

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.48 - PROCESAMIENTO DE IMÁGENES

TRABAJO PRÁCTICO N°1

Guía 1

Grupo 1:

Matías Agustín
LARROQUE
Leg. 56597

Tomás Agustín
GONZÁLEZ ORLANDO
Leg. 57090

Manuel Fernando
MOLLÓN
Leg. 58023

Profesores:

Daniel JACOBY

Entregado: 24 de Agosto de 2020

Guía 1 - Procesamiento de Imágenes

1 Modelado de Ojo

Suponiendo al ojo como un sistema lente-sensor, con el sensor conformado por una densidad de conos de 150000 conos/mm² y un área de $1.5mm \times 1.5mm = 2.25mm^2 \Rightarrow 337500$ conos en total. Por lo tanto, se pueden aproximar unos $\sqrt{337500} \simeq 580$ conos tanto de largo como de ancho. Entonces por cada milímetro (lineal) se tienen $\frac{580}{1.5} = 387 \frac{conos}{mm}$.

Para que un pixel de la imagen sea perceptible por el ojo, debe captarse un tamaño mayor o igual a el de un cono. Por lo tanto debe captarse un tamaño lineal $\geq \frac{1}{387}mm \simeq 2.5\mu m$. Teniendo en cuenta la lente, cuya distancia al sensor es de $15mm$ $0.015m$ aproximadamente, y que la distancia al pixel es de $30cm = 0.3m$, entonces el mínimo tamaño lineal "X" de un pixel se calcula de la siguiente manera:

$$\frac{X}{0.3} = \frac{2.5 \times 10^{-6}}{0.015} \Rightarrow X = 50\mu m$$

Hoy en día en los pixeles de celulares rondan en promedio aproximadamente $1\mu m$ lineal. Esto tiene sentido, ya que de esta manera el ojo humano no puede distinguir cada pixel, lo cual da la sensación de que se tiene en frente un "continuo".

También se puede destacar que al ir aumentando la resolución, llega un punto en que no tendría sentido una mayor resolución ya que significaría un tamaño menor de pixel el cual no sería detectable por el ojo humano (o al menos por el modelo planteado).

2 Cálculos de Resoluciones

Supongase que se fotografía una imagen con una cámara que tiene una distancia focal de $35mm$, con un CCD de $10mm \times 10mm$ y $1Mpixel$ de resolución. Se pretende entonces calcular cuantos milímetros representa un pixel en la imagen fotografiada. Se calcula de la siguiente manera:

Sea "X" el tamaño lineal (ya sea vertical u horizontal) de la imagen que capta la lente de la cámara, sea la distancia de la imagen a la cámara igual a $1m$ y dado que en el CCD existen $\frac{1000pixels}{10mm}$ lineales:

$$\frac{X}{1} = \frac{0.01}{0.0035} \Rightarrow X \simeq 286mm$$

Entonces para estos $286mm$ la cámara dispone de $1000pixels$, por lo tanto un pixel representa aproximadamente $5.4mm$ de la imagen captada por la lente de la cámara.

3 Medición de Resolución

Se utilizó la cámara de un celular “Iphone 7 plus” (12Mpx de resolución = 4032x3024) para fotografiar la imagen “iso-rag.jpg”.

A continuación se muestra la imagen “iso-rag”:

