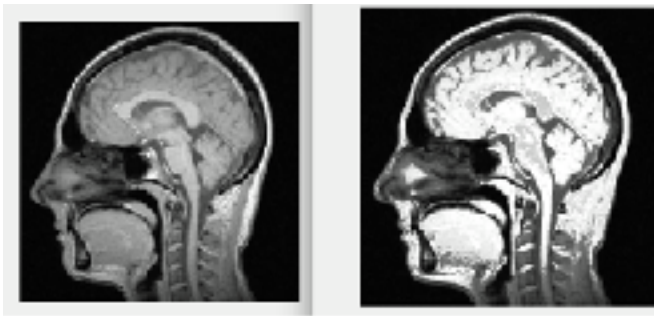


Fig. 26. Imagen resultante (izquierda) del histograma de la Fig. 25, en comparación con la imagen original (derecha).

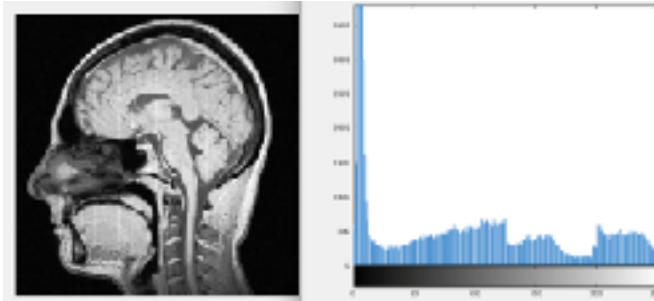


Fig. 27. Imagen resultante (derecha) al saltar uno nivel en el rango seleccionado, y su respectivo histograma (izquierda).

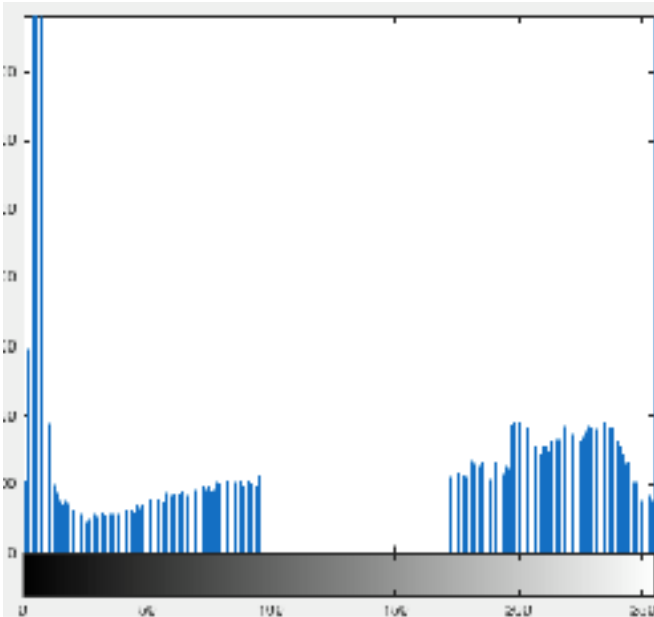


Fig. 28. Histograma resultante tomando el rango de 30 niveles abajo del valor umbral y 70 niveles arriba del valor umbral, igualmente dando realce a todos los valores del rango.

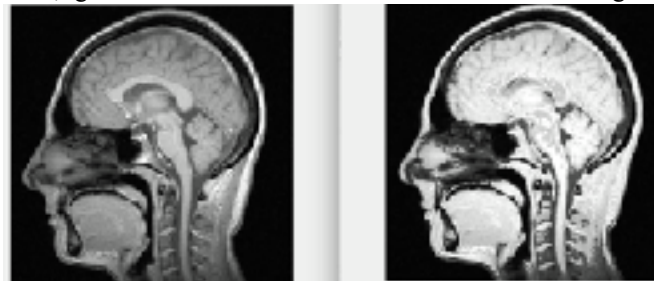


Fig. 29. Imagen resultante (izquierda) del histograma de la Fig. 28, en comparación con la imagen original (derecha).

V.

