

# **Sistemas de Percepción**

## **Práctica 1**



**UNIVERSIDAD DE SEVILLA**

**Realizado por:**

**Fernando Román Hidalgo**

**Andrés Martínez Márquez**

# Índice

<b>1. Introducción .....</b>	<b>3</b>
<b>2. Versión básica .....</b>	<b>3</b>
2.1. Creación entorno .....	3
2.2. Desarrollo apartado .....	3
<b>3. Versión avanzada .....</b>	<b>4</b>
3.1. Descripción del sistema .....	4
3.2. Versión sin renderizado .....	4
3.3. Versión con vista desde la cámara .....	5
3.4. Versión con renderizado .....	6
3.5. Movimiento de los cubos .....	6
<b>4. Instrucciones para ver el programa .....</b>	<b>7</b>
<b>5. Conclusión .....</b>	<b>7</b>

# 1. Introducción

La práctica consiste en la creación de un entorno de trabajo 3D, e implementar los conocimientos adquiridos en la asignatura para renderizar y manejar este entorno. A lo largo de la práctica, se han creado y proyectado marcadores sobre el entorno lo que simula la captura de imágenes de una cámara. Esta captura se consigue de forma adecuada mediante el uso de geometría de proyección y el renderizado de imágenes reales.

## 2. Versión básica

### 2.1. Creación entorno

En primer lugar se ha formado el modelo geométrico del sistema. Se ha creado el sistema de referencia general  $\{W\}$ , y también se ha elegido la posición y la orientación de la cámara, con su propio sistema de referencia  $\{C\}$ . Asimismo, se ha colocado un marcador rectangular en medio, que se usará para colocar un marcador ArUco en este primer apartado.

Se ha configurado la cámara con los parámetros intrínsecos facilitados en la práctica, y se han usado las imágenes de la mesa y el marcador ArUco del material del alumno.

### 2.2. Desarrollo apartado

En este primer apartado de la práctica se han generado dos puntos de vista distintos simulando dos posibles capturas de la cámara, creando 2 sistemas de referencia  $\{C\}$  distintos. Para cada uno de ellos se ha obtenido una representación tridimensional del escenario, Figuras 1 y 2, así como la proyección de los puntos de interés desde el punto de vista de la cámara, Figuras 3 y 4.

