

Ejercicio 5 – Operaciones morfológicas y Calibración

Este ejercicio tiene como objetivo aplicar los conceptos aprendidos en el Tema 6: Operaciones morfológicas y Tema 7: Calibración.

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.



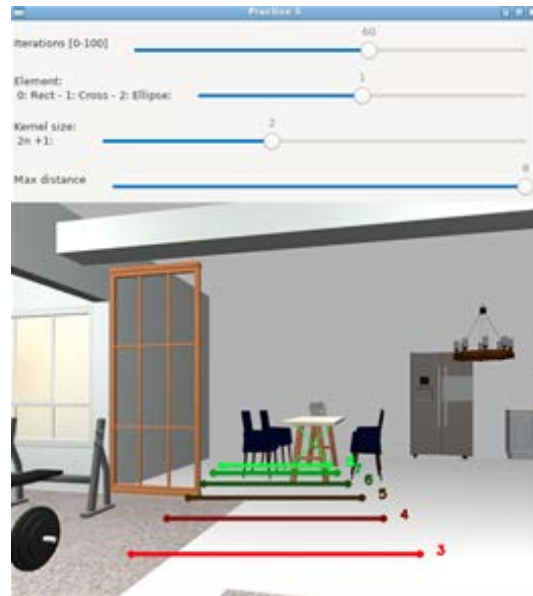
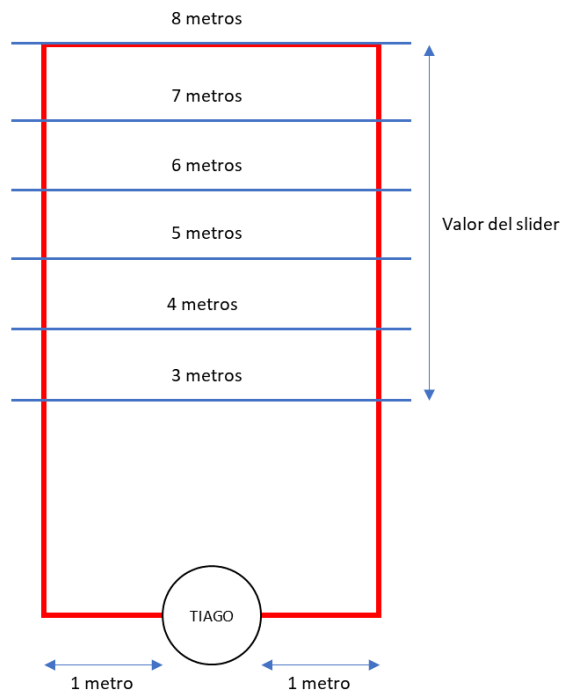
Dentro de la llamada al Callback de los sliders, o en una función aparte, se pide que se realicen diferentes tratamientos.

Proyección de líneas 3D a 2D:

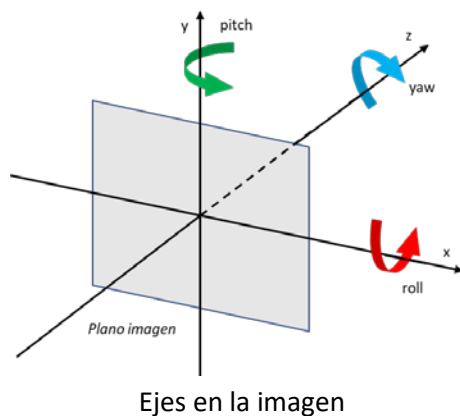
En primer lugar, se pide que se proyecten líneas de diferentes colores distanciadas 1 metro entre ellas en la realidad (mundo 3D), hasta el valor máximo seleccionado en el slider de distancia máxima. Este slider irá de 0 a 8, y junto a cada línea se indicará el valor de la distancia en ese punto.

Esta zona tendrá además 2 metros de ancho, es decir, un metro a cada lado del Tiago, y estará situada a los pies de este, por lo que cada línea nos indicará la distancia de los objetos situados en el suelo en dicho punto.

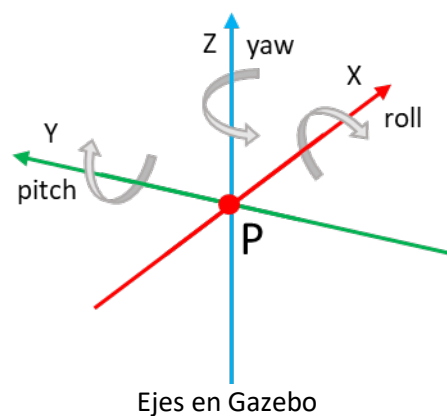
Para ello, puede verse el esquema del mundo 3D en la siguiente figura, junto al resultado final.



