



César Soltero Pérez

18310460

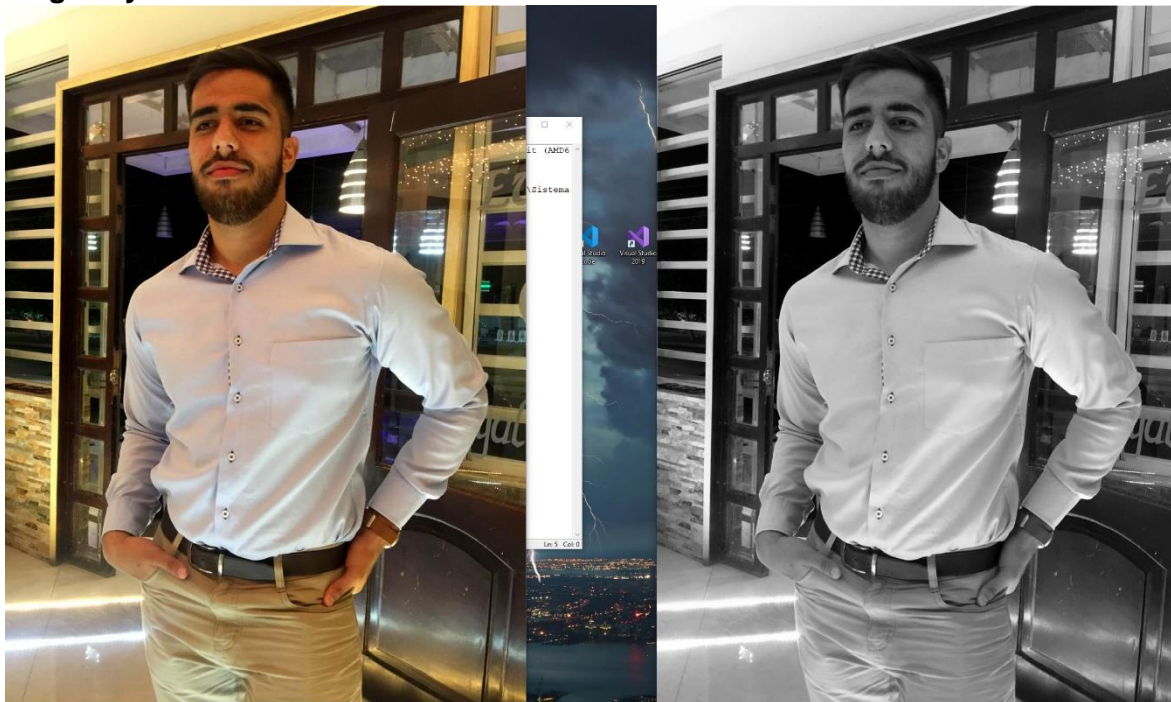
7E1

Sistemas de procesamiento de imágenes y visión artificial

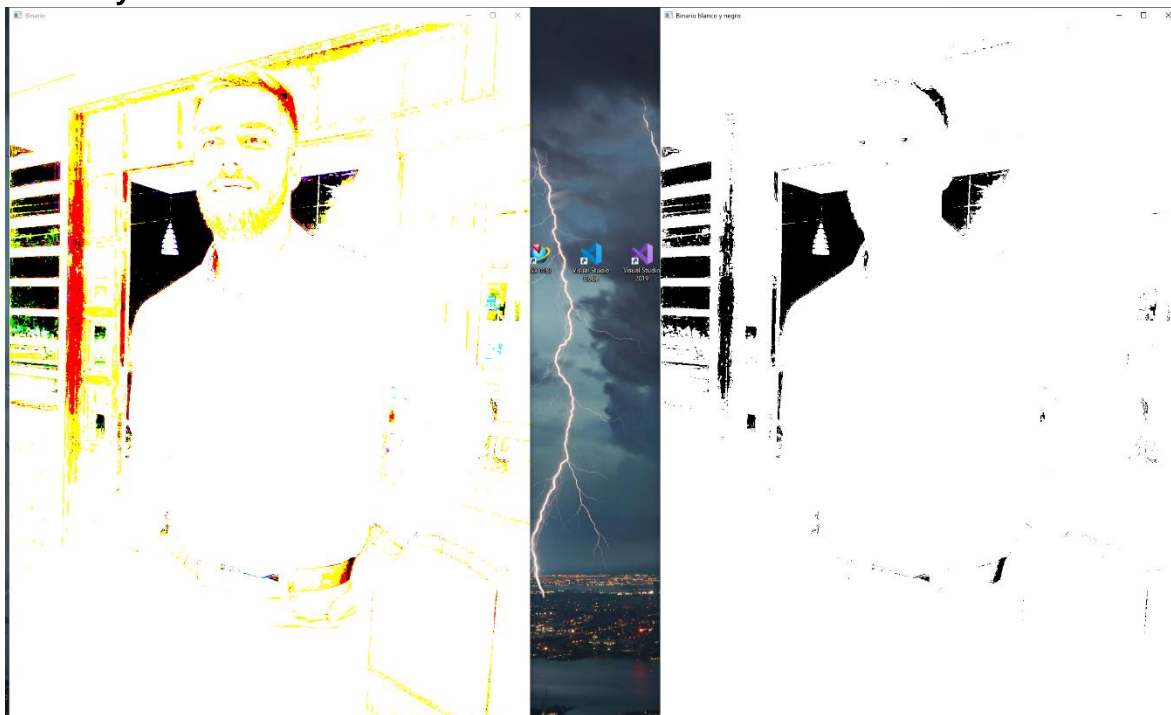
Práctica 5

