

Universidad de Guadalajara



Algoritmo 12

Muñoz Nuñez Ian Emmanuel

Visión Robótica

*”Relleno de pixels de objetos segmentados”*

# Índice general

<b>1. Ejercicio 1</b>	<b>2</b>
1.1. Código . . . . .	2
1.2. Capturas del código . . . . .	3
1.3. Imágenes obtenidas . . . . .	4
<b>2. Ejercicio 2</b>	<b>7</b>
2.1. Código . . . . .	7
2.2. Capturas del código . . . . .	8
2.3. Imágenes obtenidas . . . . .	9

# Capítulo 1

## Ejercicio 1

### 1.1. Código

```
# Se importa opencv
import cv2

# Se lee la imagen 'imagen.png'
imagen = cv2.imread("imagen.png")

# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.8, fy=0.8)

# Se cambia el formato de la imagen de BGR a HSV
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se calcula la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (21, 110, 190), (27, 255, 255))
# Se calcula la mascara de la imagen para ser usada en la funcion 'cv2.floodFill'
mask_floodfill = cv2.inRange(imagen_hsv, (21, 110, 190), (27, 255, 255))

# Se rellena la mascara con la funcion 'cv2.floodFill'
cv2.floodFill(mask_floodfill, None, (0, 0), 255)
# Se calcula el negativo del relleno de la mascara
floodfill_negativo = cv2.bitwise_not(mask_floodfill)

# Se combina la mascara con el negativo del relleno de la mascara
fill = cv2.bitwise_or(mask, floodfill_negativo)

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el relleno de la mascara
cv2.imshow("Mascara (Floodfill)", mask_floodfill)
# Se muestra el negativo del relleno
cv2.imshow("Floodfill (Negativo)", floodfill_negativo)
```

```
# Se muestra el resultado final
cv2.imshow("Fill", fill)

# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 1 del algoritmo 12 primero se importa **"opencv"**, se carga la imagen **"imagen.png"** y se cambia su tamaño. Luego se cambia el formato de la imagen de BGR a HSV, se calculan dos máscaras, una para ser usada en la función **"cv2.floodFill()"** y otra para ser rellenada. Después se calcula el relleno de una de las máscaras y su negativo. Por último combinan las máscaras y se muestran todas las imágenes con los resultados obtenidos. Al final **"opencv"** espera a que el usuario presione una tecla para terminar el programa.

## 1.2. Capturas del código

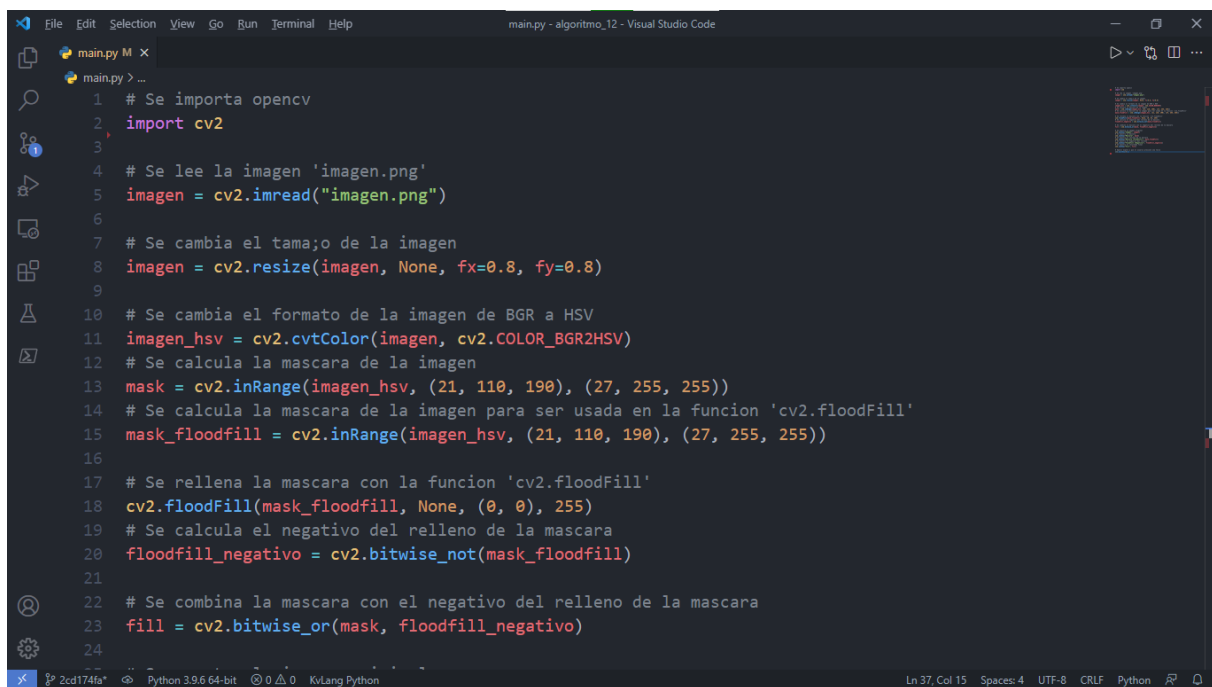


Figura 1.1: Captura 1 del código del ejercicio 1