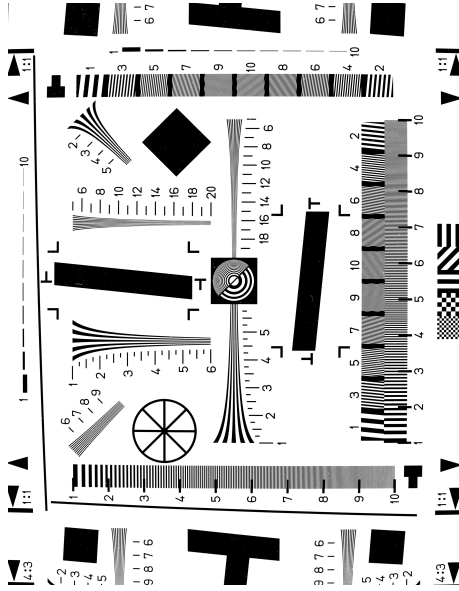


Figure 1: Imagen original

La idea del experimento, es utilizar las barras intercaladas de negro y blanco de la foto tomada a la imagen propuesta. Una vez tomada la foto, se debe identificar la zona de “alias” de las barras, tanto horizontales como verticales. Es decir, que se identifica la zona en que las barras están lo suficientemente cerca como para que se comience a tener una pérdida de fidelidad en la foto debido a falta de resolución. Entonces, en esta zona de “alias” se podrá identificar el porcentaje utilizado “efectivo” de la cámara, dado por el cociente entre el número de barras de la imagen original y el número de barras de la imagen tomada por foto.

A continuación se muestran las fotos tomas con la cámara celular:

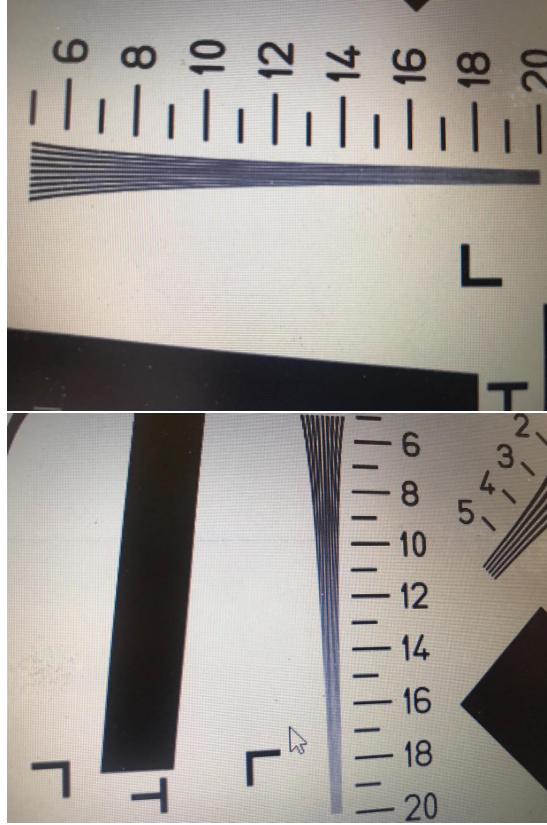


Figure 2: Imagen capturada por cámara

Entonces, se tiene lo siguiente:

$$HORIZONTAL \rightarrow Pixeles_{efectivos} = \frac{pixeles_{ideales}}{pixeles_{reales}} \cdot 4032 = \frac{9}{14} \cdot 4032 \simeq 2592$$

$$VERTICAL \rightarrow Pixeles_{efectivos} = \frac{pixeles_{ideales}}{pixeles_{reales}} \cdot 3024 = \frac{9}{13} \cdot 3024 \simeq 2093$$

Esto da como resultado una resolución efectiva de $1728 \times 2093 \simeq 5Mpixels < 12Mpixels$. Esto demuestra que los pixeles pasados como especificación, no todos son para representar pixeles visibles.

4 Frecuencia Espacial

resuelto en main.py (run_ej4())

Se requiere opencv-python 4.4.0.42 y numpy 1.19.1.

Se puede notar que el cuadrado central que tiene un fondo más oscuro, “parece” a la vista más brillante o claro, con respecto al pixel central, que tiene la misma luminancia pero con un fondo más claro que el primero.

Luego, se varió la luminancia de ambos pixeles centrales para ver el efecto a distintos valores de luminancia. En los valores extremos (0 - negro y 255 -blanco), se vió disminuido el efecto visual.

