

Algoritmo 1

Muñoz Nuñez Ian Emmanuel

Visión Robótica

Punto A

```
# Se importa la librería opencv para manipular la imagen
import cv2
# Se importa la librería numpy como np, también para manipular la imagen
import numpy as np
# Se importa la librería matplotlib.pyplot para mostrar la imagen
import matplotlib.pyplot as plt

# Se crea una imagen en escala de grises de tamaño 8x15
imagen = np.zeros((8, 15), dtype= np.uint8)

# Se dibuja un pixel con un nivel de 255 en la posición 'y'=2 y 'x'=3
imagen[2, 3] = 255

# Se dibuja un pixel con un nivel de 170 en la posición 'y'=2 y 'x'=11
imagen[2, 11] = 170

# Se dibuja un pixel con un nivel de 85 en la posición 'y'=5 y 'x'=7
imagen[5, 7] = 85

# Se muestra la imagen en el pyplot
plt.imshow(imagen, cmap="gray")

# Se muestra el pyplot
plt.show()
```

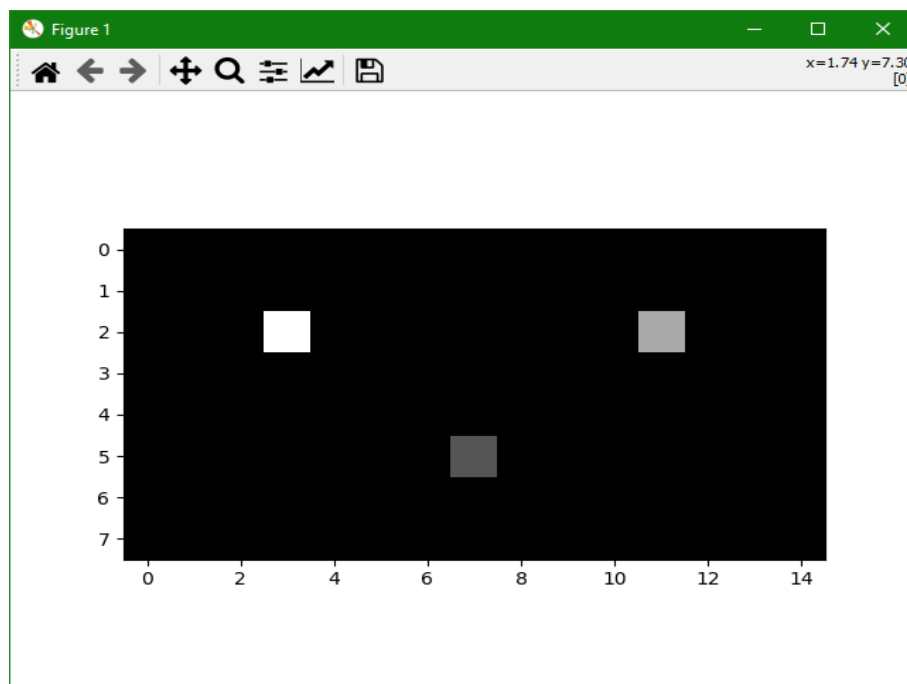
Para el código del punto A, primero se importan las librerías necesarias para realizarlo, como opencv, numpy, matplotlib, después se crea una imagen de 8x15 en escala de grises con la librería numpy. Se dibujan tres pixeles en distintas posiciones con valores distintos, se agrega la imagen al pyplot y al final se muestra el pyplot.