```
File Edit Selection View Go Run Terminal Help main.py - algoritmo_12 - Visual Studio Code
main.py M x
main.py > ...
17 # Se rellena la mascara con la funcion 'cv2.floodFill'
18 cv2.floodFill(mask_floodfill, None, (0, 0), 255)
19 # Se calcula el negativo del relleno de la mascara
20 floodfill_negativo = cv2.bitwise_not(mask_floodfill)
21
22 # Se combina la mascara con el negativo del relleno de la mascara
23 fill = cv2.bitwise_or(mask, floodfill_negativo)
24
25 # Se muestra la imagen original
26 cv2.imshow("Imagen", imagen)
27 # Se muestra la mascara
28 cv2.imshow("Mascara", mask)
29 # Se muestra el relleno de la mascara
30 cv2.imshow("Mascara (Floodfill)", mask_floodfill)
31 # Se muestra el negativo del relleno
32 cv2.imshow("Floodfill (Negativo)", floodfill_negativo)
33 # Se muestra el resultado final
34 cv2.imshow("Fill", fill)
35
36 # Opencv espera a que el usuario presione una tecla
37 cv2.waitKey(0)
38
```

Figura 1.2: Captura 2 del código del ejercicio 1

### 1.3. Imágenes obtenidas

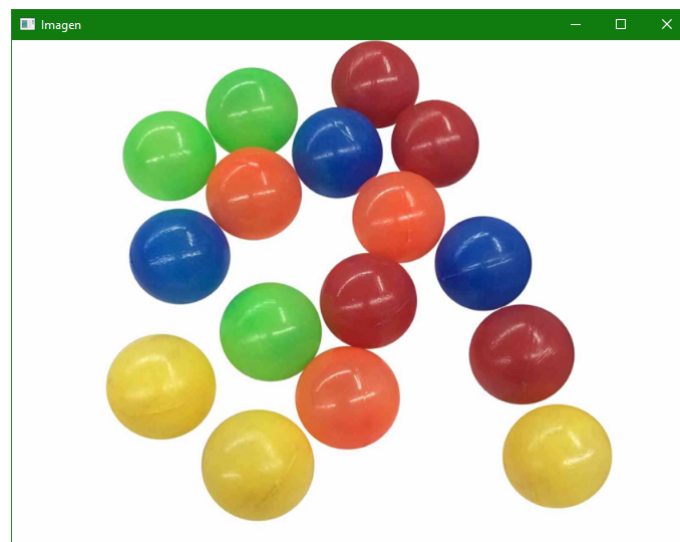


Figura 1.3: Imagen original

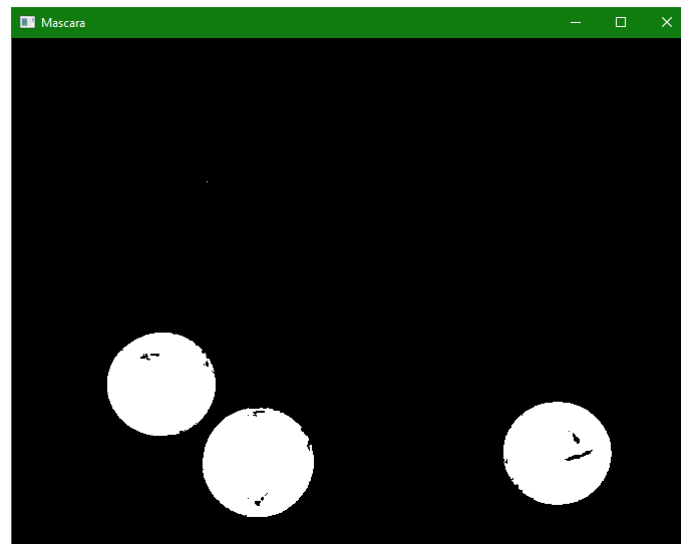


Figura 1.4: Máscara de imagen

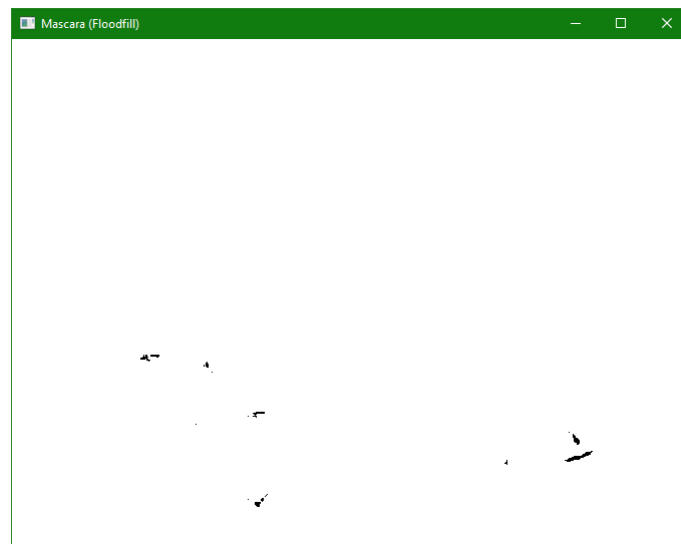


Figura 1.5: Floodfill

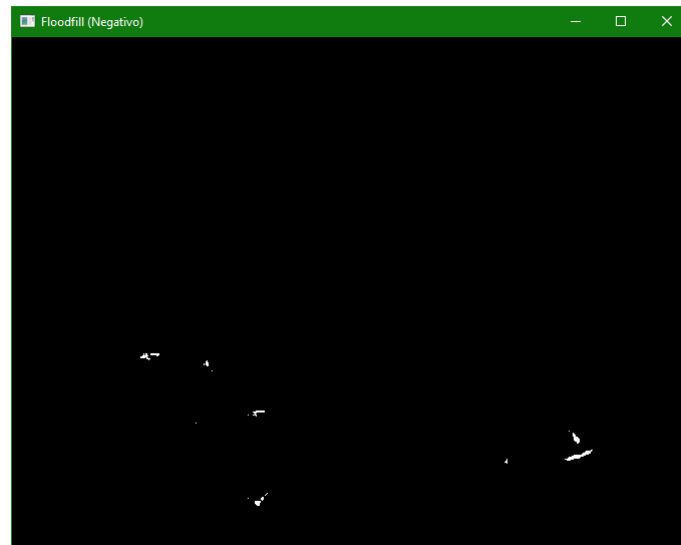


Figura 1.6: Floodfill (Negativo)

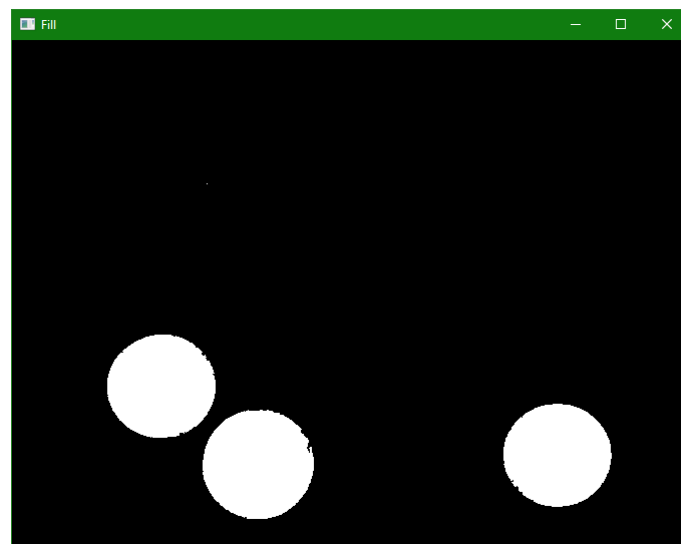


Figura 1.7: Fill

# Capítulo 2

## Ejercicio 2

### 2.1. Código

