Ejercicio 4 – Bordes y regiones

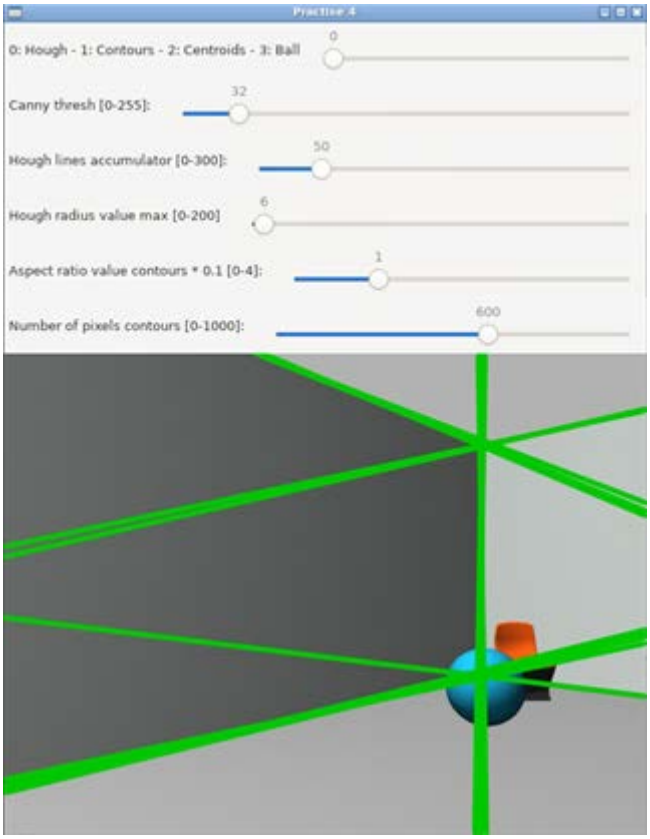
Este ejercicio tiene como objetivo aplicar los conceptos aprendidos en el Tema 5: Bordes y regiones.

La defensa del ejercicio se hará en clase, y hay que entregar un archivo **cv_node_p.cpp** con el código generado deberás subir al Aula Virtual.

Puntos totales posibles del ejercicio: 10

Instrucciones

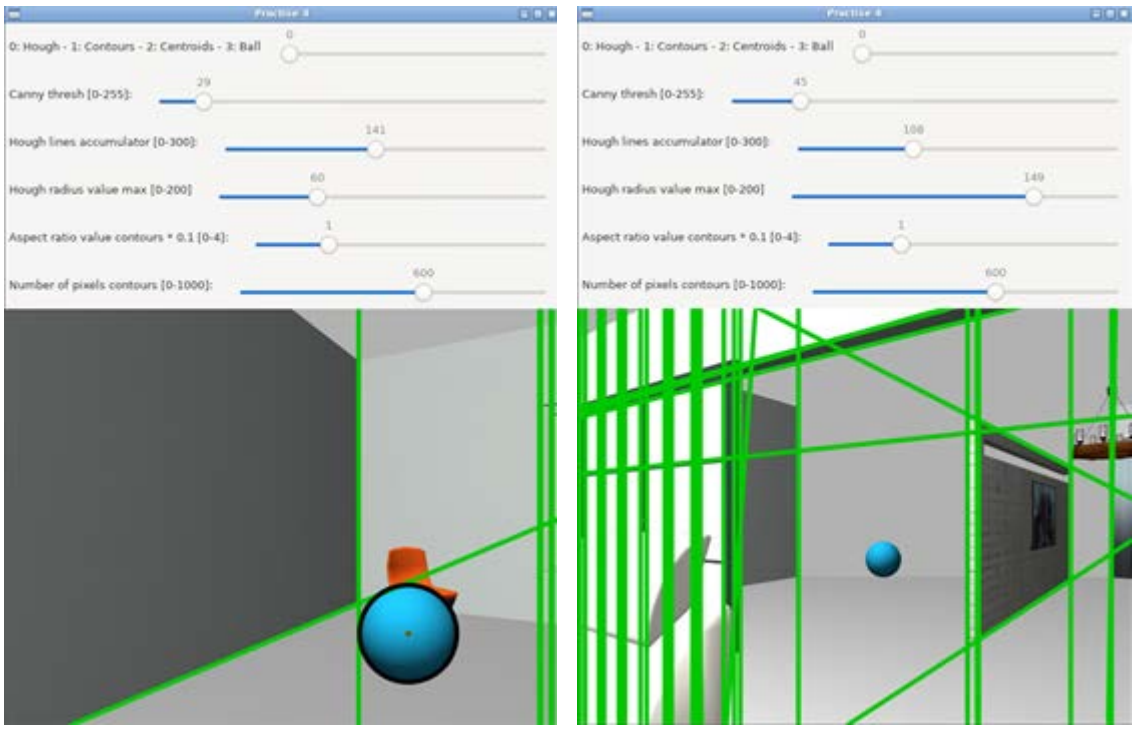
Utilizando el simulador con Tiago, se pide crear un programa que trabaje con la imagen visualizada y muestre en la parte superior varios sliders como los de la figura.



Dentro de la llamada al Callback de los sliders, o en una función aparte, se pide que se realicen diferentes tratamientos.

En primer lugar, habrá un **slider para controlar el umbral de Canny** que irá de 0 a 255, siendo el primer umbral de histéresis, el elegido mediante el slider, y el segundo umbral de histéresis será el doble del primero. Una vez hecho esto, sobre la imagen de Canny se podrá **elegir mediante otro slider el tratamiento a realizar**:

1. **Hough**: Se aplicarán en primer lugar una transformada de Hough estándar con el fin de detectar líneas. El **acumulador** de puntos estará controlado con un slider, que nos permitirá elegir entre 0 y 300 la cantidad de puntos que deben existir para considerarse línea. En segundo lugar, se aplicará una detección de Hough circular, de manera que el **radio** de detección podrá variarse entre 0 y 200 con otro slider. Ambos resultados deben de ser mostrados sobre la imagen de original.

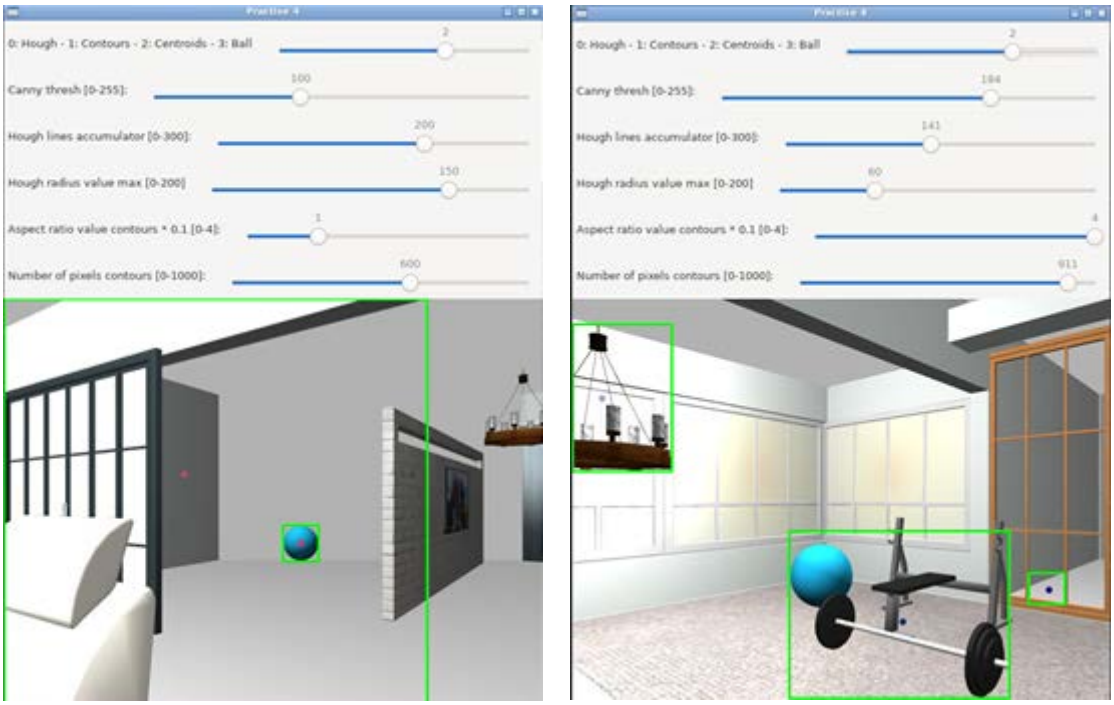


2. **Contornos**: Sobre la imagen de Canny se extraerán los contornos que aparecen en la imagen, pero únicamente se mostrarán aquellos cuya **cantidad de puntos** sea mayor que el valor modificable a través de un slider que irá de 0 a 1000, y además, estos contornos tienen que tener una relación ancho/alto (aspect **ratio**) similar, es decir, la diferencia en valor absoluto entre 1 (máxima coincidencia) y el ratio calculado sea menor que 0.1 * **valor**. Este **valor** será modificable a través de un slider en el intervalo 0-4, y los resultados se mostrarán sobre la imagen original.

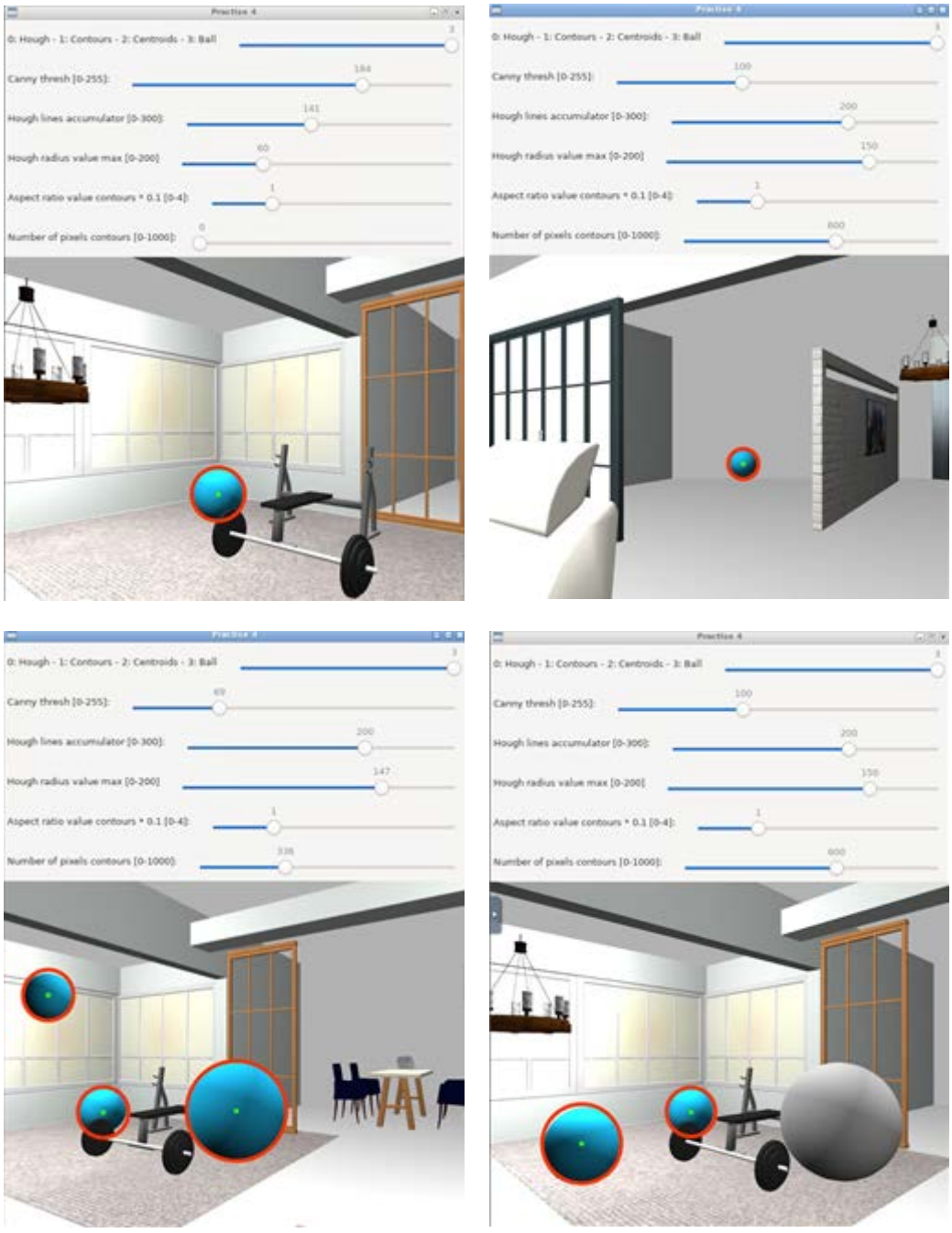
3. **Centroides**: Sobre la imagen de Canny se obtendrán los centroides de los contornos detectados en el punto anterior, para ello, es necesario **calcular los momentos** de los contornos y su centroide a partir de la siguiente ecuación:

$$C_x = \frac{M_{10}}{M_{00}}, \qquad C_y = \frac{M_{01}}{M_{00}}$$

Se pintarán los **centroides** de cada contorno que cumpla las condiciones de un **color diferente**, para ello se dibujará un **punto con radio 4**. Y además, se dibujará el bounding box de cada contorno en color verde.



4. **Pelota**: Utiliza un filtro de color y la detección circular de Hough para detectar únicamente la pelota azul. No hay límites en cuanto a la cantidad de píxeles, tamaño del círculo, etc. Queda a vuestro criterio diseñar un algoritmo que detecte la pelota lo más fiable posible independientemente de su tamaño, del número de pelotas, o de su oclusión con otros objetos. Hay que dibujar tanto el contorno como su centro.



Ayuda

Propiedades de los contornos:
https://docs.opencv.org/3.4/d1/d32/tutorial_py_contour_properties.html

Características de los contornos:
https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html