



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

ALGORITMO 2

VISION ROBOTICA

INGENIERIA EN ROBOTICA

Alumno: Carlos Omar Rodriguez Vazquez

Introducción

El procesamiento de imágenes es esencial en la visión robótica para tareas como detección de objetos y navegación autónoma. En este reporte, se exploran técnicas de tratamiento de imágenes RGB con Python y OpenCV, incluyendo la apertura de imágenes, la separación de canales de color y la conversión a escala de grises. Se analizan distintos métodos de conversión, como el promedio y los estándares ponderados BT.601 y BT.709, brindando una comprensión clave para aplicaciones en visión robótica.

Marco Teorico

Apertura de una imagen RGB

En el ámbito de procesamiento de imágenes en Python, OpenCV es una biblioteca ampliamente utilizada y robusta. Para abrir una imagen RGB con OpenCV, se emplea la función `cv2.imread()` especificando la ruta del archivo de imagen. Este método retorna una matriz NumPy que representa la imagen en formato BGR (azul, verde, rojo), donde cada canal es representado como un arreglo bidimensional.

```
Im = cv2.imread('nombre_del_archivo')
```

Separación de los canales R,G,B

Para aislar un canal específico de una imagen RGB en OpenCV, se puede acceder directamente a los valores de cada píxel y asignar cero a los canales que no se desean. Por ejemplo, para obtener solo el canal R, se asignan valores de cero a los canales G y B, manteniendo los valores del canal R intactos.

```
Im[:, :, 0] = 0
Im[:, :, 1] = 0
```

Imagen en escala de grises con un solo canal

Una imagen en escala de grises se obtiene de una imagen RGB al calcular un único valor de intensidad para cada píxel, representando su nivel de brillo sin tener en cuenta el color.

```
R = Im[:, :, 2]
G = Im[:, :, 1]
B = Im[:, :, 0]
```

Métodos para convertir a escala de grises

Método de promedio

Este método calcula el promedio de los valores de los canales RGB para cada píxel, resultando en una imagen en escala de grises.

```
Im4 = R//3 + G//3 + B//3
```

Método de promedio ponderado BT.601

Este método utiliza una combinación ponderada de los valores de los canales RGB, donde se aplican factores de ponderación específicos. El estándar BT.601, también conocido como el estándar de televisión PAL/SECAM, define estos factores para simular la respuesta del ojo humano a diferentes colores y luminancias.

$$Im5 = R * 0.299 + G * 0.587 + B * 0.114$$

Método de promedio ponderado BT.709

Similar al método BT.601, este método aplica factores de ponderación específicos para cada canal RGB. El estándar BT.709, ampliamente utilizado en aplicaciones de alta definición como HDTV y Blu-ray, proporciona una mayor precisión en la reproducción del color y la luminancia que su predecesor BT.601.

$$Im6 = R * 0.2126 + G * 0.7152 + B * 0.0722$$

Ejercicios

1. Abrir una imagen RGB.

```
Im = cv.imread('lanterns.png')

cv.imshow('image', Im)
cv.waitKey(0)
```



Figura 1: Imagen del ejercicio 1.

2. Separar los canales R,G,B.

```
# Canal R

ImR = Im.copy()
ImR[:, :, 0] = 0
ImR[:, :, 1] = 0

# Canal G

ImG = Im.copy()
ImG[:, :, 0] = 0
ImG[:, :, 2] = 0

# Canal B

ImB = Im.copy()
ImB[:, :, 1] = 0
```

```

ImB [:,:,2] = 0

cv.imshow('CanalR', ImR)
cv.imshow('CanalG', ImG)
cv.imshow('CanalB', ImB)
cv.waitKey(0)

