**P1\_RGB\_BitDepth\_ECAV.m**

% Cierra todas las ventanas y variables

close all;

% Selecciona el nombre del fichero

FileName = '1-Face.png';

% Lee la imagen

RGB=imread(FileName);

% Obten la información de la iamgen

whos RGB;

% Obten los parametros de la imagen

[Rows,Cols,C]=size(RGB);

% Presenta en pantalla la resolución de la imagen utilizando fprintf del modo

% Filas:xxxx, Columnas = xxx

fprintf('Filas: %0i, Columnas: %0i\n', Rows, Cols)

% Visuliza la imagen RGB

figure; imshow(RGB); title('RGB original');

% Visualiza cada una de las componente de la imagen con imshow() en

% ventanas independientes

Red=RGB(:,:,1);

figure; imshow(Red),title('Red');

Green=RGB(:,:,2);

figure; imshow(Green),title('Green');

Blue=RGB(:,:,3);

figure; imshow(Blue), title('Blue');

% Selecciona un area de la imagen con gran detalle y almacenala en la variable RGB\_ROI

RGB\_ROI = RGB(500:1220,500:1260,:);

% Visualiza la RGB\_ROI con imshow(RGB\_ROI);

figure; imshow( RGB\_ROI),('Cut');

% Cuentifica la imagen con N bits, N=7,6,5,4,3,2, y 1; y almacenalo en una

% variable RGB\_Nbits

for N=7:-1:1;

b=8-N;

RGB\_Nbits= uint8(floor(double(RGB\_ROI)/2^b));

% Visualiza la imagen cuantificada utiliando imshow(RGB\_Nbits\*(2^b));

figure, imshow(RGB\_Nbits\*(2^b)), title('Cuantificando');

% Escribe la imagen para cada una de las cuantificaciones en un fichero

% 'Nombre\_imagen\_Nbits.tiff', utilizando el comando imwrite() y

% utilizando los campos, 'compression','none';

text = sprintf('%d.tiff',N)

imwrite(RGB\_Nbits\*(2^b), text);

% Calcula la relación de compresión que se obtiene para valor de

% cuantificacion

Compresion = 8 /N;

% Muestra por pantalla la relación de compresión obtenida para cada

% valor de cuantificación utilizando el comando fprintf()

fprintf('Relacion de compresión %0i: %0f\n',N,Compresion)

end

**P1\_YCbCr\_Subsampling\_ECAV.m**

% Cierra todas las ventanas y variables

close all;

% Selecciona el nombre del fichero

FileName = '7-Cartoons.jpg';

% Lee la imagen

RGB=imread(FileName);

% Obten los parametros de la imagen

[Rows,Cols,C]=size(RGB);

% Visuliza la imagen RGB

figure; imshow(RGB); title('RGB original');

% Cambia la imagen al espacio de color YCbCr

YCbCr=rgb2ycbcr(RGB);

% Visualiza cada una de las componente Y, Cb y Cr de la imagen con imshow() en

% ventanas independientes

Y =YCbCr(:,:,1);

figure; imshow(Y),title('Componente Y');

Cb =YCbCr(:,:,2);

figure; imshow(Cb),title('Componente Cb');

Cr =YCbCr(:,:,3);

figure; imshow(Cr),title('Componente Cr');

% Guarda las componentes Cb y Cr en una variable independiente

Cb=YCbCr(:,:,2);

Cr =YCbCr(:,:,3);

% Escala la ocmponente Cb en potencias de dos 1/2, 1/4, 1/8,... 1/1024,

% para ambas dimensiones x,y, utilizando la función imresize()

for XY=1:10

% --------------- CB ------------------------------------------------

Cb4XY= imresize(YCbCr(:,:,2),1/(2^XY));

% Visualia la componente Cb

tituloCb = sprintf('Cb4XY Submuestreado: 1/%d', 2^XY );

figure; imshow(Cb4XY); title(tituloCb);

% Escala la componente Cb a su resolución original

Cb444= imresize(Cb4XY,2^XY);

% Visualia nuevamente la componente Cb re-escalada

tituloCb = sprintf('Cb444 escalado: %d', 2^XY );

figure; imshow(Cb444); title(tituloCb);

% copia la componente Cb escalada en la variable YCbCr

% Como el escalado puede generar una imagen de mayor resolución que la

% imagen original asegurate de recortarla a RowsxCols

YCbCr(:,:,2)=Cb444(1:Rows,1:Cols);

% --------------- CR ------------------------------------------------

% Repite los mismos pasos para la componente Cr

Cr4XY= imresize(YCbCr(:,:,3),1/(2^XY));

% Visualia la componente Cb

tituloCr = sprintf('Cr4XY Submuestreado: 1/%d', 2^XY );

figure; imshow(Cr4XY); title(tituloCr);

% Escala la componente Cb a su resolución original

Cr444= imresize(Cr4XY,2^XY);

% Visualia nuevamente la componente Cb re-escalada

tituloCr = sprintf('Cr444 escalado: %d', 2^XY );

figure; imshow(Cr444); title(tituloCr);

% copia la componente Cb escalada en la variable YCbCr

% Como el escalado puede generar una imagen de mayor resolución que la

% imagen original asegurate de recortarla a RowsxCols

YCbCr(:,:,3)=Cr444(1:Rows,1:Cols);

% Convierte la imagen nuevamente al espacio RGB con ycbcr2rgb

RGB\_Conv = ycbcr2rgb(YCbCr);

% Visualiza la nueva imagen RGB\_Escalada

tituloRGB = sprintf('RGB submuestreo croma: 1/%d', 2^XY );

figure; imshow(RGB\_Conv); title(tituloRGB);

% Calcula la relación de compresión obtenida para cada factor de

% submuestreo XY, e imprime el resultado con fprintf()

Compresion = 8 /XY;

fprintf('Relacion de compresión %0i: %0f\n',XY,Compresion)

end