5 Decimación e Interpolación

5.1 Decimación

Se procedió a realizar varias decimaciones de la imagen “mono.bmp”, que se pueden realizar en main.py (run_ej5a()), run_ej5b() y run_ej5c()), en las cuales se tomaron porciones de 4x4 pixeles y se obtuvieron los siguientes resultados:

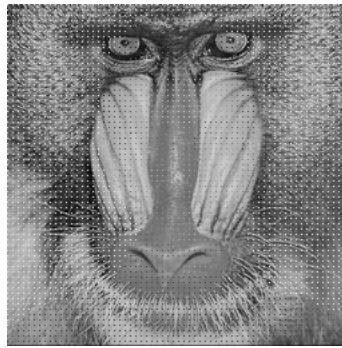


Figure 3: Imagen “mono.bmp”



Figure 4: Decimaciones - Resultados

En la primera imagen (correspondiente a la primera decimación), se tomó el pixel en la posición (1, 1). En la segunda, se tomó el pixel en la posición (0,0). En la tercera y última, se realizó un promedio entre los 16 pixeles.

En la primera decimación, se puede notar una imagen más “pixelada” lo cual tiene sentido ya que se descartan detalles al decimar.

En la segunda decimación, se puede notar una imagen que no tiene relación con la imagen original, aunque en realidad si se observa con detenimiento la imagen original, esta última presenta un punteado equiespaciado, el cual efectivamente se corresponde con la imagen que se obtuvo al decimar.

En la tercer decimación, se logra eliminar el punteado y suavizar algunos detalles de la imagen original debido a la promediación realizada.

5.2 Interpolación

Luego de haber realizado las decimaciones, se toman cada una de las imágenes y se le aplica una interpolación, tanto bilineal como cúbica. Los resultados son los siguientes (para cada resultado de muestra también el espectro de la imagen obtenida):