```
# Se importa opencv
import cv2

# Se lee la imagen 'azul.jpg'
imagen = cv2.imread("azul.jpg")

# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)

# Se cambia el formato de la imagen de BGR a HSV
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se calcula la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (90, 100, 100), (125, 255, 255))
# Se calcula la mascara de la imagen para ser usada en la funcion 'cv2.floodFill'
mask_floodfill = cv2.inRange(imagen_hsv, (90, 100, 100), (125, 255, 255))

# Se rellena la mascara con la funcion 'cv2.floodFill'
cv2.floodFill(mask_floodfill, None, (0, 0), 255)
# Se calcula el negativo del relleno de la mascara
floodfill_negativo = cv2.bitwise_not(mask_floodfill)

# Se combina la mascara con el negativo del relleno de la mascara
fill = cv2.bitwise_or(mask, floodfill_negativo)

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el relleno de la mascara
cv2.imshow("Floodfill", mask_floodfill)
# Se muestra el negativo del relleno
cv2.imshow("Floodfill (Negativo)", floodfill_negativo)
```



```
# Se muestra el resultado final
cv2.imshow("Fill", fill)

# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 1 del algoritmo 12 primero se importa **"opencv"**, se carga la imagen **"azul.jpg"** y se cambia su tamaño. Luego se cambia el formato de la imagen de BGR a HSV, se calculan dos máscaras, una para ser usada en la función **"cv2.floodFill()"** y otra para ser rellenada. Después se calcula el relleno de una de las máscaras y su negativo. Por último combinan las máscaras y se muestran todas las imágenes con los resultados obtenidos. Al final **"opencv"** espera a que el usuario presione una tecla para terminar el programa.