Objetivo: Utilizar las funciones de umbrales para la recuperación de información.
Threshold1 binary, b_inv, Trunc, To Zero, Tz_inv, Mean, Gaus, Otsu.

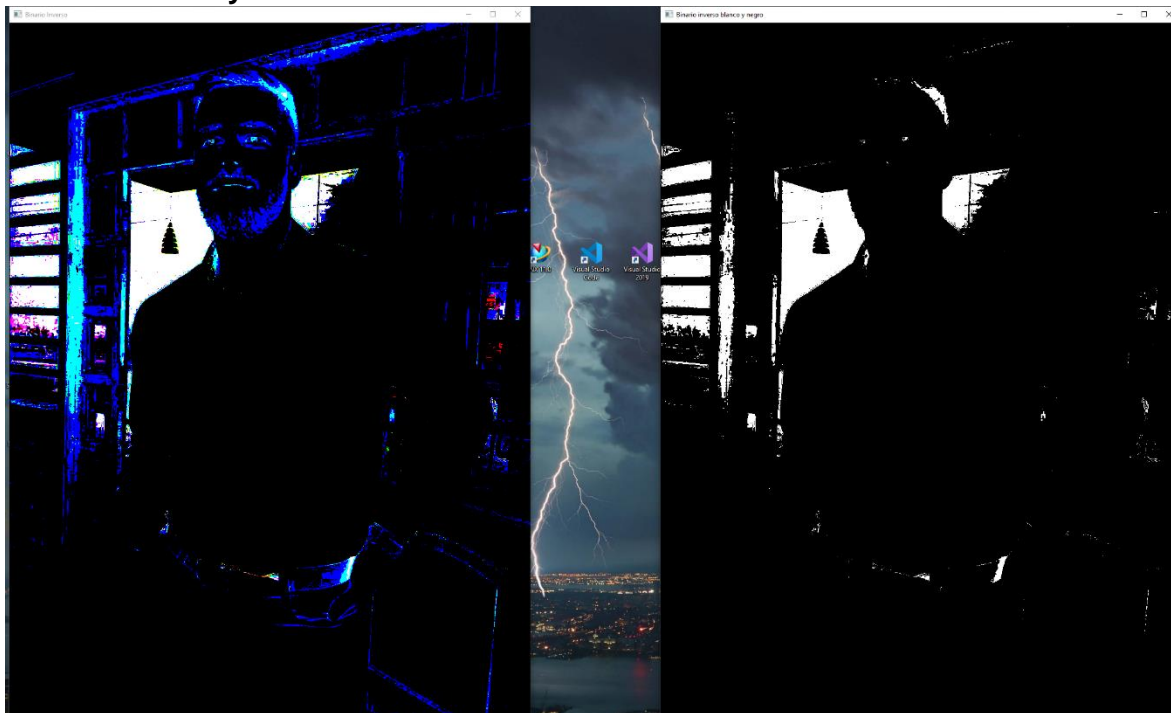
Original y B/N:



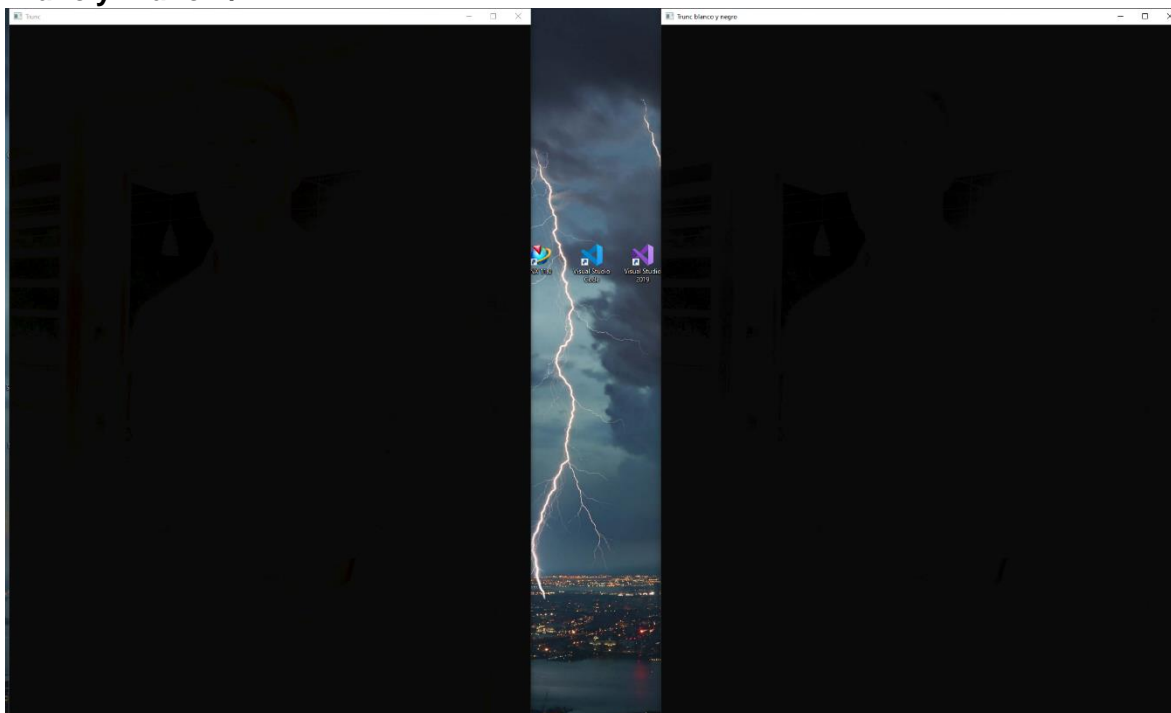
Binario y Binario/BN



Binario Inverso y Binario Inverso B/N



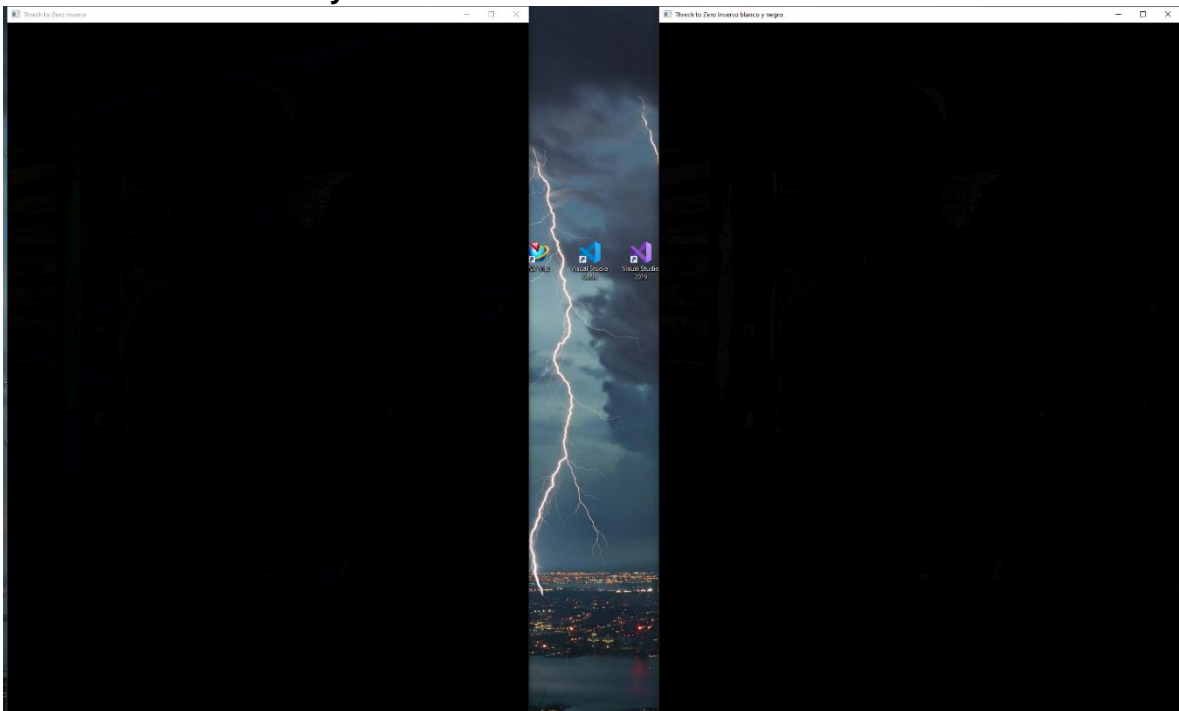
Trunc y Trunc B/N



Thresh to Zero y Thresh to Zero B/N



Thresh to Zero Inverso y Thresh to Zero Inverso B/N



Mean sobre B/N



Gauss sobre B/N



Otsu sobre B/N



Código:

```
import cv2
import numpy as np

imagen = cv2.imread("editada.jpg")

#Blanco y negro
gris = cv2.cvtColor(imagen,cv2.COLOR_BGR2GRAY)

cv2.imshow("Original",imagen)
cv2.moveWindow("Original",0,0)
cv2.imshow("Blanco y negro",gris)
cv2.moveWindow("Blanco y negro",1200,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#Binario
_,bina = cv2.threshold(imagen,10,255,cv2.THRESH_BINARY)
_,bina2 = cv2.threshold(gris,10,255,cv2.THRESH_BINARY)

cv2.imshow("Binario",bina)
cv2.moveWindow("Binario",0,0)
cv2.imshow("Binario blanco y negro",bina2)
cv2.moveWindow("Binario blanco y negro",1200,0)
```

```

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#Binario Inverso
_,binaInv = cv2.threshold(imagen,10,255,cv2.THRESH_BINARY_INV)
_,binaInv2 = cv2.threshold(gris,10,255,cv2.THRESH_BINARY_INV)

cv2.imshow("Binario Inverso",binaInv)
cv2.moveWindow("Binario Inverso",0,0)
cv2.imshow("Binario inverso blanco y negro",binaInv2)
cv2.moveWindow("Binario inverso blanco y negro",1200,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#Trunc

_,trunc = cv2.threshold(imagen,10,255,cv2.THRESH_TRUNC)
_,trunc2 = cv2.threshold(gris,10,255,cv2.THRESH_TRUNC)

cv2.imshow("Trunc",trunc)
cv2.moveWindow("Trunc",0,0)
cv2.imshow("Trunc blanco y negro",trunc2)
