

MEMORIA PRÁCTICA 2: Transformaciones geométricas

Aris Vazdekis Soria & Alejandra Ruiz de Adana Fleitas

En esta práctica, desarrollamos una aplicación en Python que implementa transformaciones geométricas en imágenes, aplicadas en tiempo real mediante una interfaz con trackbars en OpenCV. La estructura del programa fue dividida en diferentes secciones para responder a cada pregunta ordenadamente, en las que se incluyen traslación, rotación y escalado de imágenes.

1a. Transformaciones Básicas

La primera parte consistió en implementar una interfaz que permitiera aplicar traslaciones, rotaciones y escalados a una imagen cargada en tiempo real:

- **Traslación:** A través del movimiento del ratón, el usuario puede desplazar la imagen de manera interactiva haciendo click izquierdo. Esta funcionalidad requirió capturar eventos de ratón y acumular las coordenadas de desplazamiento.
- **Rotación:** El programa permite ajustar el ángulo de rotación y el usuario puede definir el centro de rotación mediante un clic izquierdo en la imagen.
- **Escalado:** Implementamos escalado uniforme, utilizando trackbars para controlar el factor de escala.

1b. Proyección de Imágenes

Completamos esta funcionalidad para proyectar una imagen dentro de los cuatro puntos de destino seleccionados. Serán necesarios hacer cuatro clicks izquierdos en la pantalla para poder el programa procesar la venta proyectiva de la imagen.

1c. Distorsión de Lente

Para jugar con la distorsión de lente, implementamos trackbars para controlar los coeficientes de distorsión en tiempo real. Se añadieron dos opciones de distorsión, efecto de barril (tipo 0) y almohadilla (tipo 1). La interfaz permite observar en vivo los efectos visuales en una imagen con distintas configuraciones de distorsión.

Aportaciones Optativas

En esta practica, integramos funcionalidades adicionales que pese a que son optativas, decidimos implementar con el fin de comprender mejor el entorno de opencv y mejorar notoriamente la interacción y la funcionalidad:

- **Manipulación de video en lugar de imágenes estáticas:** Realizamos las transformaciones geométricas sobre cada frame de un video en vez de una imagen estática, lo cual fue útil para estudiar la aplicación de transformaciones en secuencias de video. Todas las funcionalidades optativas implementadas en el apartado 1a. Tuvimos ciertas dificultades en este apartado para comprender el concepto de fps, en donde el video mostraba solo el primer frame y resultado ser que no le dabamos tiempo de retraso entre cada frame, por lo que no se reproducia

correctamente. Esto fue arreglado añadiendo una espera de 30 milisegundos entre frame dando así una velocidad de reproducción de aproximadamente 30 frames por segundo (velocidad del video nativa).

```
# Salir si se pulsa la tecla Esc
if cv2.waitKey(0) & 0xFF == 27: # Esperar 30 ms entre frames
    break
```

- **Transformación Afín:** Permitimos seleccionar tres puntos en la imagen original y otros tres en una imagen destino para realizar una transformación afín, ajustando posiciones y escalas de manera precisa.
- **Calculo de imagen especular:** También calculamos la imagen especular gracias a la función de open cv flip() en donde le hicimos un flip en el eje horizontal.

La práctica fue una buena experiencia para aprender sobre transformaciones geométricas en OpenCV. Crear una interfaz interactiva y aplicar efectos en tiempo real fue muy útil para entender cómo funcionan traslación, rotación y escalado en imágenes y video. Tuvimos algunos retos, como sincronizar los fotogramas del video, pero logramos superarlos y aprender bastante en el proceso. Fue interesante y nos dejó con una mejor comprensión de cómo manipular imágenes de forma dinámica.