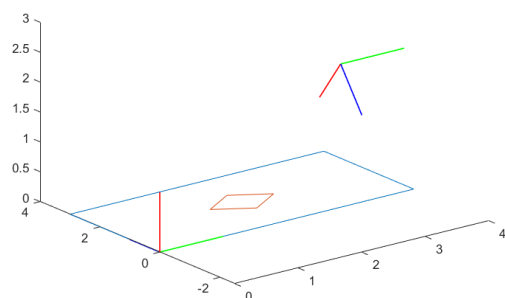


Figura 1: Representación 3D del escenario para la primera proyección

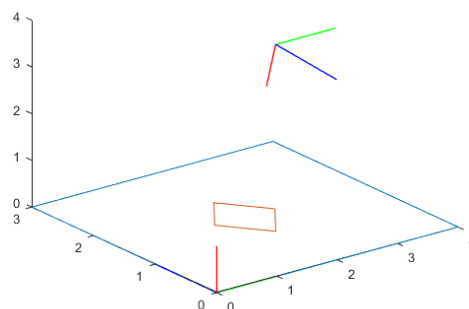


Figura 2: Representación 3D del escenario para la segunda proyección

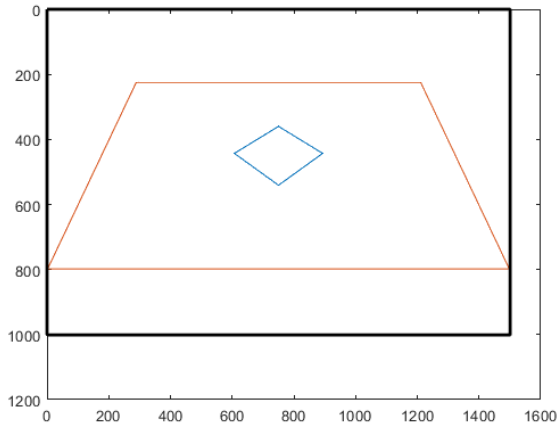


Figura 3: Proyección desde el punto de vista 1

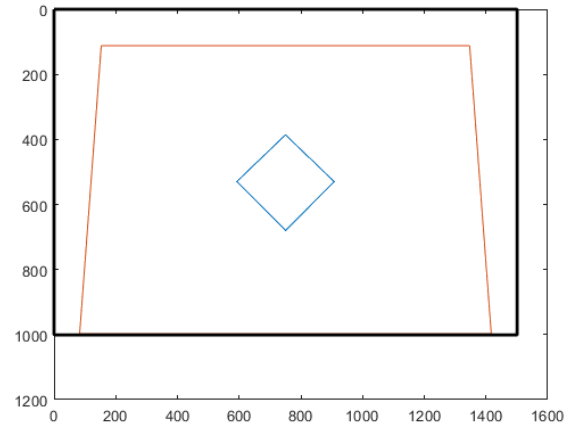


Figura 4: Proyección desde el punto de vista 2

Al final, se han obtenido las imágenes renderizadas que se podrían observar desde los diferentes puntos de vista de la cámara, Figuras 5 y 6.

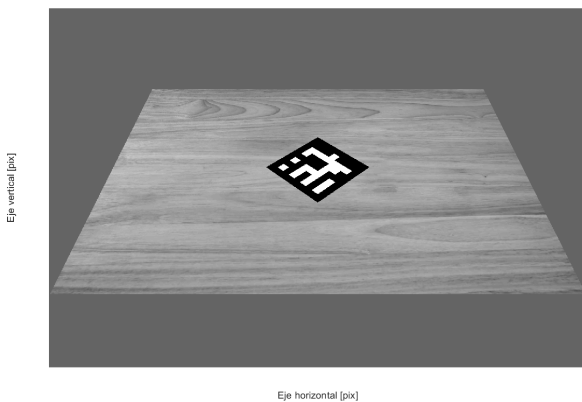


Figura 5: Renderizado desde el punto de vista 1

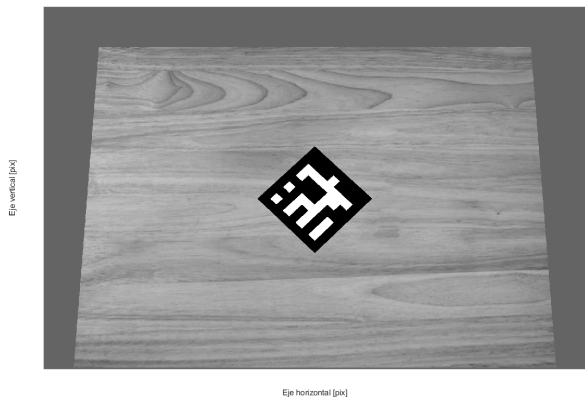


Figura 6: Renderizado desde el punto de vista 2

### 3. Versión avanzada

#### 3.1. Descripción del sistema

En el segundo apartado de la práctica, se ha simulado que los marcadores ArUco se encuentren encima de unos pequeños robots móviles que se van a ir moviendo a lo largo de la mesa.

Para facilitar la simulación, se ha supuesto que los robots son cubos de 20 cm de lado con dos ruedas. Se han simulado 4 robots, cada uno con su marcador ArUco y su propio color identificativo.

#### 3.2. Versión sin renderizado

Para comenzar, se han querido representar 4 cubos de 20x20x20 cm sobre la mesa ya creada anteriormente. Para ello se ha creado la matriz correspondiente a los vértices del cubo y a las aristas. Con esto se ha procedido a dibujar cada cubo en la posición deseada añadiéndole un offset distinto a los vértices de cada cubo. A su vez, cada cubo contará con un eje de

coordenadas propio, que se le añadirá mediante la función DrawFrame. En cuanto a las ruedas, se dibujarán mediante 50 puntos dispuestos en circunferencias de 5 cm de radio en el plano xy y centrados en el centro de las caras del cubo. Al dibujar mediante plot3 todos los elementos nombrados nos queda el dibujo correspondiente a la Figura 7.

