# LABORATORIO: ELIMINACION DE ANOMALIAS DE LA IMAGEN.

## IDENTIFACIÓN OBJETIVOS.

El fin de este laboratorio es el de familiarizar al alummnado con las técnicas y herramientas de procesado de imagen. Para ello, se han identificado los siguientes objetivos a tratar:

¿Qué anomalía se va a tratar?. Identificación de una solución general. Evaluación de dicha solución. Plasmar los resultados en un Notebook.

### Antes de empezar con el Notebook, se van a definir los siguientes apartados que dicho Notebook debe reflejar:

**REQUISITOS NOTEBOOK.** 

Nombre de los integrantes. Asistencia a las reuniones, conforme a la tabla Excel y que se aporta como anexo a este Notebook.

Descripción del problema. 2 ó más imágenes con la anomalía a tratar. Solución propuesta. Ejecución comentada del algoritmo empleado. NOMBRE DE LOS INTEGRANTES.

## Alberto Jiménez Sánchez. Iñigo Blázquez Tatiegi. Luis García Tíscar. Guillermo Botana Novillo-Fertrell.

la anomalía conocida como el "ruido de sal y pimienta".

Los integrantes de este equipo quedan referidos en el siguiente apartado:

ASISTENCIA A LAS REUNIONES. Tras la primera reunión, la cual data del lunes 28 de diciembre, los integrantes de este equipo han acordado abordar las imágenes con

seguimiento del trabajo a realizar.

señales.):

de filtros:

import time

plt.show()

#cargar imagen en color

comienzo1 BW=time.time()

final1\_BW=time.time()

**DESCRIPCION DEL PROBLEMA** 

Se define el ruido como toda señal no deseada y de naturaleza aleatoria que modifica la intensidad de la señal original a percibir. Para

Antes de describir el ruido de sal y pimienta, se van a dar unas breves pautas que introducirán este tipo de ruido.

Al finalizar la primera reunión, los integrantes acuerdan celebrar una nueva reunión el lunes 3 de enero de 2021 para realizar un

### el caso que concierne a este laboratorio, las cámaras fotográficas captan la luz y la convierten en una señal eléctrica de naturaleza digital. Por tanto, el ruido afecta a esta señal eléctrica.(TEMA 4. 4.2 Entropía: concepto y estimación.)

Las señales eléctricas se en afectadas por diferentes tipos de ruido. Estos tipos de ruido son (TEMA 4 4.3 Fuentes perturbadoras de las

Por lo tanto, a la hora de tratar este ruido se van a emplear filtros digitales. Estos filtros se pueden clasificar en

Externos al sistema. Internos al sistema. Es dentro de los ruidos externos al sistema donde se localiza el ruido de sal y pimienta, o ruido impulsivo o shot. Este tipo de ruido se produce normalmente en la cuantificación que se realiza en el proceso de digitalización de la imagen.

(http://alojamientos.us.es/gtocoma/pid/tema3-1.pdf): Filtros en el dominio del espacio. Filtros en el dominio de la frecuencia. Dentro de los filtros en el dominio del espacio, existen dos tipos

(Elimina el ruido de sal (píxeles blancos)). Máximo: selecciona el valor más alto dentro de una ventana ordenada de valores de niveles de gris. (Elimina el ruido de pimienta (píxeles negros)). Mediana: selecciona el valor en la posición intermedia. Esto es, se visita cada píxel de la imagen y se reemplaza por la mediana de los píxeles vecinos. La mediana se calcula ordenando los valores de los píxeles vecinos

en orden y seleccionando el que queda en medio. Este tipo de filtro da muy buenos resultados en caso de ruido de sal y pimienta.

Filtros estadísticos de orden: Mínimo: selecciona el valor más pequeño dentro de una ventana ordenada de valores de niveles de gris.

Filtros lineales. Filtros no lineales. Serán los filtros no lineales los que previsiblemente darán una mejor solución para este tipo de

anomalía. Dentro de los filtros no lineales, y que aborden el ruido de sal y pimienta, se encuentran los siguientes tipos:

Cargar librerias e imagen

import cv2 import numpy as np from matplotlib import pyplot as plt from skimage import img\_as\_float

### plt.figure(figsize=(16,10)) plt.subplot(121), plt.imshow(img1),plt.title(titleImg1) plt.xticks([]), plt.yticks([])

plt.xticks([]), plt.yticks([])

img\_BW=cv2.imread('imagenes/tigre.png')

#cargar imagen en blanco y negro

def showImages(img1,titleImg1,img2,titleImg2):

