

Práctica 1:

Manipulación de imágenes y dibujo con OpenCV

Manuel Alejandro Torrealba Torrealba, Alejandro Alemán Alemán.

Ejercicio 1a – Visualización de valores RGB del píxel bajo el cursor

En este primer ejercicio se desarrolló un programa en Python utilizando la biblioteca OpenCV (cv2), cuyo objetivo es cargar una imagen y mostrar el valor RGB (Red, Green, Blue) de cada píxel sobre el que se sitúe el cursor del ratón.

Para ello, se empleó la función `cv.setMouseCallback()` que permite definir una función callback asociada a los eventos del ratón (como movimientos o clics). Dentro de esta función, se obtiene la posición del cursor (x, y) y, mediante el acceso al array de la imagen (`img[y, x]`), se recuperan los valores B, G, R del píxel. Estos valores se muestran sobre una ventana específica utilizando `cv.putText()` para superponer el texto en la ventana.

El flujo principal del programa incluye:

- Carga de la imagen con `cv.imread()`.
- Creación de una ventana con `cv.imshow()`.
- Asociación del callback de ratón con `cv.setMouseCallback()`.
- Bucle de espera mediante `cv.waitKey(0)` hasta que el usuario pulse una tecla.

Finalmente, se implementó un cierre robusto de ventanas con:

```
cv.destroyAllWindows()  
  
cv.waitKey(1)
```

garantizando que las ventanas gráficas se cierran correctamente.

Aspectos destacables:

- Uso del evento `cv.EVENT_MOUSEMOVE` para capturar los movimientos del ratón.

- Conversión del formato BGR → RGB al mostrar los colores.
- Escritura del texto directamente sobre la imagen sin interferir con la visualización general.
- Estructura modular y fácil de ampliar.

Ejercicio 1b – Dibujo de primitivas con el ratón

En este segundo ejercicio se amplió el programa anterior para permitir dibujar distintas primitivas geométricas: líneas, rectángulos y círculos. Además, se añadió una interfaz gráfica que permite seleccionar el color (RGB) y definir el grosor de las líneas mediante trackbars (barras deslizantes) creadas con la función `cv.createTrackbar()`.

El funcionamiento principal se basa en:

- Detectar los eventos del ratón (`EVENT_LBUTTONDOWN`, `EVENT_LBUTTONUP`) para capturar los puntos inicial y final del dibujo.
- Dependiendo del modo seleccionado, aplicar las funciones:
 - `cv.line()` para dibujar líneas.
 - `cv.rectangle()` para rectángulos.
 - `cv.circle()` para círculos.

El color de las figuras se toma dinámicamente de los valores de las trackbars R, G y B, mientras que el grosor se obtiene de otra barra llamada Gross.

Aspectos destacables:

- Control interactivo de los parámetros de dibujo sin reiniciar el programa.
- Detección precisa de clics y arrastres para definir las figuras.
- Integración del color y grosor en tiempo real mediante interfaz gráfica.
- Uso de una variable global `mode` que permite cambiar el tipo de figura fácilmente.

Ejercicio 2 (Optativo) – Mejoras y extensiones del programa

Para la parte optativa se añadieron varias mejoras funcionales al programa base.

1. Figuras rellenas:

Se implementó la posibilidad de crear figuras rellenas seleccionando una opción en la interfaz. Cuando el usuario activa el modo de relleno, el grosor (thickness) se establece a -1, lo que en OpenCV significa que la figura se dibuja completamente sólida.

2. Nuevas primitivas:

Se incorporaron otras primitivas geométricas:

- Elipses mediante la función `cv.ellipse()`.
- Polilíneas y polígonos utilizando `cv.polylines()` y `cv.fillPoly()`.

Estas opciones permiten dibujar figuras más complejas, seleccionando varios puntos con el ratón y conectándolos en secuencia.

3. Grabación del dibujo en vídeo:

Una de las mejoras más destacables es la posibilidad de guardar el proceso de dibujo como un vídeo. Para ello se utilizó el objeto `cv.VideoWriter`, que permite registrar cada frame generado durante la sesión. De esta forma, al finalizar el dibujo, el usuario puede reproducir el vídeo y visualizar cómo se construyó la imagen paso a paso.

Conclusión general

En esta práctica se desarrolló una aplicación interactiva de dibujo con OpenCV capaz de:

- Leer y mostrar imágenes.
- Detectar movimientos y clics del ratón para obtener información o dibujar figuras.
- Personalizar el color, el grosor y el tipo de figura a dibujar.
- Extender la funcionalidad con figuras rellenas, elipses, polígonos y grabación en vídeo.

El resultado es un programa versátil y visualmente interactivo que demuestra el dominio de los eventos, interfaces y primitivas gráficas en OpenCV.