

Análisis de variables cinemáticas en tiempo real

Visión por ordenador

Marcos Garrido Ferrer
Santiago Urtiaga Pérez

Introducción

Este proyecto se puede dividir en tres módulos principales:

- El módulo de calibración, que permite al usuario calibrar la cámara Raspberry Pi, tomando imágenes durante la ejecución del programa.
- El módulo de seguridad, que sólo permite al usuario acceder al resto de funcionalidades del programa una vez escaneada la secuencia correcta de patrones dibujados.
- El módulo principal, que rastrea el movimiento de cualquier objeto en tiempo real; mostrando la gráfica de la evolución temporal de la variable deseada: velocidad o aceleración.

Este proyecto está basado en OpenCV y utiliza gráficos interactivos de Matplotlib para representar los gráficos.

Calibración de la cámara

Para capturar imágenes desde una cámara utilizando OpenCV y Picamera2, creamos un entorno que permite la captura de vídeo en tiempo real y la toma de fotografías mediante la interacción del usuario. Primero, se inicia la captura de vídeo desde la cámara y se muestra en una ventana. El sistema espera a que el usuario presione una tecla específica para capturar una imagen, la cual se guarda en un directorio predefinido.

Este proceso es útil, por ejemplo, para capturar imágenes de un tablero de ajedrez necesarias para la calibración de la cámara; aunque también es empleado en otras secciones del proyecto.

Para calibrar una cámara utilizando las imágenes capturadas previamente, se sigue un procedimiento que incluye la detección de puntos de referencia en las imágenes y el cálculo de los parámetros de la cámara. Este proceso es esencial para corregir las distorsiones ópticas y mejorar la precisión de las imágenes capturadas.

Lamentablemente, los parámetros de calibración no se guardaron correctamente. Además, al momento de redactar este informe, la Raspberry Pi y la Picamera ya habían sido entregadas, por lo que no fue posible volver a capturar y guardar los parámetros necesarios.