DISCUSIÓN Y CONCLUSIONES

Para esta práctica tuvimos más complicaciones que practicas pasadas. En nuestra primera fase tuvimos complicaciones en las operaciones, logarítmica y corrección gamma, en la primera principalmente en la aplicación del logaritmo, no lográbamos hacerla adecuadamente y por lo tanto el resultado no era el esperado, y en ambas cometimos errores de sintaxis, el descuido de no manejar matrices dobles y por supuesto olvidar el normalizar la imagen, cabe mencionar que fueron las únicas operaciones que no pudimos encontrar un comando directo para ellas, incluso podemos decir que no existe tal comando o prácticamente nuestra investigación acerca de ellos no fue la mas completa. Para el resto de las operaciones no tuvimos mayor inconveniente.

De las tres fases podríamos decir que la segunda fue la más sencilla para nosotros, y no tuvimos problema alguno.

Por otra parte la etapa tres fue la más complicada de todas, especialmente cuando tuvimos que ecualizar el histograma, por un momento no tuvimos ni idea de cómo hacerlo, por lo tanto nos vimos en la necesidad de investigar una posible manera de logarlo, descubriendo la ecuación vista en clase, que realmente fue de ayuda, nuestro segundo punto que se volvió problemático fue el umbralizar el histograma, ya que para concluir que nuestro umbral seria el promedio del valor más alto y más bajo, tuvimos que hacer pruebas previas con valores aleatorios.

Con respecto a los resultados que obtuvimos podemos decir que en la primera parte, la diferencia entre lo manual y lo hecho con comandos, es prácticamente nula a simple vista entre ambas imágenes, con ambos métodos logramos llegar al mismo resultado, sin embargo en cuestión de código, podemos decir que el uso de comando facilita mucho las cosas y reduce en gran medida el código.

Nuestra segunda etapa era más que obvio que nuestra imagen sin turbulencia seguiría siendo mejor que nuestra imagen promedio, por lo tanto podemos decir que en esta comparación no hubo mejoría alguna, sino todo lo contrario. Respecto a las imágenes con turbulencia, realmente se esperaba que se notara una mejoría en ellas, sin embargo fue sorpresivo el observar que solo en una de ellas hubo una mejoría notable, por supuesto no de alta calidad pero si al menos que el ojo pudiera distinguir ciertos detalles que en la imagen original no. En cuanto a las dos, fue decepcionante el ver que el promedio no las ayudaría en nada, si a caso a una peor muy ligeramente (prácticamente nulo).

Por último de la etapa final nos demostró diferencias que podemos encontrar entre usar comandos y hacerlo de manera manual. En esta parte notamos en los histogramas una notable diferencia entre ambos procedimientos, dándonos cuenta que hay mucha mayor precisión y exactitud en un comando que un método manual. Además el último punto nos muestra tanto en el histograma como en la imagen, las diferencias que hay cuando se realza el brillo en determinados puntos, más allá de nuestro valor umbral, tanto arriba como debajo de este valor umbral y cuando vas saltando niveles por arriba del valor umbral.