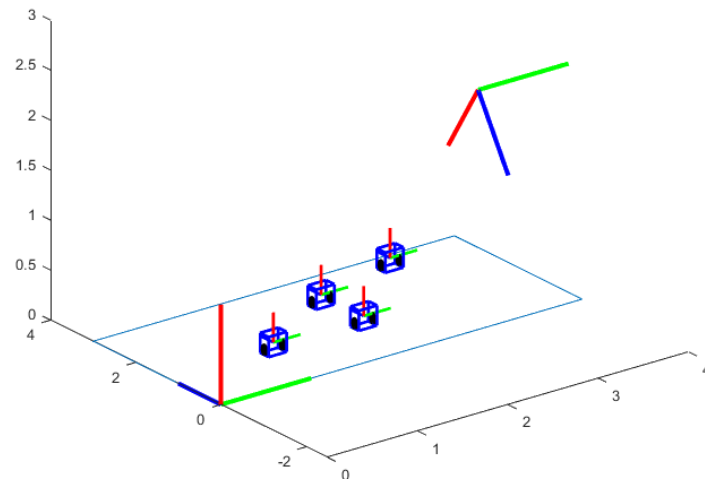


Figura 7: Representación 3D de los cubos

### 3.3. Versión con vista desde la cámara

Para proyectar el escenario en el plano de la cámara, se han calculado las matrices de transformación de los vértices y los puntos de las ruedas de los cubos. Se representarán con un plot estas matrices incluyendo la mesa y el marco de la cámara, viéndose así la proyección de los cubos desde el punto de vista de la cámara como se representa en la Figura 8.

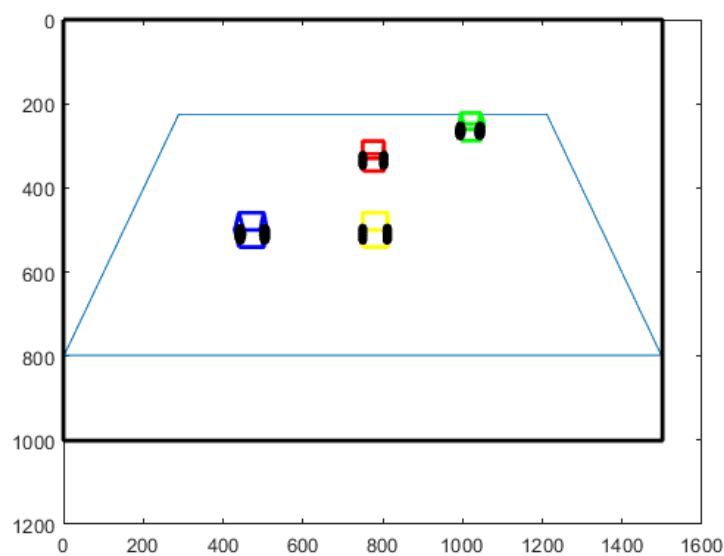


Figura 8: Vista desde la cámara de los cubos

### 3.4. Versión con renderizado

A continuación, se han repetido los procesos anteriormente descritos, pero renderizando todas las imágenes a la vez. Se han incluido 4 marcadores de ArUco generados de la forma indicada en la guía de la práctica, cada uno con un color distinto, y la mesa pero en este apartado con color y una mayor calidad lograda mediante un redimensionado de la imagen. Como se puede comprobar en la Figura 9, los ArUcos tienen cada uno un color y patrón distinto, y se encuentran en la posición de los cubos.



Figura 9: Vista desde la cámara con renderizado

### 3.5. Movimiento de los cubos

Para añadir el efecto de movimiento a los cubos, se ha planteado de forma que cada cubo se mueva de forma vectorial en dirección del eje Y del propio cubo y rote con respecto a su eje Z.

Por lo tanto, se ha propuesto un movimiento semialeatorio en el cual los cubos se mueven en cada iteración una distancia aleatoria dentro de unos rangos, y rotan un ángulo también aleatorio dentro de unos rangos de forma incremental a la posición anterior.

De esta forma en cada iteración los cubos estarán en una posición distinta a la anterior, y añadiéndole un pause al final de cada iteración crea la sensación de un movimiento fluido.