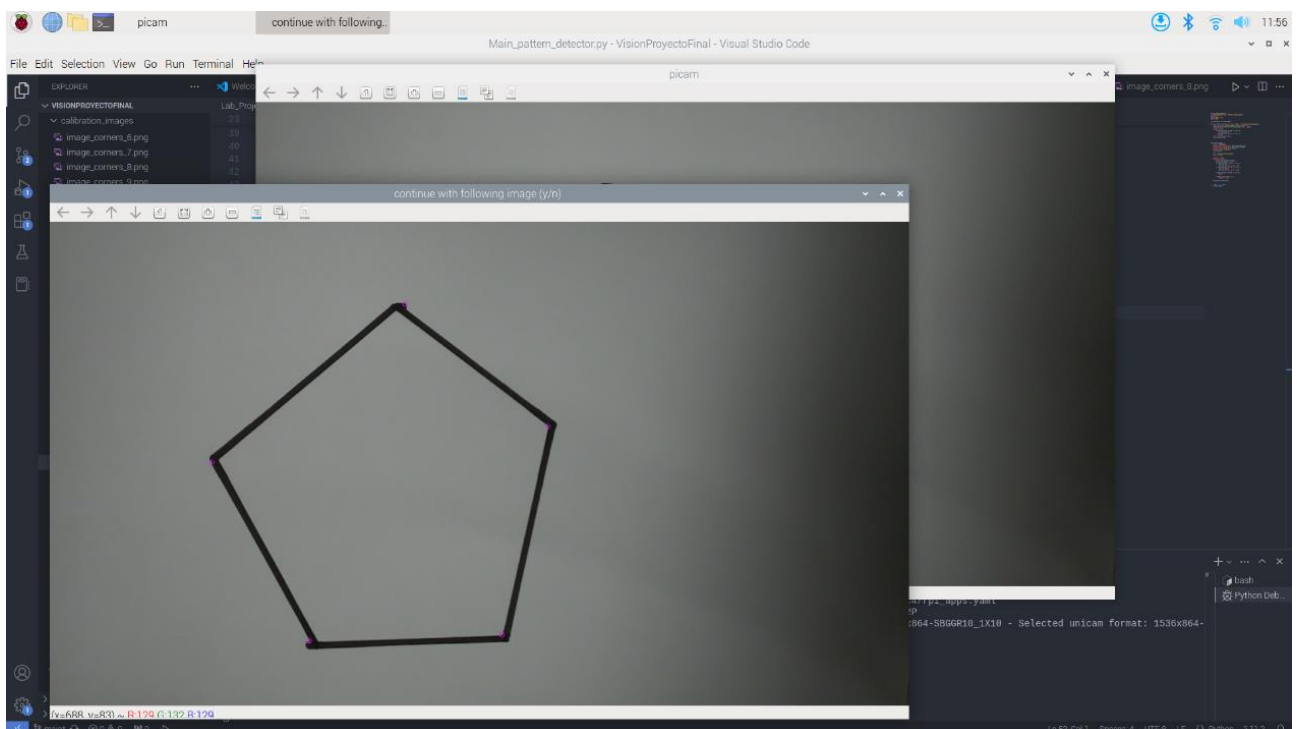
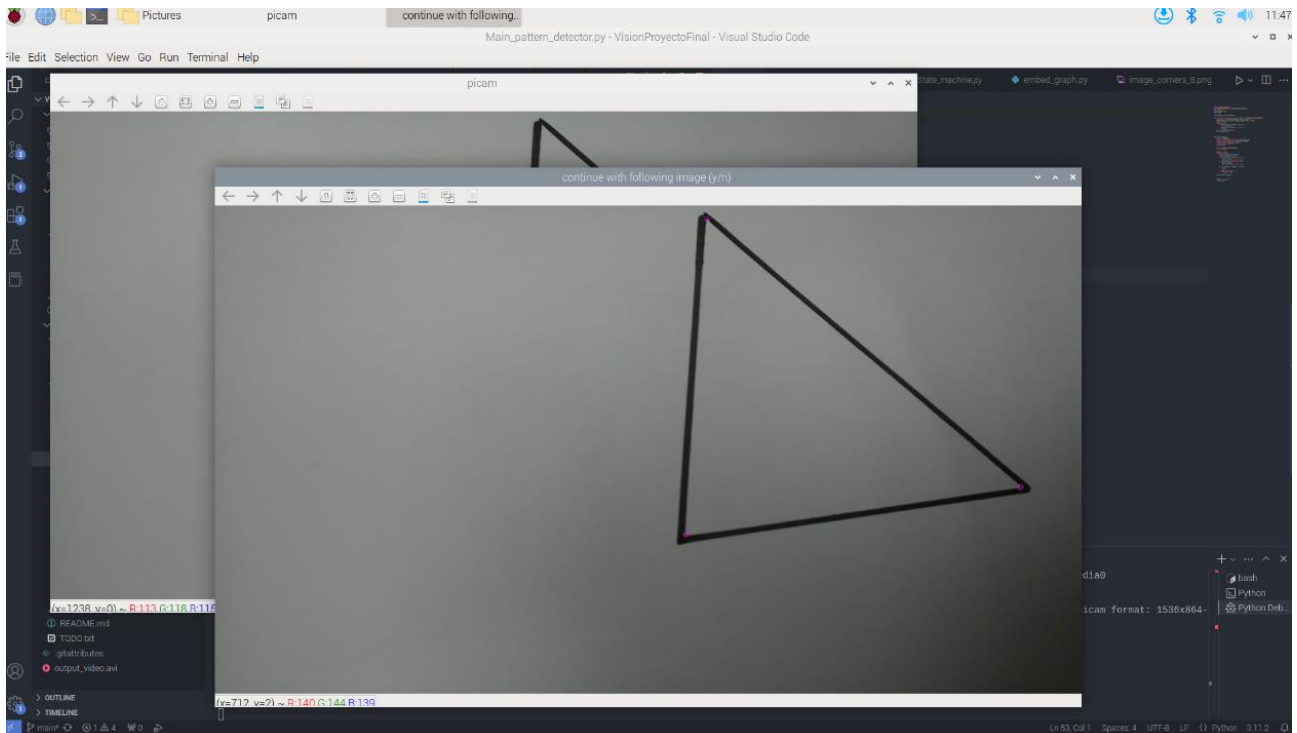
Módulo de seguridad