Hay que tener en cuenta que el sistema de coordenadas de la cámara difiere del sistema de coordenadas del simulador Gazebo (ver imagen), y que una unidad en el eje z de la imagen (Y en Gazebo), equivale a 1 metro en la realidad.



Ejes en la imagen



Ejes en Gazebo

Para calcular la transformada de 3D a 2D, necesitamos conocer los **parámetros intrínsecos** y **extrínsecos** del sistema para generar las matrices correspondientes.

Para valores de los **parámetros intrínsecos**, hay que observar la **matriz de proyección** que nos facilita el simulador a través de los siguientes comandos vistos en clase:

1. Ver el tipo del mensaje enviado en ROS para la información de la cámara:

```
ros2 topic info /head_front_camera/rgb/camera_info
```

- Una vez sabemos el tipo del mensaje, utilizaremos el siguiente comando para ver la ayuda del mensaje y observar los parámetros que nos interesaría ver:

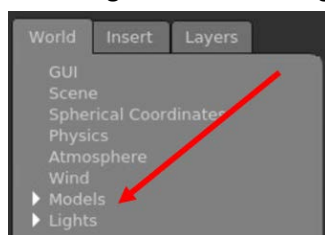
```
ros2 interface show <msg>
```

- Observadas las variables qué necesitamos, tenemos que ejecutar el siguiente comando para ver qué valores tiene la cámara del simulador y así poder generar la **matriz K** correspondiente a los **parámetros intrínsecos** del sistema.

```
ros2 topic echo <topic>
```

Para la obtención de los **parámetros extrínsecos**, es necesario ir al simulador de Gazebo.

- Una vez en él, debemos seleccionar la pestaña **Wolds** situada arriba a la izquierda, y dentro del desplegable de **Models** encontraremos el correspondiente al **Tiago**, en el cuál sabemos que en su cabeza tiene montada la cámara.
- Una vez tenemos localizado el modelo, veremos que hay dos partes que hace referencia a la cabeza. Seleccionándolas podemos ver resaltado sobre el Tiago cuál pertenece a la parte donde está la cámara. Una vez localizada la parte que nos interesa, debemos ver su posición en el mundo 3D, esto lo podemos encontrar en la información de abajo en la izquierda dentro del desplegable **pose**.
- Para la generación de la **matriz de parámetros extrínsecos**, matriz de rotación **R** y de traslación **t**, es necesario tener en cuenta qué eje del simulador Gazebo se corresponde con qué eje en el plano de la imagen, de este modo podemos poner la **altura, desplazamiento e inclinación** de la cámara en el eje que corresponda.
- En cuando a los ángulos de inclinación, observar que la nomenclatura (**roll, pitch, yaw**) coindice en ambos sistemas, pero las rotaciones no son las mismas. Si la rotación en un eje es al revés, debemos cambiar el sentido, y además se debe hacer una conversión de **radianes a grados**, ya que las rotaciones en Gazebo están dadas en radianes y en el plano de la imagen se desean en grados.

