

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Análisis de Algoritmos

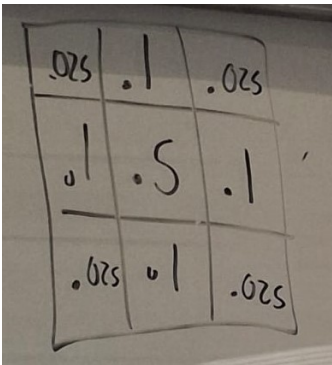
Reporte de práctica

Nombre del alumno:	Samuel David Pérez Brambila
Código del alumno:	222966286
Profesor:	Erasmus Gabriel Martínez Soltero
Título de la práctica:	"Filtros"
Fecha:	30 de Septiembre de 2024

Metodología

Siguiendo las indicaciones y en base a la lista, se aplicará el filtro para impar, es decir, el que se ve en la imagen:

17 222966286 PEREZ BRAMBILA SAMUEL DAVID



Código:

```
Spyder (Python 3.12)
Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

C:\Users\samuel\OneDrive\Escritorio\Archivos UDG\5º\Análisis de Algoritmos\Tarea4.py

prueba_filtros_derecha.py X prueba_filtros.py X Tarea4.py X
3 Created on Sun Sep 22 13:13:39 2024
4
5 @author: SamuelBrambila
6 """
7 import cv2 as cv
8 import numpy as np
9
10 # Cargar la imagen a color
11 imagenOriginal = cv.imread('ciudad_oscura.jpg')
12
13 # Obtener las dimensiones de la imagen (filas y columnas)
14 filas, columnas, canales = np.shape(imagenOriginal)
15
16 # Crear una copia de la imagen para el resultado del filtro
17 imagenFiltrada = imagenOriginal.copy()
18
19 # Definimos el filtro con apoyo de una matriz
20 filtro_3x3 = np.zeros((3, 3), dtype=float)
21 for i in range(3):
22     for j in range(3):
23         # Asignar los valores dinámicamente
24         if i == 1 and j == 1:
25             filtro_3x3[i, j] = 0.5 # Centro
26         elif (i == 0 and j == 1) or (i == 1 and j == 0) or (i == 1 and j == 2) or (i == 2 and j == 1):
27             filtro_3x3[i, j] = 0.1 # Bordes
28         else:
29             filtro_3x3[i, j] = 0.025 # Esquinas
30
31 # Aplicar el filtro
32 for canal in range(canales):
33     for fila in range(1, filas-1):
34         for columna in range(1, columnas-1):
35             suma = 0
36             # Recorrer la ventana 3x3
37             for i in range(-1, 2): # Fila de la ventana
38                 for j in range(-1, 2): # Columna de la ventana
39                     valor_filtro = filtro_3x3[i+1, j+1] # Valor del filtro 3x3 generado
40                     suma += imagenOriginal[fila+i, columna+j, canal] * valor_filtro
41
42             # Asignar el valor calculado, asegurando que esté en el rango [0, 255]
43             imagenFiltrada[fila, columna, canal] = np.clip(suma, 0, 255)
44
45 # Guardar las imágenes
46 cv.imwrite('imagen_original.png', imagenOriginal)
47 cv.imwrite('imagen_filtrada.png', imagenFiltrada)
48
49 # Mostrar la imagen original y la imagen filtrada
50 cv.imshow('imagen original', imagenOriginal)
51 cv.imshow('imagen filtrada', imagenFiltrada)
52 cv.waitKey(0)
53 cv.destroyAllWindows()
```

Resultados

Imagen original



Imagen con filtro