```

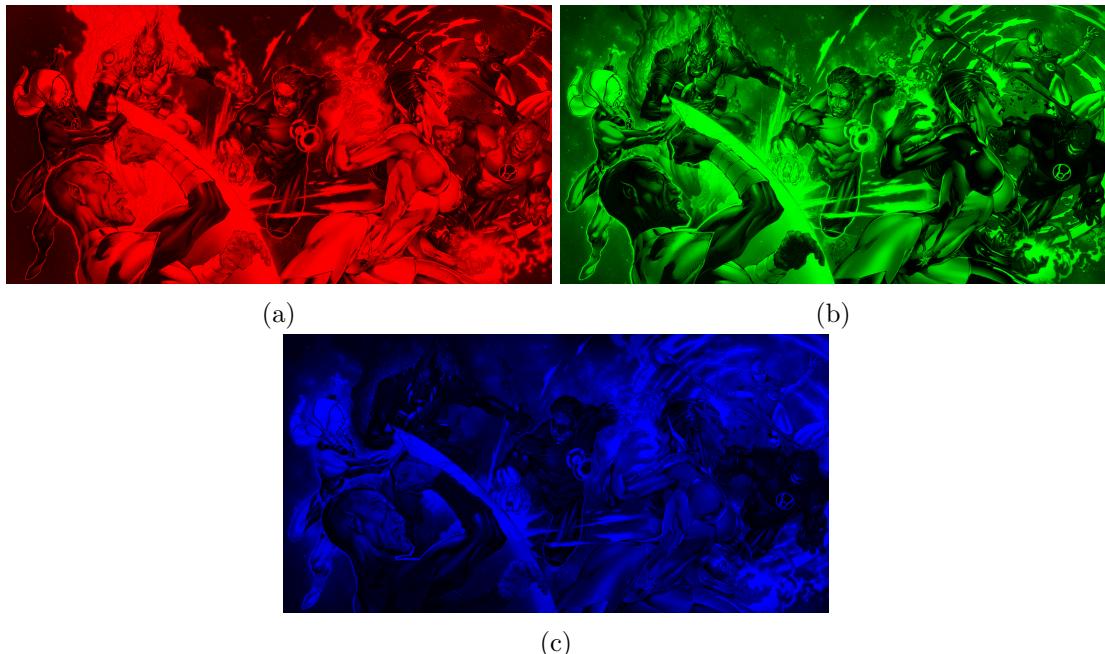


Figura 2: Imagenes del ejercicio 2.

3. Imagen en escala de grises con un solo canal.

```

R = Im[:, :, 2]
G = Im[:, :, 1]
B = Im[:, :, 0]

```

```

# Comparacion de resultado un solo canal

cv.imshow('CanalR', R)
cv.imshow('CanalG', G)
cv.imshow('CanalB', B)
cv.waitKey(0)

```

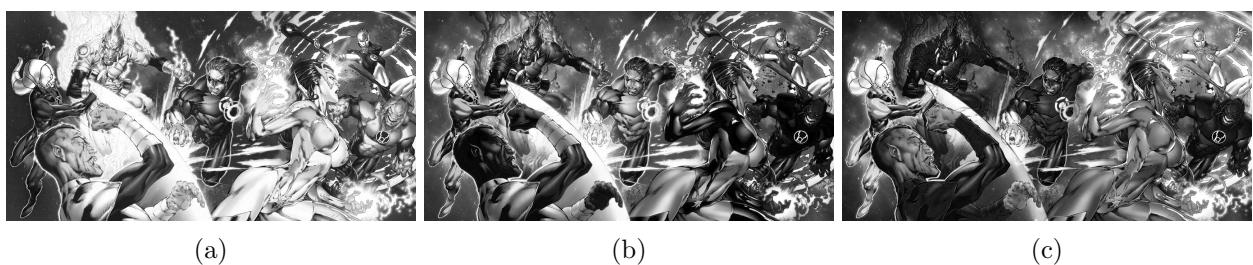


Figura 3: Imagenes del ejercicio 3. (a) Canal R (b) Canal G (c) Canal B.

4. Conversión a escala de grises por el método de promedio.

$$\text{Im4} = \text{R}/3 + \text{G}/3 + \text{B}/3$$

```
cv.imshow('Original Image', Im)
cv.imshow('Escala de grises', Im4)
cv.waitKey(0)
```



Figura 4: Imagen del ejercicio 4.