Apéndice A

Operaciones Básicas

```
function [ I, I2, I3 ] = Comandos( input_args )
%Comandos y ciclos en Imagenes en escala de grise
% Este programa prueba comandos para modificar imagenes en escala de
% grises y luego se realiza la misma accion utilizando ciclos recorriendo
% pixel por pixel la imagen. Se compara el resultado con cada accion.

%Leemos la imagen de Resonancia Magnetica

I=imread('RMI2.jpg');
%I2 sera nuestra matriz destinada a comandos, en este caso hacemos una
%inversion de colores en escala de grises
I2=imcomplement(I);
%I3 sera nuestra matriz destinada a ciclos, creamos la matriz llena de
%seros del tama?o de nuestra matriz original.
I3=zeros(size(I));

%Recorremos toda la matriz
for j=1:size(I,1);
    for i=1:size(I,2);
        % Al hacer la resta de 255 menos el valor de la posici?n podemos
        % asegurarnos de obtener el valor contrario al que tenemos, transformamos 0
        % en 255 y viceversa, igualmente con valores intermedios.
        I3(i,j)=255-I(i,j);
    end
end
%Se transforma la matriz a un formato de 8 bits.
I3=uint8(I3);

%Se guardan las imagenes obtenidas

imwrite(I2,'negativo_comando.jpg');
imwrite(I3,'negativo_ciclo.jpg');

%Rotacion 90 grados

%Se utiliza el comando de rotacion de 90 grados para girar la imagen y
%guardarla en una nueva matriz
I2=rot90(I);

for j=1:size(I,1);
    for i=1:size(I,2);
        %Las columnas pasan a ser los renglones de nuestra matriz y los renglones
        %tienen que ser las columnas pero en sentido inverso, as? que tenemos que
        %restar el tama?o de nuestra matriz menos nuestra variable i que va
        %cambiando en nuesro ciclo for. le aumentamos uno a este valor para
        %permitir que nuestra matriz vaya desde el 400 a 1, o desde nuestro valor
        %limite a 1.
        I3(i,j)=I(j,size(I,1)-i+1);
    end
end

%Se guardan las imagenes
```

```

imwrite(I2,'rotacion_comando.jpg');
imwrite(I3,'rotacion_ciclo.jpg');

%reflejo Espejo

%Utilizamos el comando para crear una imagen reflejo de nuestra imagen
%original
I2=fliplr(I);

for j=size(I,2):-1:1;
    for i=1:size(I,1);
        %creamos un valor auxiliar que vaya desde nuestro valor limite a 1.
        a=size(I,2)-j+1;
        %Este valor nos permitira guardar la primera columna de la imagen original
        %en la ultima columna de nuestra nueva imagen.
        I3(i,a)=I(i,j);
    end
end

%Guardamos nuestras imagenes

imwrite(I2,'reflejo_comando.jpg');
imwrite(I3,'reflejo_ciclo.jpg');

%Recorte

%utilizamos el comando de imcrop. Utilizando herramientas de imagen
%obtenemos las coordenadas de inicio y fin de nuestra imagen, inicio en
%(248,153) y el final en (325,245). El comando imcrop funciona con
%coordenadas como primeros parametros y tamaños como segundo parametro, es
%por eso que usamos una resta para obtener este tamaño en los segundos
%parametros
I2=imcrop(I, [248, 153, 325-248, 245-153]);
%creamos nuestras variables con estos valores
x=248; y=153;
sizeX=325-248; sizeY=245-153;
%Inicializamos nuestra matriz con el tamaño que tendrá nuestro recorte
I3=zeros(sizeY,sizeX);
%le damos formato de 8 bits
I3=uint8(I3);

for j=1:sizeX;
    for i=1:sizeY;
        %Aquí vamos recorriendo la matriz para copiar los valores de nuestro
        %recorte en nuestra nueva matriz
        I3(i,j)=I(y+i,x+j);
    end
end

%Guardamos nuestras imagenes recortadas.

imwrite(I2,'recorte_comando.jpg');
imwrite(I3,'recorte_ciclo.jpg');

%Logaritmo
%Devolvemos nuestras matrices al tamaño original
I3=zeros(size(I));
%Le damos formato doble a nuestra matriz

```

```

I2=double(I);
%Normalizamos nuestros valores
I2=I2/255;
%Aplicamos el logaritmo en toda la matriz
I2=1.9*log(1+I2);
%La devolvemos a valores de 8 bits
I2=im2uint8(I2);

for j=1:400;
    for i=1:400;
        %Para el ciclo utilizamos el mismo concepto. Un valor en formato doble de
        %nuestra matriz original dividido entre 255 y le aplicamos el logaritmo a
        %este valor para finalmente guardarlo en la matriz original.
        I3(i,j)=1.9*log(1+(double ( I(i,j) ) ) /255 );
    end
end

%guardamos nuestras imagenes
imwrite(I2,'logaritmo_comando.jpg');
imwrite(I3,'logaritmop_ciclo.jpg');

%Correccion Gamma

%Leemos nuestra nueva imagen oscura que utilizaremos para la correccion
%gamma
I=imread('pp.jpeg');
%Al ser una imagen a color la transformaremos a escala de grises
I2=rgb2gray(I);
%le damos formato doble
I2=double(I2);
%aplicamos nuestra correccion gamma
I2=255*((I2/255).^(1/1.5));
%le damos formato de 8 bits
I2=uint8(I2);
%creamos la matriz de ciclos con el tamaño de nuestra nueva imagen
I3=zeros(size(I));

for j=1:size(I,1);
    for i=1:size(I,2);
        %creamos nuestra variable auxiliar para facilitar la aplicacion de la
        %correccion gamma
        a=1/1.5;
        %creamos otra variable auxiliar que albergue el valor en formato doble de
        %nuestro pixel actual
        aux=double(I(i,j));
        %Guardamos el valor de nuestra correccion gamma en nuestra matriz
        I3(i,j)=255*((aux/255).^(a));
    end
end
%Damos el formato de 8 bits a nuestra matriz
I3=uint8(I3);

%guardamos nuestras imagenes

imwrite(I2,'gamma_comando.jpg');
imwrite(I3,'gamma_ciclo.jpg');

