

UNIVERSIDAD INDUSTRIAL DE SANTANDER.
PABLO JOSUE ROJAS YEPES.
PROCESAMIENTO DE IMÁGENES DIGITALES.
LABORATORIO 2: IMÁGENES.

1. Objetivo

Comprender los conceptos de Cuantificación, y submuestreo. Y el ruido asociado a estos conceptos.

2. Cuantificación

Cuantificación se puede definir como un mapeo de una señal de un dominio en el espacio R a un espacio N. Lo que provoca que la diferencia entre 2 niveles cuantificados sea un paso de tamaño definido (quant). La relación del quant y el número de bits que se necesitan para la cuantificación es:

$$q = \frac{A}{2^B - 1}$$

Donde q hace referencia al quant y B al número de bits.

2.1. Problemas:

- a. ¿Qué hace la función de MATLAB quant()? Como podemos usarla para uniformemente cuantificar una imagen con N niveles?}

quant():

Discretiza los valores como múltiplos del quant.

Sintaxis:

quant (X, Q)

Descripción:

quant (X, Q) toma dos entradas:

X...Matriz, vector o escalar



Q...Valor mínimo

y devuelve los valores de X redondean al múltiplo más próximo de Q.

Ejp:

<pre>x = [1.333 4.756 -3.897]; y = quant(x,0.1) y = 1.3000 4.8000 -3.9000</pre>	<pre>x = [1.333 4.756 -3.897]; y = quant(x,0.001) y = 1.3330 4.7560 -3.8970</pre>
--	--

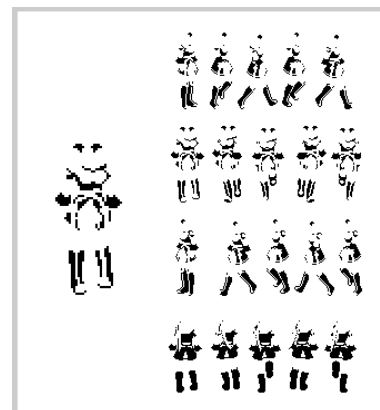
- b. Cree una función para cuantificar una imagen I, de tal manera que el rango dinámico de la imagen sea cuantificado en N niveles uniformes. Donde N es un parámetro de entrada de su función (al igual que la imagen I).

 <p><i>Img 1. Sin cuantificar.</i></p>	 <p><i>Img 2. Cuantificada.</i></p>
<pre> 1 function [Ic]=cuantificar(I,N) 2 % I = Imagen original 3 % N = Niveles. 4 5 Ic = round(I*N) ./ (N-1); 6 end </pre>	<pre> 1 clc; clear all; close all; 2 3 I=rgb2gray(im2double(imread('sprite.tif'))); 4 figure,imshow(I); 5 B=3; %Cantidad de bits 6 N=2.*B; 7 [Ic] = cuantificar(I,N); 8 figure,imshow(Ic); </pre>

- c. Use una imagen en escala de grises. Muestre dicha imagen y las imágenes cuantificadas usando desde 1 hasta 8 bits.



Img 3. Sin cuantificar.



Img 4. Cuantificada a 1 bit.



Img 5. Sin cuantificar.



Img 6. Cuantificada a 2 bits.



Img 7. Sin cuantificar.



Img 8. Cuantificada a 3 bits.



Img 9. Sin cuantificar.



Img 10. Cuantificada a 4 bits.



Img 11. Sin cuantificar.



Img 12. Cuantificada a 5 bits.



Img 13. Sin cuantificar.



Img 14. Cuantificada a 6 bits.



Img 15. Sin cuantificar.



Img 16. Cuantificada a 7 bits.



Img 17. Sin cuantificar.



Img 18. Cuantificada a 8 bits.

d. ¿Cuántos bits se necesitaron para que note la diferencia?

Entre menos bits se usen más se va a ver las diferencias.

e. ¿Qué es el ruido de cuantificación?, ¿Cuál es el máximo valor posible de ruido de cuantificación para los ejemplos anteriores?.

Ruido de cuantificación: Señal en tiempo discreto y amplitud continua introducida por el proceso de cuantificación, y que resulta de igualar los niveles de las muestras de amplitud continua a los niveles de cuantificación más próximos.

Para 1 bit	1.2471	Para 5 bit	0.1359
Para 2 bit	0.3492	Para 6 bit	0.1068
Para 3 bit	0.2471	Para 7 bit	0.0932
Para 4 bit	0.1690	Para 8 bit	0.0826

3. Muestreo

El muestreo es un proceso mediante el cual se crea una señal discreta a partir de una continua, de tal forma que los valores (muestras) son tomados en ciertas posiciones. Si las muestras se toman con el mismo espacio, el muestreo se llama uniforme. Dado que los computadores no pueden almacenar señales continuas. Trabajaremos con señales densamente muestreadas.

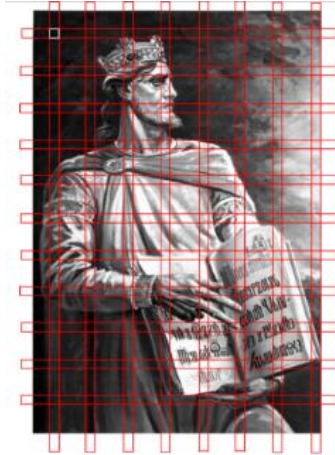


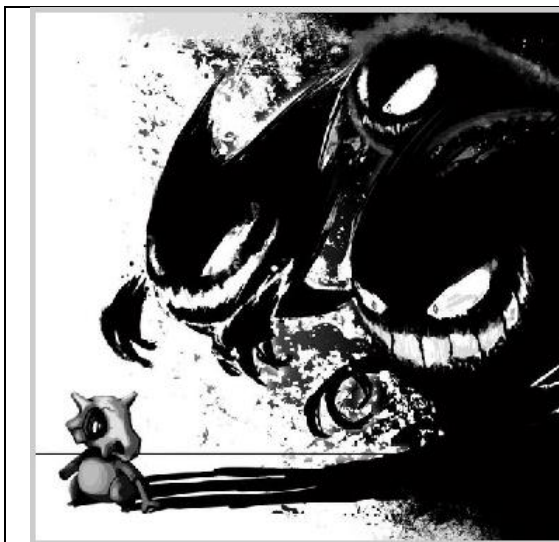
Figura 1: El muestreo toma únicamente los valores de los píxeles que se interceptan en las líneas verticales y horizontales.

3.1. Ejemplo

```
>> [img,map]=imread('coins.png');  
>> coins =img(1:10:end,1:10:end,:);
```

3.2. Problemas

- Submuestree una imagen con factor 1 a 10. Muestre y comente los resultados.



Me perdí bastante en como redimensionar la imagen y aun no entiendo bien por qué salieron ciertos resultados pero espero comprenderlos más adelante.

- b. Seleccione una imagen, de esta seleccione un objeto que desea censurar y usando el comando imresize a este pequeño objeto reduzca y aumente su tamaño varias veces y luego ponga el objeto en su lugar. Esto generaría que dicho objeto se pixele. El resultado sería algo como esto.