cv2.moveWindow("Trunc blanco y negro",1200,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#To Zero

_,zero = cv2.threshold(imagen,10,255,cv2.THRESH_TOZERO)
_,zero2 = cv2.threshold(gris,10,255,cv2.THRESH_TOZERO)

cv2.imshow("Thresh to Zero",zero)
cv2.moveWindow("Thresh to Zero",0,0)
cv2.imshow("Thresh to Zero blanco y negro",zero2)
cv2.moveWindow("Thresh to Zero blanco y negro",1200,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#To Zero Inverso

_,zeroInv = cv2.threshold(imagen,10,255,cv2.THRESH_TOZERO_INV)
_,zeroInv2 = cv2.threshold(gris,10,255,cv2.THRESH_TOZERO_INV)

cv2.imshow("Thresh to Zero Inverso",zeroInv)
cv2.moveWindow("Thresh to Zero Inverso",0,0)
cv2.imshow("Thresh to Zero Inverso blanco y negro",zeroInv2)
cv2.moveWindow("Thresh to Zero Inverso blanco y negro",1200,0)

```

```

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#Mean

mean2 =
cv2.adaptiveThreshold(gris,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,2
)
cv2.imshow("Mean blanco y negro",mean2)
cv2.moveWindow("Mean blanco y negro",0,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#Gauss

gauss =
cv2.adaptiveThreshold(gris,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,
115,1)
cv2.imshow("Gauss",gauss)
cv2.moveWindow("Gauss",0,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

#Otsu

_,otsu = cv2.threshold(gris,125,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
cv2.imshow("Otsu",otsu)
cv2.moveWindow("Otsu",0,0)

opc = cv2.waitKey(0)
cv2.destroyAllWindows()

```

Repositorio:

<https://github.com/Cesarsp41/Practica-5>