

Pontificia Universidad Javeriana



Facultad de Ingeniería

Ingeniería Electrónica

Procesamiento de Imágenes y visión

Proyecto Final: Detección de objeto por Matching Template

Profesor:

Francisco Carlos Calderón Bocanegra

Integrantes Grupo 10:

Nicolás Alberto Bravo Silva

Nicolás Enrique Rodríguez Villate

1. Introducción:

Nuestro proyecto consiste en diseñar e implementar un código funcione parecido a la herramienta de trabajo en física Tracker, que sea capaz de identificar un objeto en un video con la herramienta de Matching Template que es una técnica en el procesamiento de imágenes digitales para encontrar pequeñas partes de una imagen que coincidan con una imagen de plantilla. Se puede utilizar en la fabricación como parte del control de calidad, una forma de navegar por un robot móvil o como una forma de detectar bordes en imágenes. Los principales desafíos en la tarea de coincidencia de plantillas son: oclusión, detección de transformaciones no rígidas, cambios de iluminación y de fondo, desorden de fondo y cambios de escala.

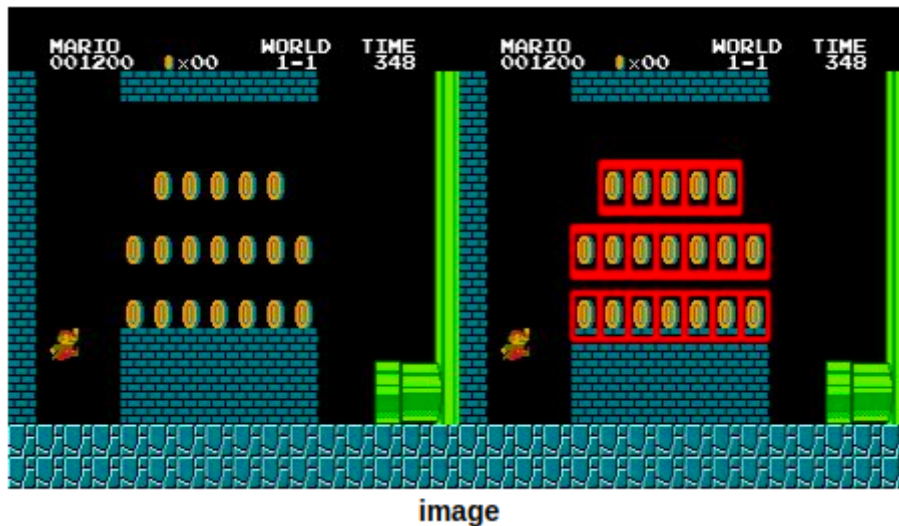


Imagen 1 Matching Template ejemplo

Además de esto, también se utilizó el método de contornos, ya que es otro método muy útil a la hora de encontrar objetos; Un contorno es un conjunto de puntos que conectados unos con otros de manera consecutiva forman una figura que rodea un objeto determinado, la detección de contornos en OpenCV se aplica sobre imágenes binarias, la figura obtenida puede ser analizada posteriormente para determinar cuál es el objeto que hemos detectado, para obtener los contornos utilizaremos la función `findContours()`, mientras que `drawContours()` nos ayudará a dibujarlos.

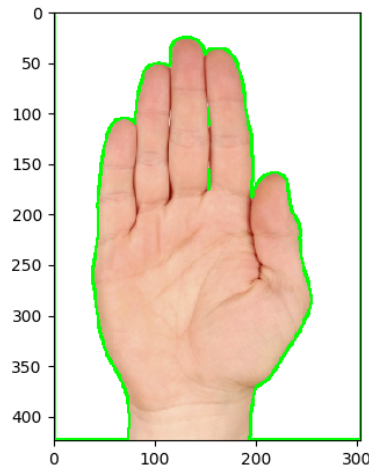


Imagen 2 Ejemplo contornos opencv

2. Desarrollo y diagrama de flujo:

Se realiza un diagrama general del sistema, en el cual se describen los procesos mas importantes de este, y luego se describe cada subproceso por separado, se realizan de manera muy general, con el objetivo de aclarar la forma en la que se resuelve el problema planteado.

