Car Racing PDI

FACULTAD DE INGENIERÍA
UNIVERSIDAD DE ANTIOQUIA
Departamento de Ingeniería de Sistemas

Semestre 2019-2

Leon Dario Arango Amaya
leon.arango@udea.edu.co
Jose Alberto Arango Sánchez
jose.arangos@udea.edu.co

Descripción del Problema

1. Problema humano a resolver:
    a. ¿Cómo podemos aprovechar los otros dispositivos de entrada y salida(cámara) para generar interacción entre un juego y el entorno físico del usuario?

2. Problema técnico a resolver:
    a. Si queremos generar interacción entre un juego y el usuario utilizando el color y la posición de un objeto, ¿Cuáles técnicas de procesado de imágenes podemos implementar para reconocerlos?

3. Antecedentes en trabajos similares
    i. (2014, septiembre 6). PyGame tutorial series - Python Programming Tutorials.
    ii. (2012, abril 25). Vison artificial con Python, pygame y OpenCV 4
    iii. (n.d.). Color Detection For Interaction Mobile Game Using OpenCV.

Marco teórico de las técnicas a emplear

- Técnica 1: Umbralización

```
# Convert BGR to HSV
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
# define range of blue color in HSV
lower_blue = np.array([110,50,50])
upper_blue = np.array([130,255,255])
# Threshold the HSV image to get only blue colors
mask = cv2.inRange(hsv, lower_blue, upper_blue)
# Bitwise-AND mask and original image
res = cv2.bitwise_and(frame,frame, mask= mask)
```

Marco teórico de las técnicas a emplear

Técnica 2: Detección de contornos

Código

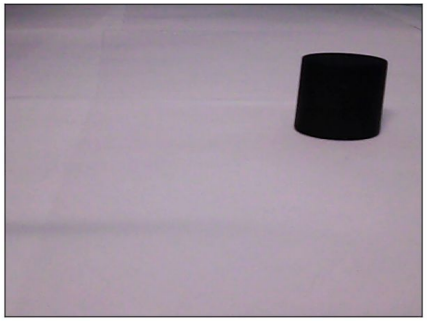
contours, hierarchy = cv.findContours(thresh, cv.CV_RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)

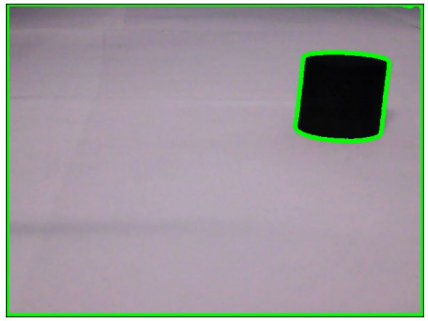
Parámetros

1. Imagen
2. Modo de recuperación de contorno
3. Método de aproximación de contorno

Retornos

1. lista de Python de todos los contornos de la imagen. Cada contorno individual es una matriz Numpy de coordenadas (x, y) de puntos límite del objeto.