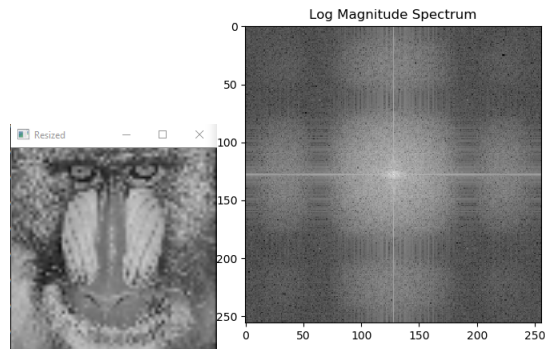


Figure 5: Interpolación A - bilineal

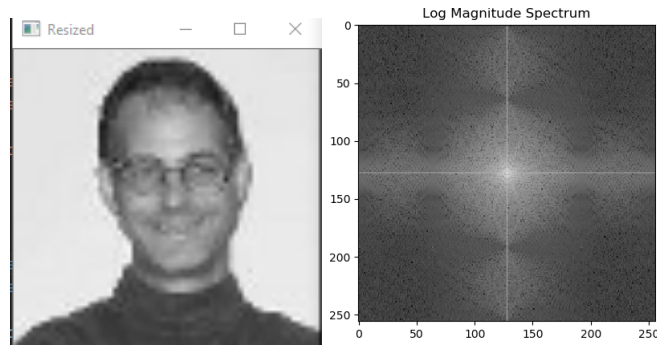


Figure 6: Interpolación B - bilineal

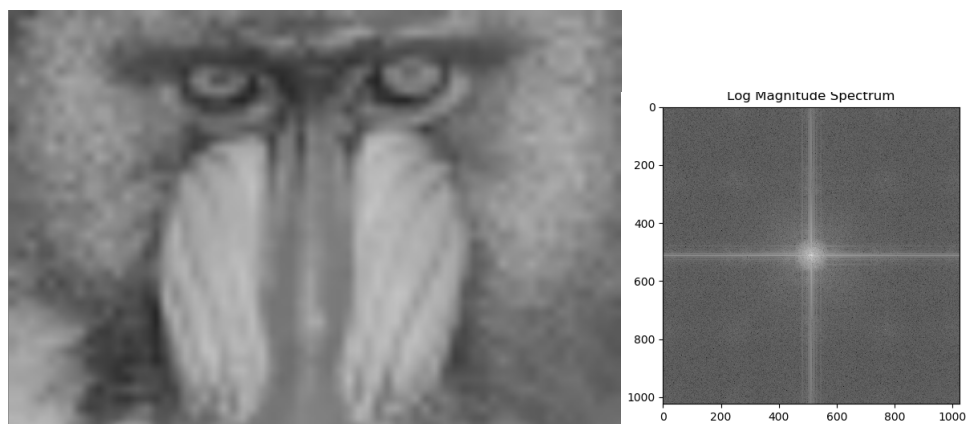


Figure 7: Interpolación C - bilineal

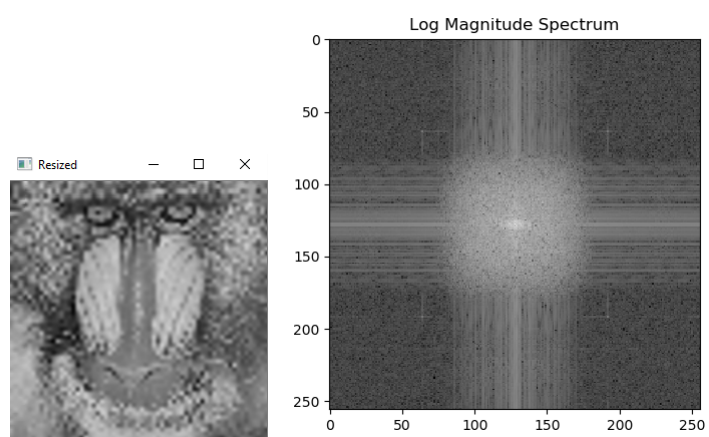


Figure 8: Interpolación A - bicúbica

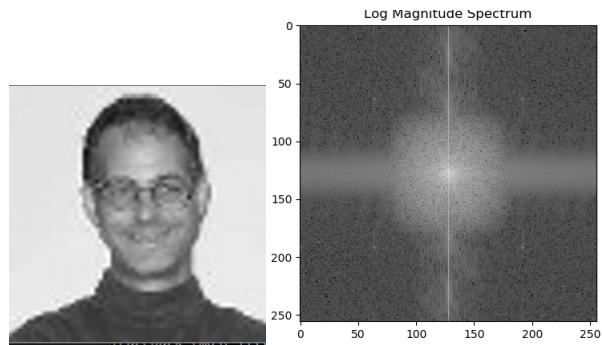


Figure 9: Interpolación B - bicúbica

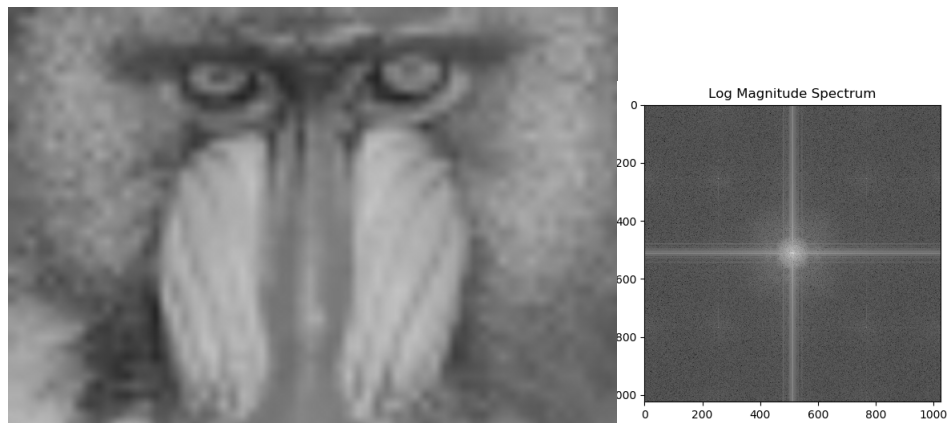


Figure 10: Interpolación C - bicúbica

Se puede concluir que se regeneran las imágenes correctamente, en el sentido que se sigue identificando tanto el mono como la persona (Dani) luego de interpolar. En cuanto al tipo de interpolación utilizado, se puede notar que con la interpolación bicúbica se obtienen rasgos mas “suaves”, que se notan tanto en la imagen como en los cambios de la respuesta en frecuencia, la cual toma valores de pasabajos.