Cuando se ejecuta, la biblioteca Picamera2 comienza a capturar video en vivo desde Raspberry Pi. El vídeo se muestra en una ventana. En este estado, el programa espera que el usuario realice una secuencia de capturas de patrones para poder acceder al módulo principal. Esto se logra presionando

una tecla una vez que se muestra el patrón a la cámara. El usuario puede elegir el algoritmo de detección a emplear: Harris o Shi Tomasi.

- El método de Harris es un algoritmo clásico para la detección de esquinas que utiliza la matriz de autocorrelación para identificar puntos de interés en la imagen.
- El método de Shi-Tomasi es una mejora del método de Harris que selecciona las esquinas basándose en el valor mínimo de las respuestas de autocorrelación.

Una vez que el usuario elija uno de estos algoritmos, aparecerá otra ventana que mostrará el resultado del algoritmo de detección. Aquí hay dos ejemplos separados para diferentes patrones:



El usuario ahora podrá pasar al siguiente paso utilizando esta imagen o, por el contrario, repetir la captura. Actualmente, por simplicidad, la contraseña tiene tres dígitos, por lo que el usuario tendrá que repetir este proceso dos veces.

Para comparar las imágenes con la secuencia de la contraseña, se toma el número de esquinas detectadas en una imagen y determina el tipo de polígono correspondiente. Por ejemplo, un círculo o una línea no tienen esquinas, un cuadrado tiene cuatro esquinas, un triángulo tiene tres, y un pentágono tiene cinco.

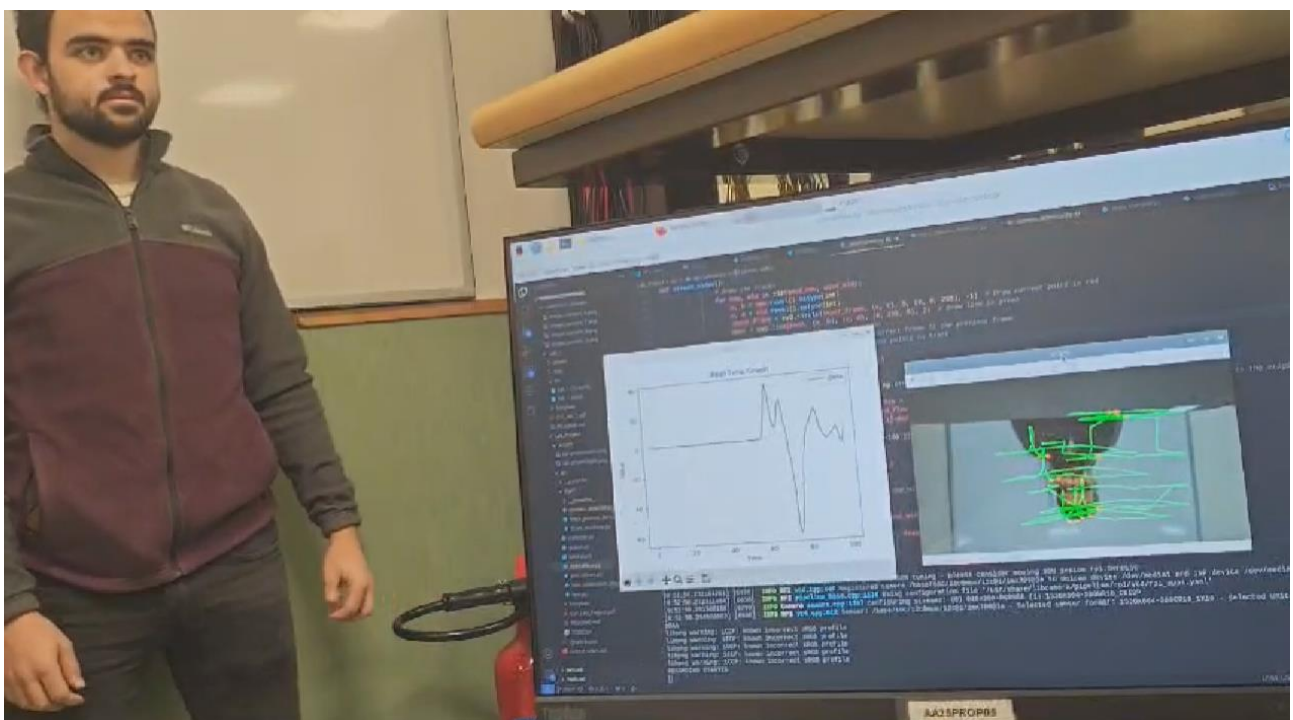
Una máquina de estados se implementa con un alfabeto, una función de estados y un diccionario de transiciones. Esta máquina de estados permite gestionar de manera eficiente la detección de polígonos y las transiciones entre estados según el tipo de polígono identificado. Si se introduce un polígono no válido, la máquina de estados vuelve al estado inicial.