5. Conversión a escala de grises por el método de promedio ponderado BT.601.

$$\text{Im5} = \text{R}*0.299 + \text{G}*0.587 + \text{B}*0.114$$

```
Im5 = Im5.astype(np.uint8)
```

```
cv.imshow('Original Image', Im)
cv.imshow('Escala de grises', Im5)
cv.waitKey(0)
```



Figura 5: Imagen del ejercicio 5.

6. Conversión a escala de grises por el método de promedio ponderado BT.709.

$$\text{Im6} = \text{R}*0.2126 + \text{G}*0.7152 + \text{B}*0.0722$$

```
Im6 = Im6.astype(np.uint8)
```

```
cv.imshow('Original Image', Im)
cv.imshow('Escala de grises', Im6)
cv.waitKey(0)
```



Figura 6: Imagen del ejercicio 6

7. Comparación de imágenes a escala de grises.

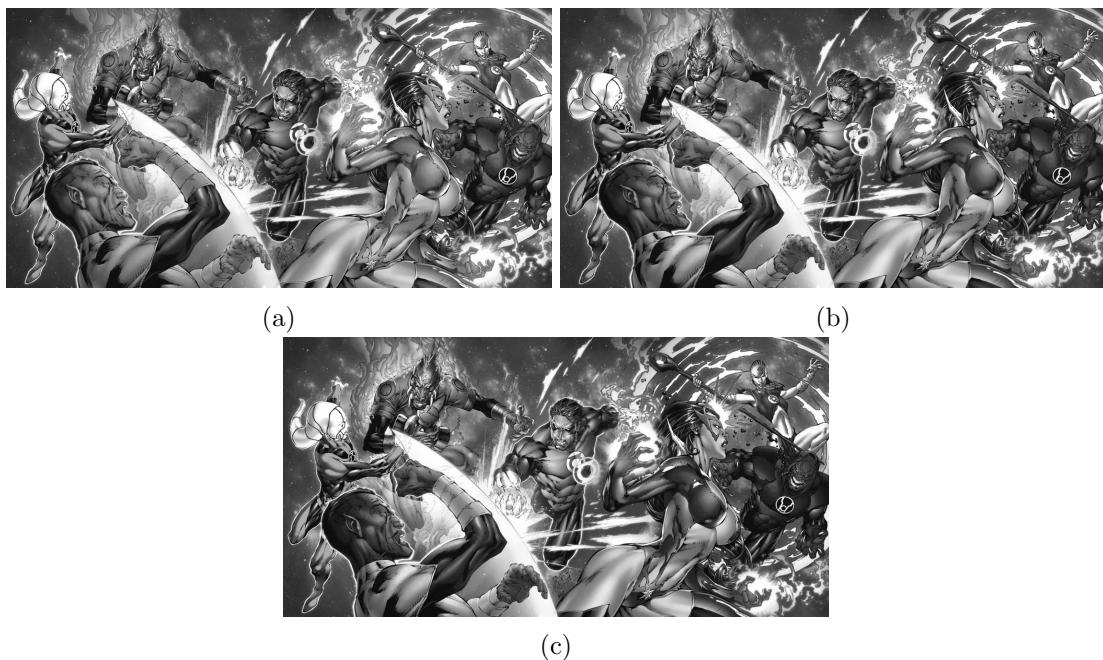


Figura 7: (a) Método de promedio (b) Método de promedio ponderado BT.601 (c) Método de promedio ponderado BT.709.

Conclusión

En conclusión, el presente reporte ha proporcionado una visión general de los procesos básicos para el manejo de imágenes RGB en el contexto de la visión robótica. Desde la apertura de imágenes hasta la conversión a escala de grises, se han explorado diferentes técnicas y métodos utilizando la biblioteca OpenCV en Python. La comprensión de estos fundamentos es crucial para el desarrollo de sistemas de visión artificial robustos y eficientes en aplicaciones robóticas. Además, la comparación de métodos para la conversión a escala de grises, como el promedio y los estándares ponderados BT.601 y BT.709, ha permitido apreciar las sutilezas en la representación de la luminosidad en imágenes digitales. Este conocimiento sienta las bases para el diseño y la implementación de algoritmos más avanzados en futuros proyectos de visión robótica.