Se muestra una imagen del código.

```
File Edit Selection View Go Run Terminal Help
main.py - opencv - Visual Studio Code

main.py x
main.py > ...
1 # Se importa la librería opencv para manipular la imagen
2 import cv2
3 # Se importa la librería numpy como np, también para manipular la imagen
4 import numpy as np
5 # Se importa la librería matplotlib.pyplot para mostrar la imagen
6 import matplotlib.pyplot as plt
7
8 # Se crea una imagen en escala de grises de tamaño 8x15
9 imagen = np.zeros((8, 15), dtype= np.uint8)
10
11 # Se dibuja un pixel con un nivel de 255 en la posición 'y'=2 y 'x'=3
12 imagen[2, 3] = 255
13
14 # Se dibuja un pixel con un nivel de 170 en la posición 'y'=2 y 'x'=11
15 imagen[2, 11] = 170
16
17 # Se dibuja un pixel con un nivel de 85 en la posición 'y'=5 y 'x'=7
18 imagen[5, 7] = 85
19
20 # Se muestra la imagen en el pyplot
21 plt.imshow(imagen, cmap="gray")
22
23 # Se muestra el pyplot
24 plt.show()
```

Se muestra una imagen de lo que el código realiza.



Punto B

```
# Se importa la librería opencv para manipular la imagen
import cv2
# Se importa la librería numpy como np, también para manipular la imagen
import numpy as np
# Se importa la librería matplotlib.pyplot para mostrar la imagen
import matplotlib.pyplot as plt

# Se crea una imagen en escala de grises de tamaño 12x6
imagen = np.zeros((12, 6), dtype= np.uint8)

# Se dibuja un rectangulo con un nivel de 200 desde la fila 1 hasta la 2 y
# desde la columna 1 hasta la 4
imagen[1:3, 1:5] = 200

# Se dibuja un rectangulo con un nivel de 150 desde la fila 4 hasta la 10 y
# desde la columna 1 hasta la 4
imagen[4:11, 1:5] = 150

# Se muestra la imagen en el pyplot
plt.imshow(imagen, cmap="gray")

# Se muestra el pyplot
plt.show()
```

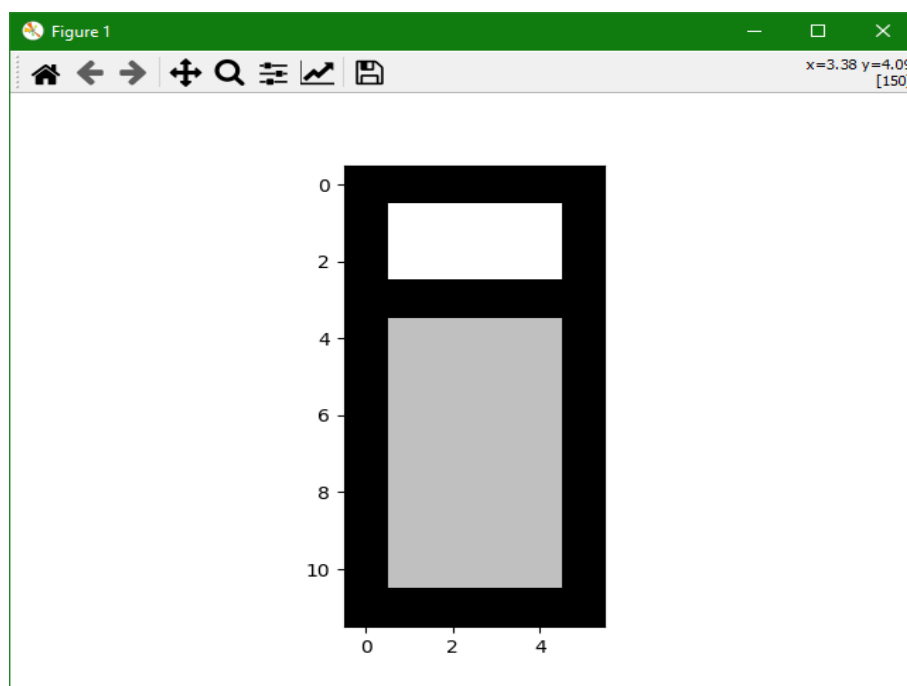
Para el código del punto B, primero se importan las librerías necesarias para realizarlo, como opencv, numpy, y matplotlib, después se genera una imagen de tamaño 12x6 en escala de grises, con la librería numpy. Luego de esto se generan los rectangulos, uno desde la fila 1 a la 2 y desde la columna 1 a la 4, y otro desde la fila 4 hasta la 10 y desde la columna 1 a la 4, ambos con distintos valores. Al final se agrega la imagen al pyplot de matplotlib y se muestra el pyplot.