Función para enseñar en pantalla 2 imagenes

img\_Color=cv2.imread('imagenes/paisaje\_color.png')

plt.subplot(122), plt.imshow(img2),plt.title(titleImg2)

Utilizando funcion cv2.medianBlur In [4]: figure\_size = 3 #aplicando filtro mediana a la imagen en blanco y negro

filtered\_img\_BW = cv2.medianBlur(img\_BW, figure\_size)

salt pepper effect

#aplicando filtro mediana a la imagen en color

salt pepper effect

filtered img Color = cv2.medianBlur(img Color, figure size)

showImages(img Color,'salt pepper effect',filtered img Color,'filtered')

comienzo1 color=time.time()

final1 color=time.time()

showImages(img\_BW,'salt pepper effect',filtered\_img\_BW,'filtered')



else:

comienzo2 BW=time.time() img1=filtrarImagen(img BW) final2 BW=time.time()

showImages(img\_BW,'salt pepper effect',img1,'filtered')

salt pepper effect

else:

temp.append(0)

temp.append(0)

for k in range(filter size):



filtered

filtered

temp.sort() data\_final[i][j] = temp[len(temp) // 2] temp = []return data final def filtrarImagen(img): Funcion que recive una imagen lo divide en tres canales R GB y a cada uno de ellos les aplica la funcio img = img as float(img) r = img[:, :, 0]g = img[:, :, 1]b = img[:, :, 2]#aplicamos el filtro a cada canal newR=median filter(r,3) newG=median filter(g,3) newB=median filter(b,3) #volvemos a juntar todos los canales rows, cols=r.shape mixed image = np.zeros((rows,cols, 3)) mixed image[:,:,0]=newR mixed image[:,:,1]=newG mixed image[:,:,2]=newB return mixed image

temp.append(data[i + z - indexer][j + k - indexer])



print("tiempo requerido por la funcion cv2.medianBlur")

print("tiempo requerido por la funcion manual")

tiempo requerido por la funcion cv2.medianBlur

CONTROL DE ACTIVIDAD GRUPAL

Image(filename = "imagenes/asistencia.png")

HOJA DE CONTROL ACTIVIDAD GRUPAL

reunión o

Ninguna o solo

from IPython.display import Image

Asistencia a reuniones de

equipo por cada integrante

Out[1]:



filtered

### Blanco y negro 0.004996538162231445 color: 0.0 tiempo requerido por la funcion manual Blanco y negro 2.3452799320220947 color: 4.0795066356658936

Asistencia a 3

reuniones

V

print("Blanco y negro", final2 BW-comienzo2 BW, "color:", final2 color-comienzo2 color)

Marcar la opción que corresponda a cada integrante del equipo, en función del número de reuniones a las que ha asistido y el número de actividades definidas por el equipo HORA COMIENZO HORA FIN HITOS **FECHA** ALBERTO JIMENEZ SANCHEZ 28/12/2020

Asistencia a 2

reuniones

Tareas o entregas a realizar por cada miembro del equipo	una tarea	Dos tareas	Tres tareas			
		V				
Estudiante 2 (Apellidos, Nombre)		GO BLAZQUEZ TATI	EGI			
Asistencia a reuniones de equipo por cada integrante	Asistencia a 1 reunión o	Asistencia a 2 reuniones	Asistencia a 3 reuniones			
			V			
Tareas o entregas a realizar por cada miembro del equipo	Ninguna o solo una tarea	Dos tareas	Tres tareas			
		V				
Estudiante 3 (Apellidos, Nombre)		LUIS GARCIA TISCA	₹			
Asistencia a reuniones de equipo por cada integrante	Asistencia a 1 reunión o ninguna	Asistencia a 2 reuniones	Asistencia a 3 reuniones			
			V			
Tareas o entregas a realizar por	Ninguna o solo una tarea	Dos tareas	Tres tareas			

04/01/2021

11/01/2021

11:00

20:00

11:22 IDENTIFICACION ANOMALIA

12:24 PROXIMA REUNION 11 ENERO

20:32 SUBIDA A GITHUB.

SELECCIÓN ALGORITMO IÑIGO

COMPROBACION FUNCIONAMIENTO Y EVALUACION GENERAL.

cada miembro del equipo Asistencia a 2 Asistencia a 3 reunión o Asistencia a reuniones de reuniones reuniones equipo por cada integrante V Ninguna o solo Dos tareas Tres tareas Tareas o entregas a realizar por una tarea cada miembro del equipo V