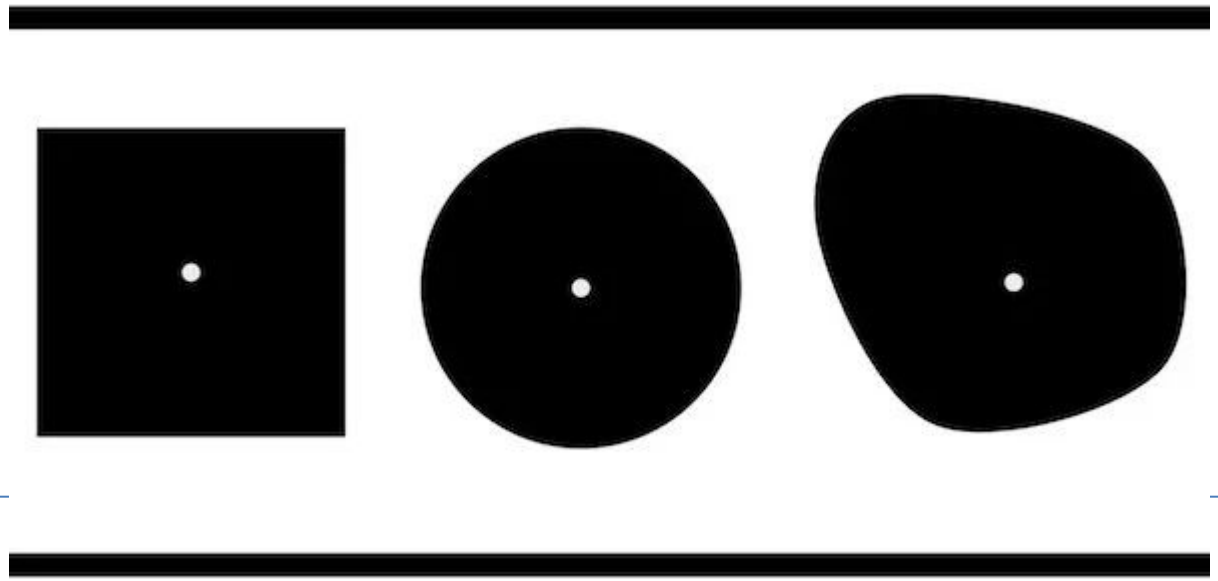
Marco teórico de las técnicas a emplear

Técnica 3: Calcular centros

Image moments: Un momento de imagen es un cierto promedio ponderado particular (momento) de las intensidades de los píxeles de la imagen.

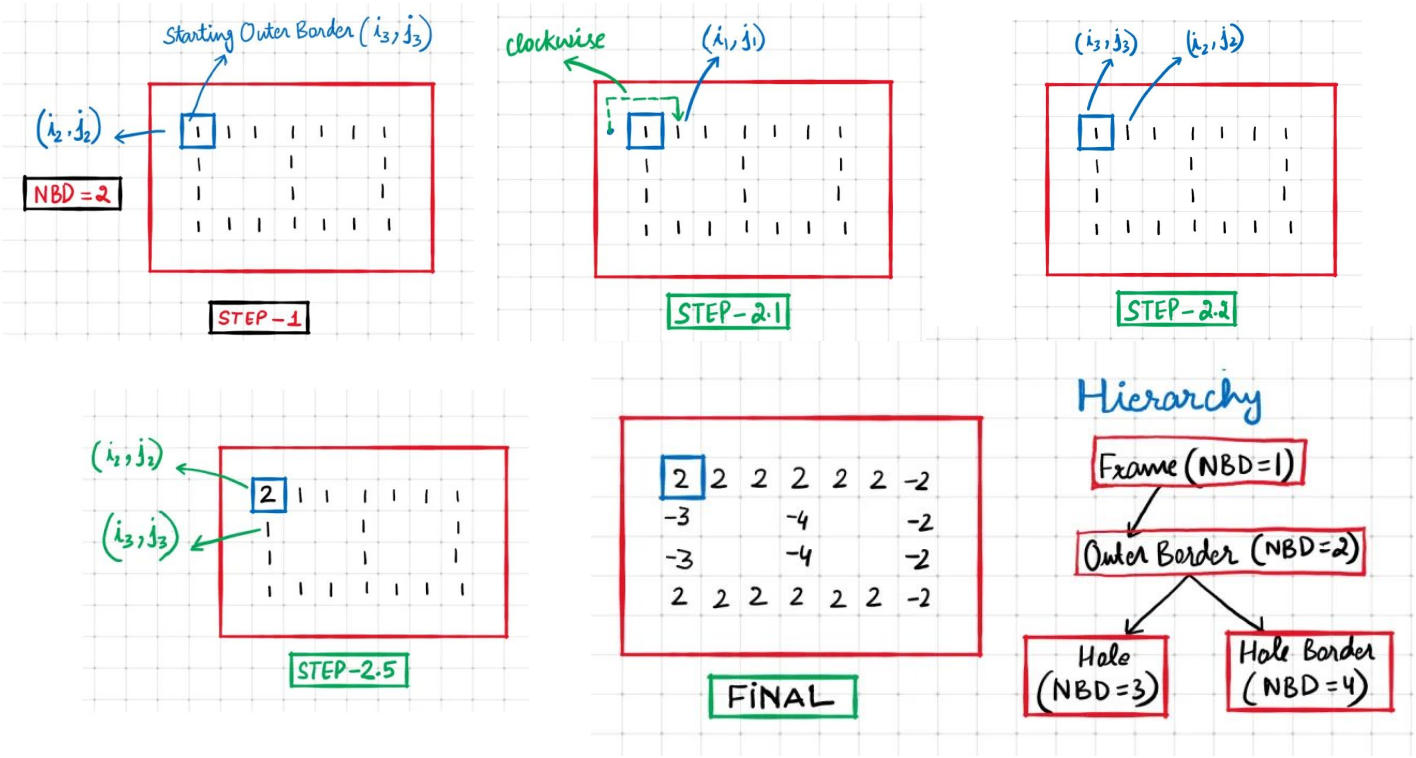
M = cv.moments(cnt)
cx = int(M['m10']/M['m00'])
cy = int(M['m01']/M['m00'])