Se muestra una imagen del código.

```
File Edit Selection View Go Run Terminal Help
main.py - opencv - Visual Studio Code

main.py x
main.py > ...
1 # Se importa la librería opencv para manipular la imagen
2 import cv2
3 # Se importa la librería numpy como np, también para manipular la imagen
4 import numpy as np
5 # Se importa la librería matplotlib.pyplot para mostrar la imagen
6 import matplotlib.pyplot as plt
7
8 # Se crea una imagen en escala de grises de tamaño 12x6
9 imagen = np.zeros((12, 6), dtype= np.uint8)
10
11 # Se dibuja un rectángulo con un nivel de 200 desde la fila 1 hasta la 2 y
12 # desde la columna 1 hasta la 4
13 imagen[1:3, 1:5] = 200
14
15 # Se dibuja un rectángulo con un nivel de 150 desde la fila 4 hasta la 10 y
16 # desde la columna 1 hasta la 4
17 imagen[4:11, 1:5] = 150
18
19 # Se muestra la imagen en el pyplot
20 plt.imshow(imagen, cmap="gray")
21
22 # Se muestra el pyplot
23 plt.show()
24
```

Se muestra una imagen de lo que realiza el código.



Punto C

```
# Se importa la librería opencv para manipular la imagen
import cv2
# Se importa la librería numpy como np, también para manipular la imagen
import numpy as np
# Se importa la librería matplotlib.pyplot para mostrar la imagen
import matplotlib.pyplot as plt

# Se crea una imagen RGB de tamaño 9x8
imagen = np.zeros((9, 8, 3), dtype= np.uint8)

# Se dibuja un pixel en el canal B con un nivel de 255 en la posición 'y'=1 y 'x'=1
imagen[1, 1, 0] = 255
# Se dibuja un pixel en el canal G con un nivel de 255 en la posición 'y'=1 y 'x'=1
imagen[1, 1, 1] = 255

# Se dibuja un pixel en el canal B con un nivel de 255 en la posición 'y'=3 y 'x'=6
imagen[3, 6, 0] = 255
# Se dibuja un pixel en el canal R con un nivel de 255 en la posición 'y'=3 y 'x'=6
imagen[3, 6, 2] = 255

# Se dibuja un pixel en el canal G con un nivel de 255 en la posición 'y'=7 y 'x'=3
imagen[7, 3, 1] = 255
# Se dibuja un pixel en el canal R con un nivel de 255 en la posición 'y'=7 y 'x'=3
imagen[7, 3, 2] = 255

# Se modifica el formato de la imagen de BGR a RGB
rgb = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)

# Se muestra la imagen en el pyplot
plt.imshow(rgb)

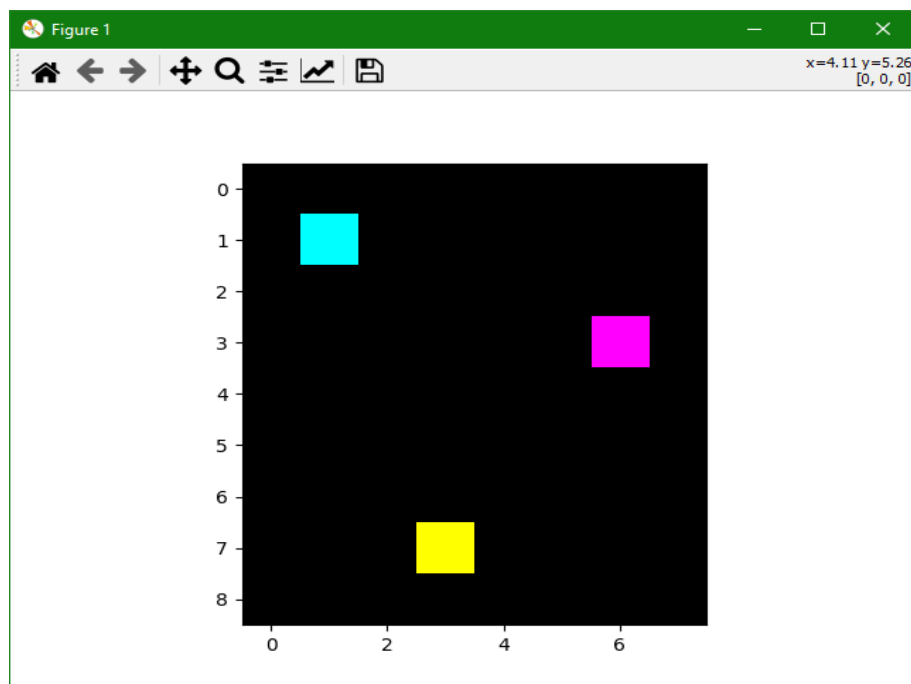
# Se muestra el pyplot
plt.show()
```