## 2.2. Capturas del código

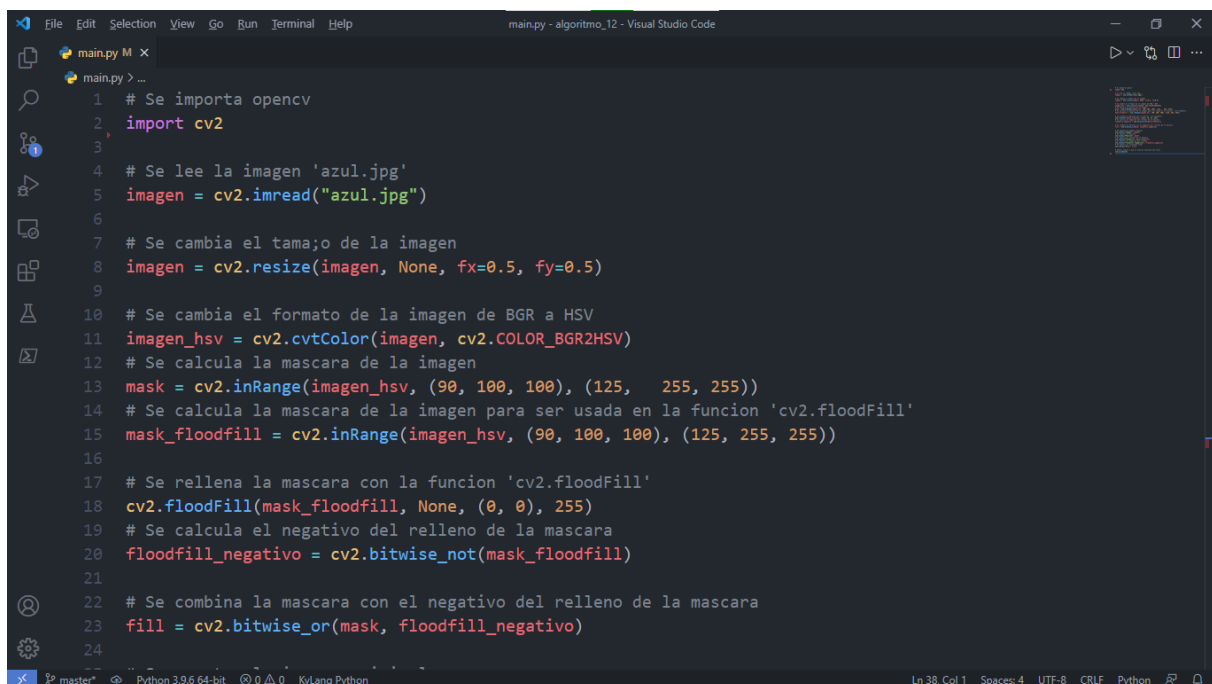


Figura 2.1: Captura 1 del código del ejercicio 2

```
File Edit Selection View Go Run Terminal Help main.py - algoritmo_12 - Visual Studio Code
main.py M x
main.py > ...
18 cv2.floodFill(mask_floodfill, None, (0, 0), 255)
19 # Se calcula el negativo del relleno de la mascara
20 floodfill_negativo = cv2.bitwise_not(mask_floodfill)
21
22 # Se combina la mascara con el negativo del relleno de la mascara
23 fill = cv2.bitwise_or(mask, floodfill_negativo)
24
25 # Se muestra la imagen original
26 cv2.imshow("Imagen", imagen)
27 # Se muestra la mascara
28 cv2.imshow("Mascara", mask)
29 # Se muestra el relleno de la mascara
30 cv2.imshow("Floodfill", mask_floodfill)
31 # Se muestra el negativo del relleno
32 cv2.imshow("Floodfill (Negativo)", floodfill_negativo)
33 # Se muestra el resultado final
34 cv2.imshow("Fill", fill)
35
36 # Opencv espera a que el usuario presione una tecla
37 cv2.waitKey(0)
38
```

Figura 2.2: Captura 2 del código del ejercicio 2

## 2.3. Imágenes obtenidas



Figura 2.3: Imagen original



Figura 2.4: Máscara de imagen

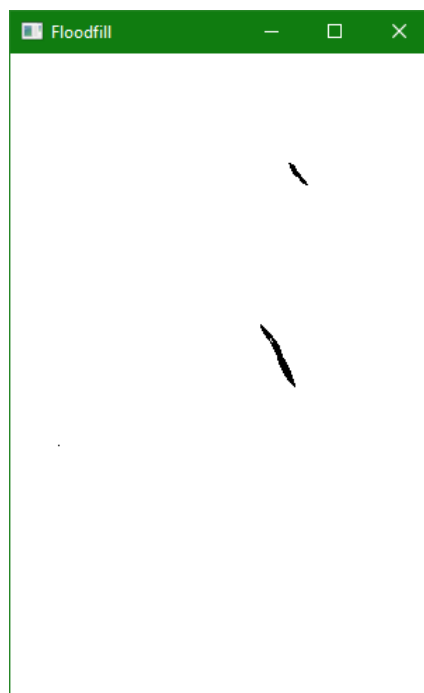


Figura 2.5: Floodfill

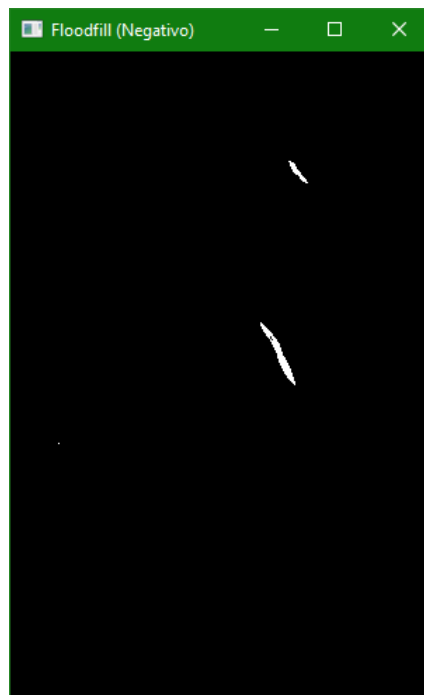


Figura 2.6: Floodfill (Negativo)



Figura 2.7: Fill