$$\{\bar{x},\ \bar{y}\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\}$$



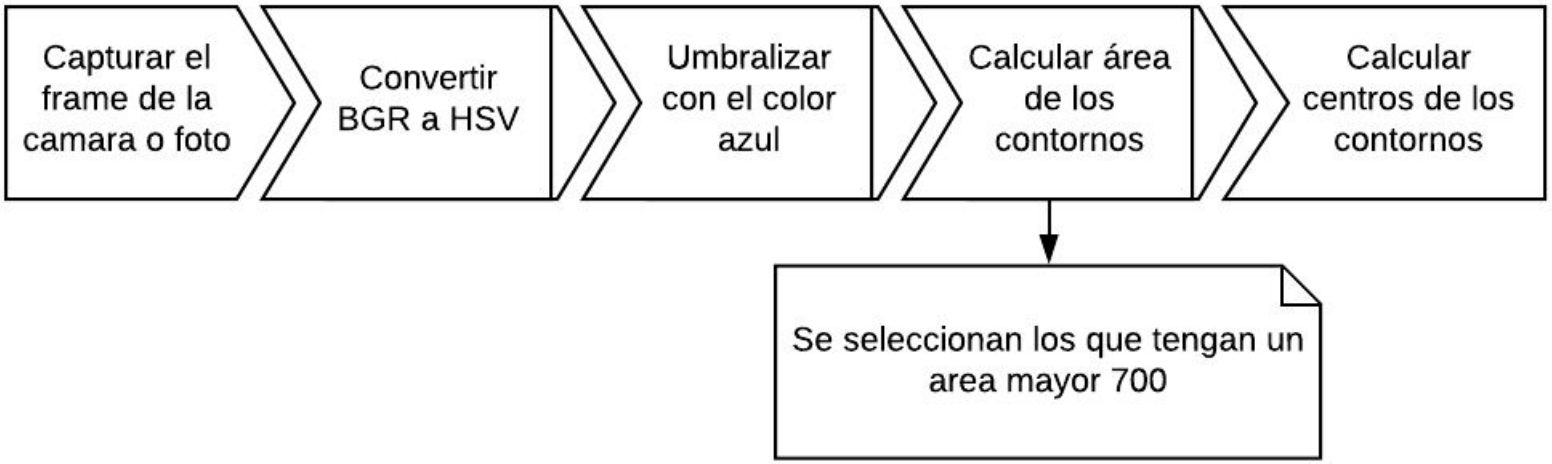
Algoritmos de prueba demostrativos

OpenCV utiliza el algoritmo Suzuki85

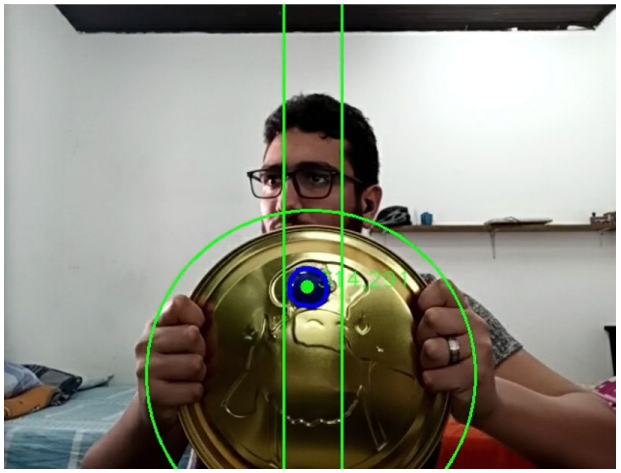


Propuesta de Solución

Conceptos del procesado de imágenes que se utilizaron: Umbralización, Detección de contornos, Convex Hull.

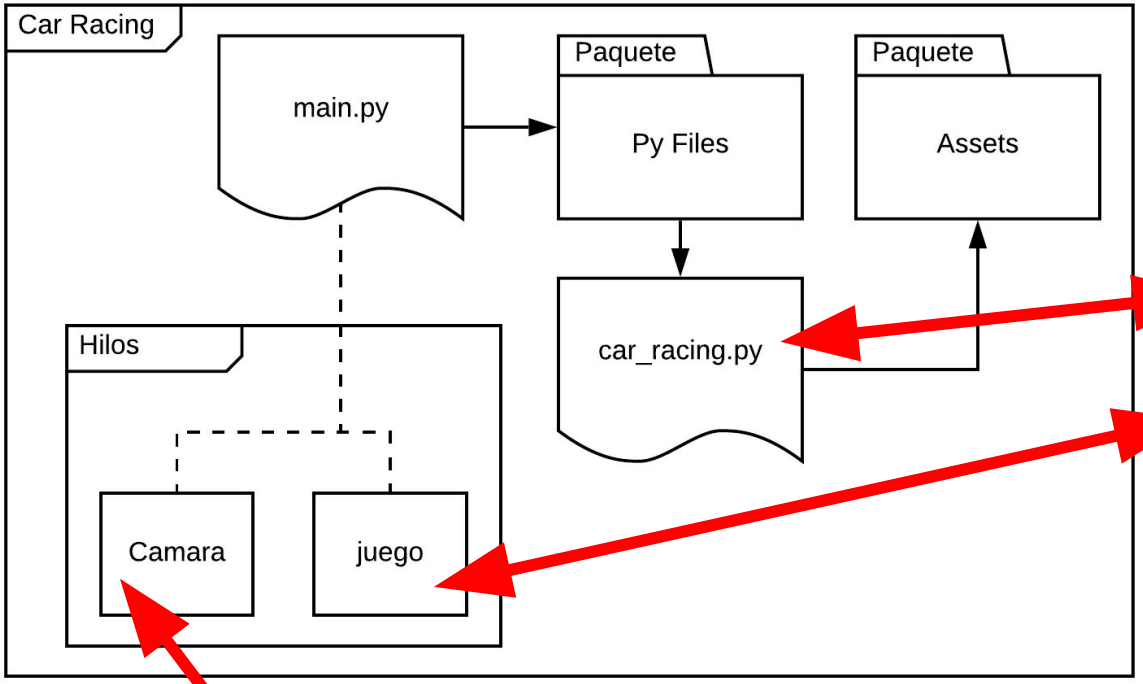


Propuesta de Solución



Estructura del Código

El código es simple, son dos hilos, ya que la cámara debe correr independiente al juego para no bloquearlo. Estos hilos se comunican a través de una variable global.

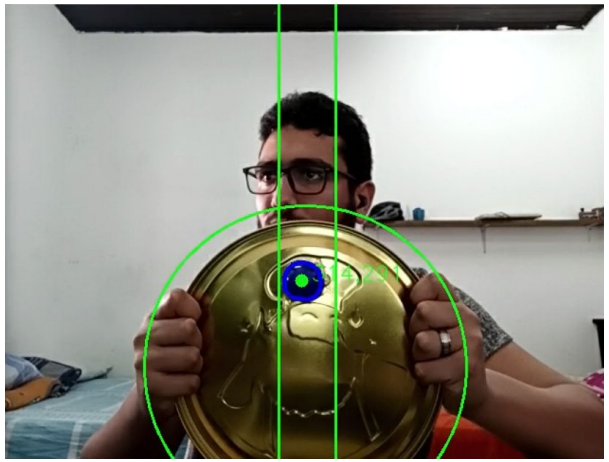


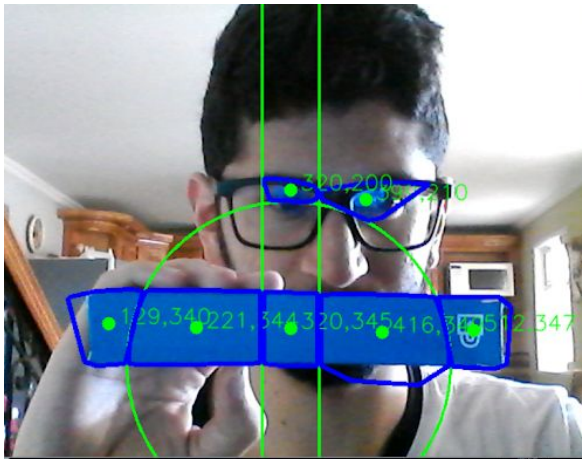

```
def main():
    car.intro()
    car.gameloop()
if __name__ == "__main__":
    main()
```

```
threadCamera = threading.Thread(target=openCamera)
threadCamera.start()
```

Resultados, Líneas Futuras

- Como resultado podemos observar que nuestro algoritmo detecta la posición y el color del objeto de forma correcta.





- Líneas futuras:
  - Un algoritmo que logre detectar el objeto con mayor precisión.
  - Optimizar los hilos del juego para más interacción, como un palanca de cambios.
  - Lograr detectar más colores para poder hacer un menú más interactivo.

Bibliografía y webgrafía

1. (n.d.). Changing Colorspaces — OpenCV 3.0.0-dev documentation. Recuperado el abril 7, 2020, de https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html
2. Contour Features - OpenCV. Recuperado el abril 7, 2020, de https://docs.opencv.org/trunk/dd/d49/tutorial_py_contour_features.html
3. (n.d.). OpenCV: Contours : Getting Started - OpenCV documentation. Recuperado el abril 7, 2020, de https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html
4. (2019, diciembre 31). Structural Analysis and Shape Descriptors — OpenCV 2.4 .... Recuperado el abril 7, 2020, de https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html
5. Image moment - Wikipedia. Recuperado el abril 7, 2020, de https://en.wikipedia.org/wiki/Image_moment
6. Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985).
7. (n.d.). pygame.display. Recuperado el abril 7, 2020, de https://www.pygame.org/ftp/contrib/pygame_docs.pdf
8. (n.d.). Sounds and Music with PyGame - PythonProgramming.net. Recuperado el abril 7, 2020, de https://pythonprogramming.net/adding-sounds-music-pygame/

Repositorio del proyecto: https://github.com/josearangos/car_racing_PDI

Preguntas?

Aplausos?

...

Marco teórico de las técnicas a emplear

- Técnica 4: Convex Hull

Matriz que especifica el polígono más pequeño que puede contener la región. Cada fila de la matriz contiene las coordenadas x y y de cada uno de los vértices del polígono.

Tomado de: Extracción de Características RegionProps