Para el código del punto C, primero se importan todas las librerías necesarias para realizarlo, como opencv, matplotlib y numpy. Luego se genera una imagen de 9x8 en BGR con la librería numpy, después se dibujan 3 pixeles en distintas posiciones con colores y valores distintos. Luego se modifica el formato de la imagen de BGR a RGB, es más sencillo manejarla en RGB. Al final se agrega la imagen(RGB) al pyplot y se muestra el pyplot.

Se muestra una imagen del código.

```
main.py - opencv - Visual Studio Code
1 # Se importa la libreria opencv para manipular la imagen
2 import cv2
3 # Se importa la libreria numpy como np, también para manipular la imagen
4 import numpy as np
5 # Se importa la libreria matplotlib.pyplot para mostrar la imagen
6 import matplotlib.pyplot as plt
7
8 # Se crea una imagen RGB de tamaño 9x8
9 imagen = np.zeros((9, 8, 3), dtype= np.uint8)
10
11 # Se dibuja un pixel en el canal B con un nivel de 255 en la posición 'y'=1 y 'x'=1
12 imagen[1, 1, 0] = 255
13 # Se dibuja un pixel en el canal G con un nivel de 255 en la posición 'y'=1 y 'x'=1
14 imagen[1, 1, 1] = 255
15
16 # Se dibuja un pixel en el canal B con un nivel de 255 en la posición 'y'=3 y 'x'=6
17 imagen[3, 6, 0] = 255
18 # Se dibuja un pixel en el canal R con un nivel de 255 en la posición 'y'=3 y 'x'=6
19 imagen[3, 6, 2] = 255
20
21 # Se dibuja un pixel en el canal G con un nivel de 255 en la posición 'y'=7 y 'x'=3
22 imagen[7, 3, 1] = 255
23 # Se dibuja un pixel en el canal R con un nivel de 255 en la posición 'y'=7 y 'x'=3
24 imagen[7, 3, 2] = 255
25
26 # Se modifica el formato de la imagen de BGR a RGB
27 rgb = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)
28
29 # Se muestra la imagen en el pyplot
30 plt.imshow(rgb)
31
32 # Se muestra el pyplot
33 plt.show()
34
```

Se muestra una imagen de los que el código realiza.



Punto D

```
# Se importa la librería opencv para manipular la imagen
import cv2
# Se importa la librería numpy como np, también para manipular la imagen
import numpy as np
# Se importa la librería matplotlib.pyplot para mostrar la imagen
import matplotlib.pyplot as plt

# Se crea una imagen RGB de tamaño 16x8
imagen = np.zeros((16, 8, 3), dtype= np.uint8)

# Se dibuja un rectangulo en el canal B con un nivel de 125 desde la fila 1 hasta la 2 y
# desde la columna 1 hasta la 6
imagen[1:3, 1:7, 0] = 125
# Se dibuja un rectangulo en el canal G con un nivel de 225 desde la fila 1 hasta la 2 y
# desde la columna 1 hasta la 6
imagen[1:3, 1:7, 1] = 225

# Se dibuja un rectangulo en el canal B con un nivel de 100 desde la fila 4 hasta la 10 y
# desde la columna 2 hasta la 5
imagen[4:11, 2:6, 0] = 100
# Se dibuja un rectangulo en el canal R con un nivel de 250 desde la fila 4 hasta la 10 y
# desde la columna 2 hasta la 5
imagen[4:11, 2:6, 2] = 250

# Se dibuja un rectangulo en el canal G con un nivel de 200 desde la fila 12 hasta la 14 y
# desde la columna 1 hasta la 6
imagen[12:15, 1:7, 1] = 200
# Se dibuja un rectangulo en el canal R con un nivel de 190 desde la fila 12 hasta la 14 y
# desde la columna 1 hasta la 6
imagen[12:15, 1:7, 2] = 190

# Se modifica el formato de la imagen de BGR a RGB
rgb = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)

# Se muestra la imagen en el pyplot
plt.imshow(rgb)

# Se muestra el pyplot
plt.show()
```

Para el código del punto D, primero se importan las librerías necesarias para realizarlo, como opencv, numpy y matplotlib. Luego se genera una imagen de 16x8 en BGR, después se dibujan 3 rectángulos de distinto color en distintos rangos. Luego se cambia el formato de la imagen de BGR a RGB. Al final se agrega la imagen(RGB) al pyplot y se muestra el pyplot.

Se muestra una imagen del código.

```

1 # Se importa la librería opencv para manipular la imagen
2 import cv2
3 # Se importa la librería numpy como np, también para manipular la imagen
4 import numpy as np
5 # Se importa la librería matplotlib.pyplot para mostrar la imagen
6 import matplotlib.pyplot as plt
7
8 # Se crea una imagen RGB de tamaño 16x8
9 imagen = np.zeros((16, 8, 3), dtype=np.uint8)
10
11 # Se dibuja un rectángulo en el canal B con un nivel de 125 desde la fila 1 hasta la 2 y
12 # desde la columna 1 hasta la 6
13 imagen[1:3, 1:7, 0] = 125
14 # Se dibuja un rectángulo en el canal G con un nivel de 225 desde la fila 1 hasta la 2 y
15 # desde la columna 1 hasta la 6
16 imagen[1:3, 1:7, 1] = 225
17
18 # Se dibuja un rectángulo en el canal R con un nivel de 100 desde la fila 4 hasta la 10 y
19 # desde la columna 2 hasta la 5
20 imagen[4:11, 2:6, 0] = 100
21 # Se dibuja un rectángulo en el canal R con un nivel de 250 desde la fila 4 hasta la 10 y
22 # desde la columna 2 hasta la 5
23 imagen[4:11, 2:6, 2] = 250
24
25 # Se dibuja un rectángulo en el canal G con un nivel de 200 desde la fila 12 hasta la 14 y
26 # desde la columna 1 hasta la 6
27 imagen[12:15, 1:7, 1] = 200
28 # Se dibuja un rectángulo en el canal R con un nivel de 190 desde la fila 12 hasta la 14 y
29 # desde la columna 1 hasta la 6
30 imagen[12:15, 1:7, 2] = 190
31
32 # Se modifica el formato de la imagen de BGR a RGB
33 rgb = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)
34
35 # Se muestra la imagen en el pyplot
36 plt.imshow(rgb)
37
38 # Se muestra el pyplot
39 plt.show()
40

```

Se muestra una imagen de lo que realiza el código.