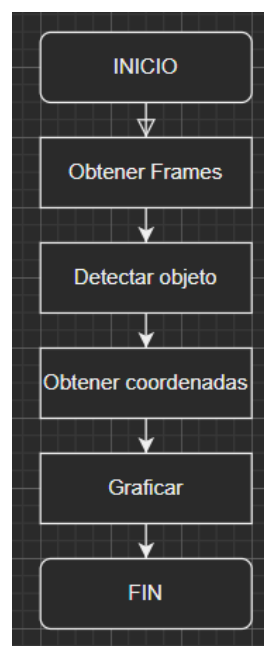


Imagen 3 Diagrama de flujo general del sistema

Obtener frames: En este espacio se toma el video escogido para analiza, y este video se dividirá en frames, para lograr tener acceso a cada fotograma del video, esto se realiza con opencv, ya que tiene una función que nos f acilita este procedimiento, los frames entregados son almacenado en una carpeta para su posterior uso.

Detectar objeto: En este espacio utilizamos un método de encontrar coincidencias en una imagen, conocido como matching template, con el cual tenemos que dar la oportunidad al usuario de que seleccione en una imagen el objeto que desea analizar, este objeto será almacenado como la coincidencia que el programa tratara de encontrar en cada fotograma del video propuesto, el programa tomara todos los frames obtenidos anteriormente y compara la imagen a analizar, para cada frame en el caso de que coincida se ubicara un rectángulo alrededor del objeto, si no hay coincidencia simplemente se pasa al siguiente frame, cabe aclarar que guardaremos los nuevos frames en una nueva carpeta.

Obtener coordenadas: En este espacio analizaremos los frames del video luego de pasar el método de matching template, es este caso ubicaremos el color amarillo (el cual es el color escogido para el rectángulo que se coloca luego de usar matching template), y hallaremos posteriormente los contornos, luego escogeremos los contornos cuya área es mayor modificando esta misma hasta asegurar que solo se toma uno de los dos contornos generados por el rectángulo, esto para luego encontrar el punto central de la coincidencia y su ubicación en la imagen, posteriormente se guardan los datos en dos archivos .txt nombrados como datos_x, datos_y, respectivamente.

Graficar: para graficar se utiliza la librería matplotlib, y para leer el archivo txt con los datos se utiliza pandas.

3. Resultados:

Para los resultados, se prueba el código de sacar frames(Sacar_frames.py), con el cual generamos una carpeta la cual contiene cada frame de video y además nos entrega los datos de la duración y los fotogramas, los cuales nos servirán al momento de graficar.



Imagen 4 Frame extraído del video original

Luego realizamos el mapeo de cada frame (Object_detection.py) con el fin de encontrar la coincidencia, se debe haber elegido ya la imagen a buscar, para nuestro caso se escogió la bolita.

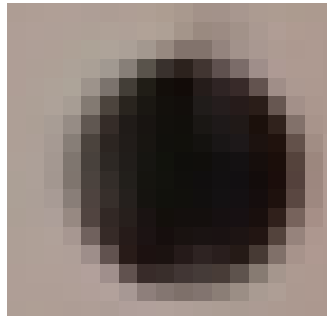


Imagen 5 Template

Se ejecuta el código y obtenemos que para cada frame ubicamos la coincidencia de la siguiente forma

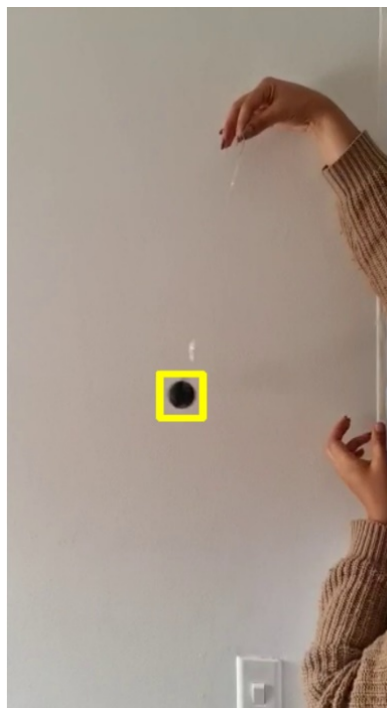


Imagen 6 Muestra de la manera como salen los frames luego de ejecutar el método de Matchig Template

Ahora se aplica el método de contornos mediante la utilización del código (centro.py), con el cual aplicamos el método de contornos para lograr encontrar el centro del 'template' y de esta forma lograr ubicar un punto más preciso al momento de graficar, mediante la utilización de este código obtenemos lo siguiente:

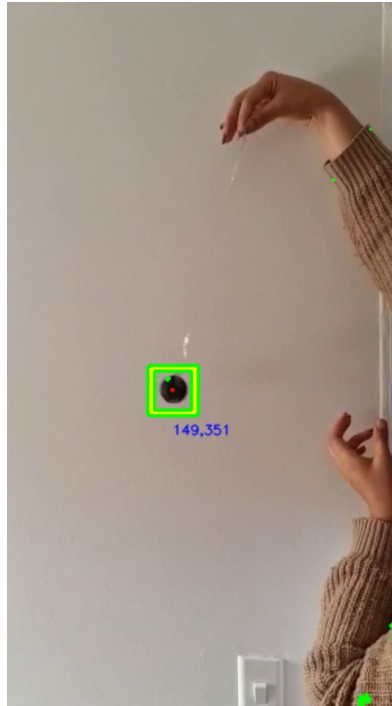


Imagen 7 Muestra de la manera como salen los frames luego de ejecutar el método de Matchig Template seguido de contornos

Como se puede observar se encuentra el punto medio de manera precisa, además se observa la aparición de varios contornos los cuales logramos evitar ya que solo tomamos el mayor contorno, con el fin de no repetir procesos o poner más puntos de los necesarios, se comprobó además que de los dos contornos que genera el borde amarillo obtenido con matching template, se toma solo, en esta parte generamos documentos de texto donde se encuentran los datos de posición en X y Y, para cada frame.

Con los documentos generados en el paso anterior, los leemos con la librería pandas, y asignamos estos valores a dos variables respectivamente, y luego graficamos con matplotlib obteniendo lo siguiente

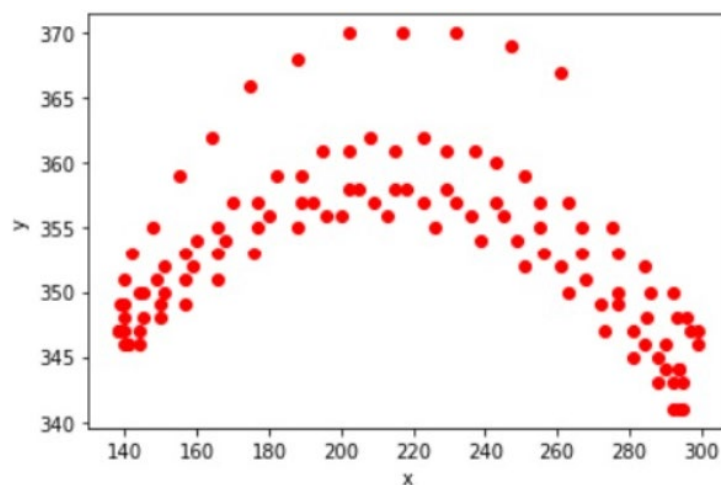


Imagen 8 Grafica de coordenadas de posición

4. Conclusiones:

- Se evidencio que el uso conjunto del método de matching template y contornos, genera unos resultados bastante aceptables ya que al observar los frames luego de pasarlos por estos dos métodos demuestran unos resultados con bastante precisión.
- Se decidió usar los dos métodos, ya que matching template nos ofrecía una buena base para lograr lo que queríamos hacer, pero no encontramos una manera de encontrar el punto medio que nos dejaba el cuadro de coincidencia, por lo cual se optó por usar uno de los métodos vistos en clase, el cual es el método de contornos, más específico el método de contornos en colores, en el cual ya habíamos realizado un trabajo similar, por ende partiendo de este concepto visto se logra adecuar para conseguir nuestro objetivo.
- Tuvimos problemas al momento de agregar el tiempo en la gráfica, pero de igual manera se dejó planteado ya que solo sería necesario generar un vector de tiempos del mismo tamaño que el vector de posiciones, donde cada valor corresponda al tiempo en el que se ejecuta un fotograma del video.
- Se deja abierta la posibilidad de modificar un poco mas el proyecto, ya que se tiene la base, pero no sería de mucha utilidad para un usuario final que no tenga conocimientos básicos en Python.