```

end

Apéndice B

Promedio

```
function [ output_args ] = Promedio( input_args )
%Promedio de una imagen
% Programa que lee 4 imagenes especificas, una sin turbulencias y 3 con
% distintos niveles de turbulencia. Sacaremos promedio de nuestras 3
% imagenes turbulentas para comparar la eficacia de un promedio entre
% estas para disminuir esta turbulencia.

%Leemos nuestras 4 imagenes, una sin turbulencias y otras 3 con distintos
%niveles de turbulencia
I=imread('aerial_view_no_turb.tif');
I2=imread('aerial_view_turb_c1.tif');
I3=imread('aerial_view_turb_C2.tif');
I4=imread('aerial_view_turb_c3.tif');
%Les damos un formato doble a nuestras imagenes a modificar
I2=double(I2);
I3=double(I3);
I4=double(I4);
%Sumamos nuestras tres imagenes en una nueva matriz
res=I2+I3+I4;
%Dividimos este resultado entre 3 para obtener un promedio de nuestras
%turbulencias
res=res/3;
%Devolvemos un formato de 8 bits
```



```

res=uint8(res);
%Guardamos nuestra imagen
imwrite(res,'promedio.jpg','jpg');
%Mostramos nuestras dos imagenes para compararlas
figure(2), imshow(res);
figure(1), imshow(I);

end

```

Apéndice C

Histograma

```

function [ I2, prom ] = Histograma( input_args )
% Operaciones con Histogramas
% Este programa prueba comandos de histogramas como ecualizacion a la vez
% que prueba una implementacion de operaciones mediante ciclos de los
% mismos comandos.

%Leemos nuestra imagen
I=imread('RMI2.jpg');
%creamos un vector para la frecuencia de cada valor de nuestra matriz
frec=zeros(1,256);
%Obtenemos el tamaño de nuestra matriz en dos variables.
[fil, col]=size(I);

%Histograma manual

%Recorremos nuestra matriz para obtener la repetición de cada uno de los
%valores de gris en nuestro vector.
for j=1:400;
    for i=1:400;
        a=I(i,j);
        frec(1,a+1)=frec(a+1)+1;
    end
end

%Creamos nuestro vector de valores en x
x=1:256;

%Enseñamos y guardamos nuestro histograma manual utilizando el comando bar
%para crear una gráfica de barras dependiendo de la frecuencia de cada
%valor.

```

```

figure(1), bar(x,frec);
print('histograma_manual','-dpng');

%Mostramos el histograma normal utilizando imhist

figure(2), imhist(I);
print('histograma_comando','-dpng');

%Damos un formato doble a nuestra matriz
I2=double(I);

%Ecuacion

%Para la ecualizacion necesitaremos un suma de nuestros valores, un vector
%de ecualizacion y un vector de la frecuencia ademas de tener el numero del
%total de nuestros pixeles.
suma=zeros(1,256);
ecual=suma;
frec1=suma;
totalpix=fil*col;

%Lo primero por hacer es obtener la frecuencia relativa de nuestros niveles
%de gris de nuestra imagen.
for k=1:256

    frec1(k)=frec(k)/totalpix;

    if k==1;
        suma(k)=frec1(k);
    end
    %Al mismo tiempo iremos obteniendo la suma de todos estos valores en una
    %suma secuencial
    if k>1;
        suma(k)=suma(k-1) + frec1(k);
    end

    %Despues conseguiremos el valor de ecualizacion de para cada nivel de gris
    ecual(k) = suma(k)*256;

end

%Creamos una matriz que albergara nuestra ecualizacion
Imgeq=zeros(fil,col);

%Albergaremos en esta nueva matriz el valor de nuestra ecualizacion
for i=1:fil

    for j=1:col

        %Nuestra variable K tomar? el valor del pixel en nuestra matriz en formato
        %doble
        k = I2(i,j);
        %Se asigna el valor de ecualizacion del nivel de gris del pixel original a
        %nuestra nueva matriz
        Imgeq(i,j)=ecual(k+1);

    end

end
end