Módulo principal

Este programa comenzará a mostrar la grabación en tiempo real de la Picamera una vez que se inicie. Una vez que la cámara esté quieta y orientada en la dirección correcta, el usuario puede presionar una tecla para iniciar el seguimiento. El usuario ahora puede elegir qué métrica mostrar en el gráfico: velocidad o aceleración.

Una vez realizada la elección, se realizará un seguimiento del objeto en movimiento en tiempo real. Además, aparecerá una ventana auxiliar con un gráfico de la métrica seleccionada. Este gráfico también se actualizará en tiempo real.

A veces, los puntos estáticos en la imagen se rastrearán por error. Sin embargo, se considerarán valores atípicos y no se tendrán en cuenta al calcular la velocidad y la aceleración del objeto en movimiento. Esta selección se realiza de forma natural: se tiene en cuenta la mediana de los flujos ópticos para determinar la velocidad y la aceleración del objeto en la pantalla. De este modo, varios valores atípicos no afectarán a la salida del gráfico.



En la demostración en vídeo, la cámara se colocó boca abajo para asegurar su estabilidad. Sin embargo, los aspectos más destacables que se pueden notar son:

- En esta prueba, el video y el gráfico tienen una frecuencia de actualización de 30 cuadros por segundo. Aunque el programa había demostrado funcionar a 50 fps, decidimos limitarlo porque el RPi empezó a calentarse notablemente a esa mayor velocidad de cuadros por segundo.
- El gráfico muestra la evolución temporal de la velocidad del alumno. Cuando se mueve hacia la derecha en la imagen (su izquierda), el gráfico muestra un valor superior a 0. Cuando se mueve hacia la izquierda, el valor es negativo. Cuando está parado, el valor es aproximadamente 0.
- Los primeros segundos del vídeo muestran un gráfico muy ruidoso. Este ruido es varios órdenes de magnitud menor que los valores que se muestran cuando se mueve (como se puede ver, cuando el alumno empieza a moverse, el ruido es tan pequeño en comparación que se desvanece). Este ruido puede atribuirse a pequeños movimientos en la imagen y su objetivo.

Conclusión

Creemos que este programa, con algunas modificaciones, se puede implementar en una amplia gama de situaciones:

- En un radar de carretera, para rastrear la velocidad de los coches que se aproximan.
- En deportes: para calcular la velocidad y la aceleración de los jugadores, los saques de tenis o los swings de golf.
- En la producción mecanizada, para controlar el correcto funcionamiento de los robots de fabricación