

Informe de práctica 1

Se comienza importando la librería de opencv que sirve para manejar capturas de video, mostrar imágenes, etc.

Adicional, importamos librería Math que usaremos para calcular la distancia euclidiana entre un punto (x, y) y el origen de coordenadas (0, 0) para obtener el punto arriba a la izquierda. Para el punto abajo a la derecha se usa la distancia menor entre un punto (x, y) y el tamaño máximo del frame.

$$distancia = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

La función **distancia** utiliza esta fórmula para calcula el punto más cercano y más lejano al ordenar en forma ascendente los todos los puntos dibujados.

La función clave es **save_frames**, encargada de recortar y almacenar una región de interés (ROI) dentro del frame de vídeo. Sus parámetros de entrada son el frame actual, las coordenadas de la zona a recortar (los puntos calculados), y el código necesario para el almacenamiento de la imagen. El control de la selección de la ROI se gestiona mediante un evento de clic izquierdo del ratón. Cada clic registra una coordenada, y el sistema utiliza un contador de puntos y una estructura de datos (objeto) para capturar los cuatro puntos necesarios para definir completamente el rectángulo de recorte.

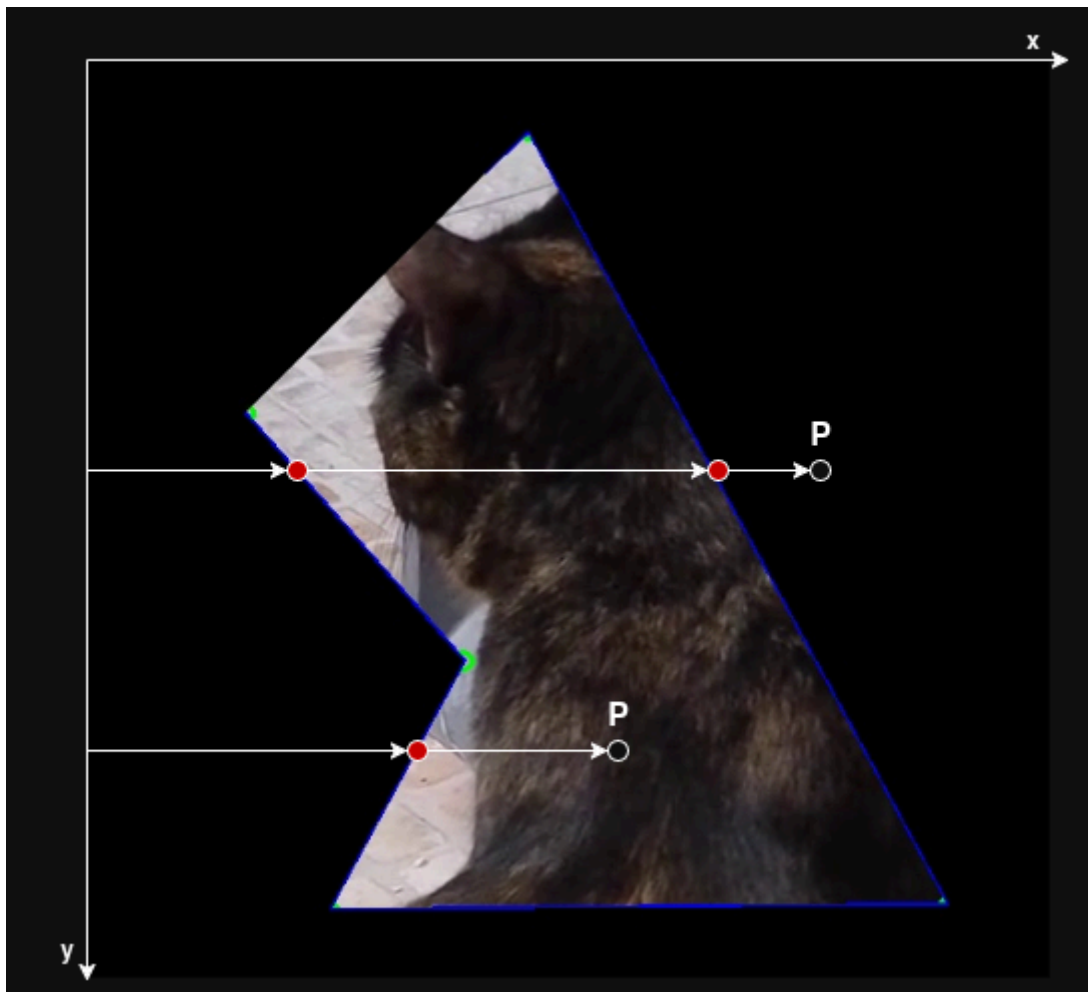
Se actualiza el callback del mouse y si se presiona la tecla **s** se ejecuta un hilo para almacenar las imágenes **frame_recortado_{code}.jpg** y **frame_with_lines_{code}.jpg** con los puntos y líneas dibujadas en el frame de la pantalla.

Extracción de la forma irregular

Se usa el concepto del algoritmo del rayo para calcular los puntos que se encuentran dentro de la figura irregular. Si para llegar a un punto se intersecciona un número impar de lados se considera que el punto está dentro del polígono. De la misma manera cuando se extrae la forma irregular se usa un hilo, al ser una tarea más pesada al calcular todos los puntos que se encuentran dentro de la figura.

Con esta ecuación se calcula la coordenada en **x** donde se cruza la línea con uno de los lados de la figura.

$$x_{intersección} = (point[1] - y_1) * (x_2 - x_1) / (y_2 - y_1) + x_1$$



El código está diseñado para recortar una región poligonal irregular del frame de vídeo. La función **validate_point**, que implementa el algoritmo del rayo para determinar si cada píxel individual está dentro del polígono definido por una lista de **vertices**. La función principal, **cut_irregular_frame**, aplica la validación de todos los píxeles del frame, manteniendo su contenido original y reemplazando todo lo demás con un fondo negro, logrando así un recorte de forma irregular.

Nota: Enlace del repositorio en GitHub

<https://github.com/militem/computervision>, el manual de uso se encuentra en el README.md