```

```
%Guardamos nuestra matriz en formato de 8 bits
Imgeq = uint8(Imgeq);
%Utilizamos un comando de ecualizacion para modificar nuestra imagen
%original y lo guardamos en otra matriz
ImOEq = histeq(I);

%Comparamos ambas imagenes para ver los resultados.

figure(1), imhist(Imgeq);
print('ecual_manual','-dpng');
figure(2), imhist(ImOEq);
print('ecual_comando','-dpng');

%Binarizar imagen

%Creamos variables para obtener nuestra frecuencia mayor y menor. Esto es
%para obtener el valor del gris de un valle y una cresta.
max=0;
min=totalpix;

%recorremos nuestro vector de frecuencias para encontrar nuestro valor
%maximo y minimo, el minimo tendra que ser un numero distinto a 0.
for k=1:256

    if max<frec(k);
        max=frec(k);
    end

    if min>frec(k)&&frec(k)~=0;
        min=frec(k);
    end

end

%Obtenemos la posicion de cada valor para obtener el valor del gris que le
%corresponde, se le resta uno porque nuestros valores van de 1 a 256
posmax=find(frec==max)-1;
posmin=find(frec==min)-1;

%Obtenemos el valor promedio entre estos valores para conseguir un valor
%intermedio entre la cresta y el valle.
prom=(posmax+posmin)/2;
%Se redondea este valor
prom=round(prom);

%Reiniciamos nuestras matrices
I2=zeros(fil,col);
I=uint8(I);

%Recorremos nuestra matriz
for i=1:fil
    for j=1:col

        %Si nuestro valor esta por encima del promedio este tomara un valor de 255
        if I(i,j)>prom;
            I2(i,j)=255;
        else
            %Si en cambio esta por debajo o es igual tomara un valor de 0
```

```

        if I(i,j)<=prom;
            I2(i,j)=0;
        end
    end

end

end

%Devolvemos el formato a nuestra matriz
I2=uint8(I2);

%Guardamos nuestra imagen binaria
imwrite(I2,'Imagen_binaria.jpg','jpg')

%Obtenemos otros dos limites para poder crear nuestro histograma resaltando
%nuestro valor umbral
prominf=round(prom/2);
promsup=round(prom/2+prom);

y=frec;

%Los siguientes For nos permiten obtener una matriz llena de 0, prom y 255.
%Esto nos permitira crear un histograma que nos permita resaltar nuestro
%valor umbral para la binarizacion.

for k=prominf:-1:2;

    y(1)=y(1)+y(k);
    y(k)=0;

end

for k=prominf:prom-1;

    y(prom)=y(prom)+y(k);
    y(k)=0;

end

for k=promsup:-1:prom+1;

    y(prom)=y(prom)+y(k);
    y(k)=0;

end

for k=promsup:255;

    y(256)=y(256)+y(k);
    y(k)=0;

end

end

%Histograma con umbral

figure(1), bar(x,y);
print('histograma_umbral','-dpng');

%Niveles de grises
I2=I;

```

```
%A partir de nuestro promedio tomamos 30 valores inferiores y 70
%superiores para este caso específico. Este rango de valores ser?n
%resaltados (multiplicados por 1.5) para aumentar estos niveles de gris.
```

```
for k=prom-30:prom+70;
    for i=1:size(I,1);
        for j=1:size(I,2);
            if I(i,j)==k;
                I2(i,j)=I(i,j)*1.5;
            end
        end
    end
end
```

```
%Guardamos nuestra imagen y nuestro histograma de la imagen
```

```
imwrite(I2,'Resaltar_grises.jpg','jpg');
figure(1), imhist(I2);
print('Histograma_Resalt','-dpng');

end
```