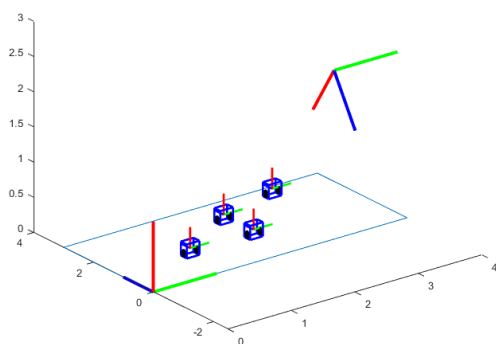


Figura 10: Posición inicial

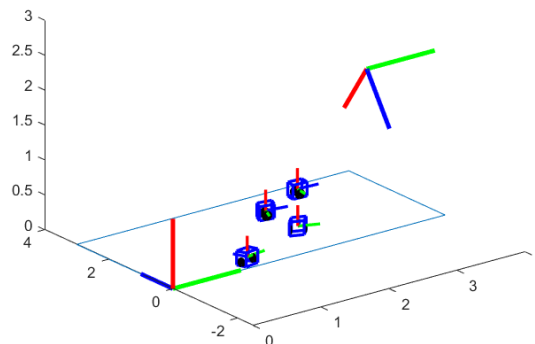


Figura 11: Posible posición final

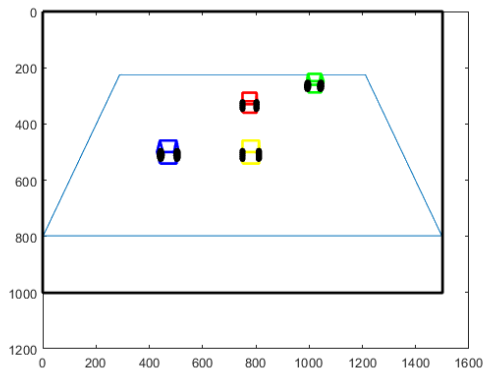


Figura 12: Posición inicial vista desde la cámara

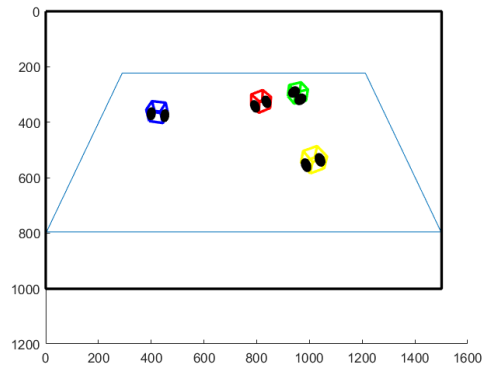


Figura 13: Posible posición final vista desde la cámara

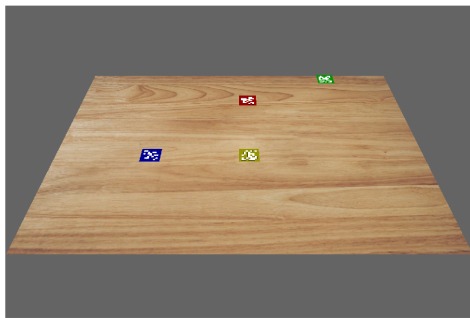


Figura 14: Posición inicial con renderizado



Figura 15: Posible posición final con renderizado

Una mejora que ha sido añadida es la implementación de límites en el movimiento de los cubos, con objetivo de que no se choquen entre ellos ni se salgan de la mesa. Esto se ha logrado mediante condiciones que miden la distancia de separación de los cubos entre ellos y hasta los bordes de la mesa. Si esta distancia se vuelve muy pequeña, los cubos implicados tomarán la dirección contraria en la siguiente iteración para evitar el choque.

## 4. Instrucciones para ver el programa

Los vídeos de cada una de las vistas están adjuntados en el *.zip*. A su vez se puede ejecutar cada parte del código desde el archivo *ProgramaPrincipal.m*. Como está indicado en el programa, hay que ejecutar las secciones en orden para obtener variables necesarias para las siguientes partes.

## 5. Conclusión

En esta práctica se ha trabajado el cambio de sistemas de referencia y las propiedades de una cámara, el uso de ArUcos y la renderización de imágenes. Con la realización de la versión avanzada se ha complementado el conocimiento ya adquirido con otros como la simulación de movimientos y la representación de formas complejas como cubos y pareciéndose más a un caso real.