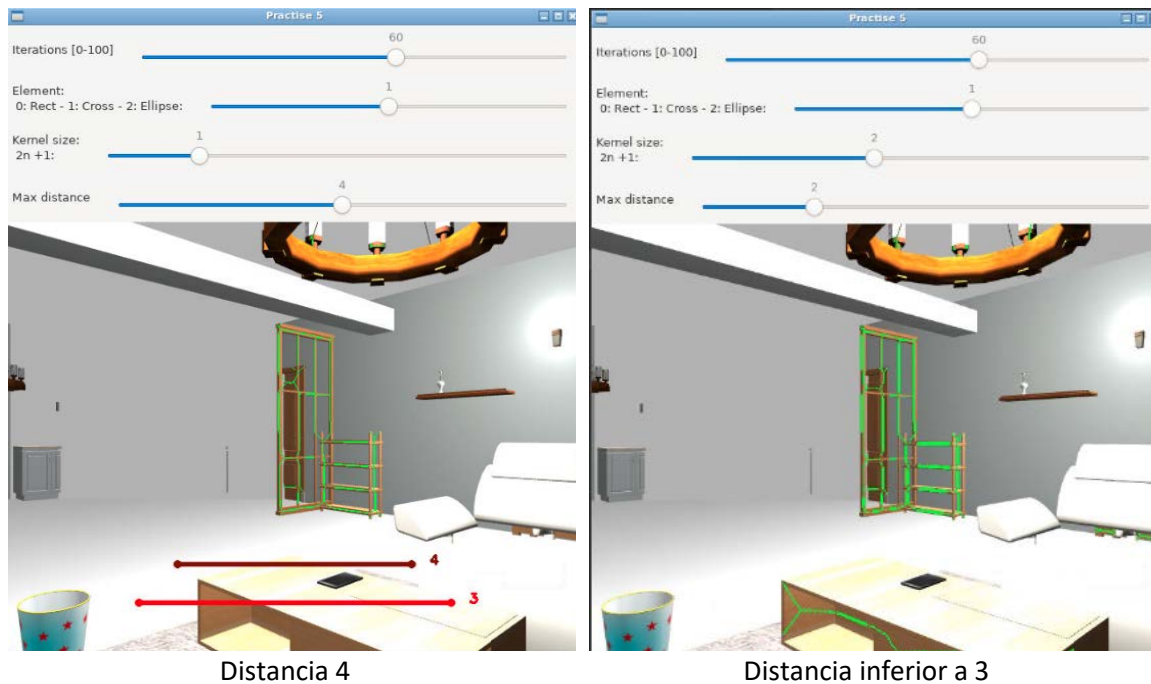


Property	Value
name	tiago:torso_lift...
self_collide	<input type="checkbox"/> False
gravity	<input checked="" type="checkbox"/> True
kinematic	<input type="checkbox"/> False
canonical	<input checked="" type="checkbox"/> False
enable_wind	<input type="checkbox"/> False
pose	
x	-0.062000
y	0.00
z	0.888497
roll	0.00
pitch	0.00
yaw	0.00

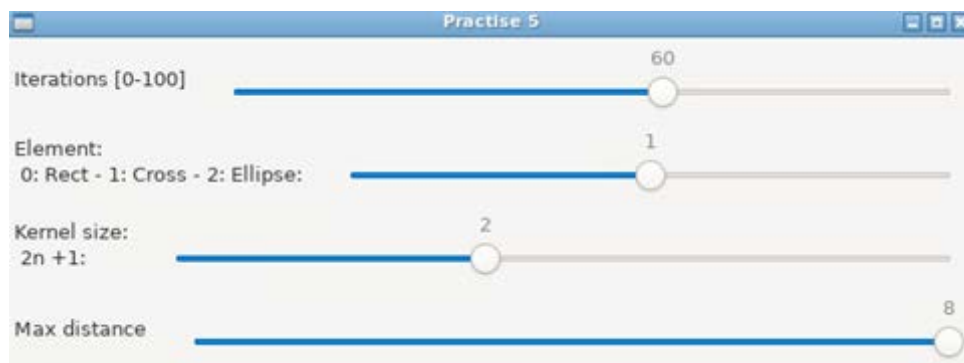
Crear un esqueleto de los objetos hecho de madera:

En este apartado, se pide hacer una **identificación de los objetos** que están hechos de **madera**, pero tienen un color **marrón claro** en el simulador (véase que las lámparas tienen un marrón oscuro), para ello, se puede aplicar cualquier técnica vista en clase para identificar únicamente aquellos objetos que son de madera clara.

Además, se pide que únicamente se identifiquen los objetos que aparecen a partir de la distancia máxima seleccionada en el apartado anterior, dentro de los 2 metros de ancho, salvo en el caso de que la distancia sea inferior a 3 metros y por consiguiente no se proyecte ningún punto en el plano de la imagen. En este caso, se obtendrá el esqueleto de todos los objetos que aparecen en la imagen.



Una vez identificados los objetos en la zona deseada, se pide crear el esqueleto de la imagen a partir de operaciones morfológicas. Para ello, se dotará al programa de tres sliders más. El **primero** controlará el **número** máximo de **iteraciones** que se van a realizar. El **segundo**, el **tipo** de **máscara** que se desea utilizar (Rectángulo, Cruz, Elipse). Y el **tercero** el **tamaño** de la **máscara**, siguiendo los ejemplos vistos en clase.



El algoritmo que hay que implementar para conseguir un esqueleto a partir de operaciones morfológicas es el siguiente:

1. Crear una imagen **esqueleto** vacía.
2. Realizar una **apertura** de la imagen. La llamaremos **open**.
3. **Restar** esta nueva imagen open a la imagen original. La llamaremos **temp**.
4. **Erosionar** la imagen original, y redefinir la imagen esqueleto calculando la **unión** entre el **esqueleto** actual y la imagen **temp**.
5. **Repetir los pasos 2-4** tantas veces como se haya elegido en el slider correspondiente.

El **esqueleto** se mostrará sobre la imagen original con las líneas del apartado anterior, y el **esqueleto** calculado superpuesto será de color **verde**.

Capturas:

