

# Diseño de algoritmo para obtener la posición y orientación de una cámara con respecto a ArUcos.

- Martín Martínez - 1800000
- Maximiliano García - 180026
- Edgar García - 180032
- Fernando Durán - 180079
- Allan González - 180083

# Introducción:

En este presente documento se mostrarán los resultados del desarrollo de un algoritmo para obtener la posición de una cámara con respecto a un par de ArUcos, así como la metodología empleada.

## Planteamiento del problema:

Conociendo las dimensiones de los cuadros que se encuentran en el tablero de ajedrez para calibrar la cámara, se obtienen los parámetros de esta. De esta manera, podemos aplicar un algoritmo para conocer la posición de la cámara con respecto al ArUco.

Sin embargo, en [1] se menciona que puede existir un problema de ambigüedad a la hora de estimar la posición utilizando un sólo ArUco. Este problema se presentó en nuestro caso, ya que variaba mucho la posición.

# Objetivo:

Desarrollar un algoritmo para obtener la posición de una cámara con respecto a un par de ArUcos.

# Objetivos particulares:

- Capturar fotografías de tablero de ajedrez con la cámara.
- Desarrollar algoritmo para calibrar la cámara.
- Obtener matriz de calibración de la cámara.
- Obtener posición y orientación del ArUco con respecto a la cámara.
- Generar gráfica para representar la posición y orientación del ArUco.
- Validar algoritmo

# Marco teórico:

## Marcador ArUco:

Un marcador ArUco es un identificador único compuesto por un borde y una matriz binaria que se utiliza como un sistema de referencia, el cual permite conocer la posición y orientación utilizando la librería de OpenCV.

El tamaño del marcador determina el tamaño de la matriz interna. Los distintos tipos de arucos contienen matrices de 4x4, 5x5, 6x6, 7x7 y funcionan con un diccionario que son un conjunto de marcadores para una aplicación específica para cada uno de los valores de las matrices.

## OpenCV:

Biblioteca de código abierto para aplicaciones que utilicen visión artificial, es utilizada en varias aplicaciones como control de procesos, sistemas de seguridad con detección de movimiento, reconocimiento de objetos, robótica avanzada, etc.

Algunos ejemplos de sus aplicaciones son:

- ❖ Identificación de objetos
- ❖ Reconocimiento de rostros
- ❖ Interfaces de usuario
- ❖ Representación 3D.

## Matriz de rotación:

Matriz que representa la rotación en grados del plano en sentido antihorario. La orientación de un punto puede expresarse con los tres ángulos de Euler que corresponden a la rotación respecto a los ejes x, y, z. La rotación en cada eje puede expresarse como una matriz para cada ángulo.

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Matriz de transformación homogénea:

Matriz 4x4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro.

Permite representar conjuntamente la posición y orientación y se facilita su uso mediante el álgebra matricial

$$T = \begin{bmatrix} R_{3x3} & P_{3x1} \\ f_{1x3} & 1x1 \end{bmatrix} = \begin{bmatrix} \text{matriz de rotación} & \text{vector de posición} \\ \text{transformación de perspectiva} & \text{escalado} \end{bmatrix}$$

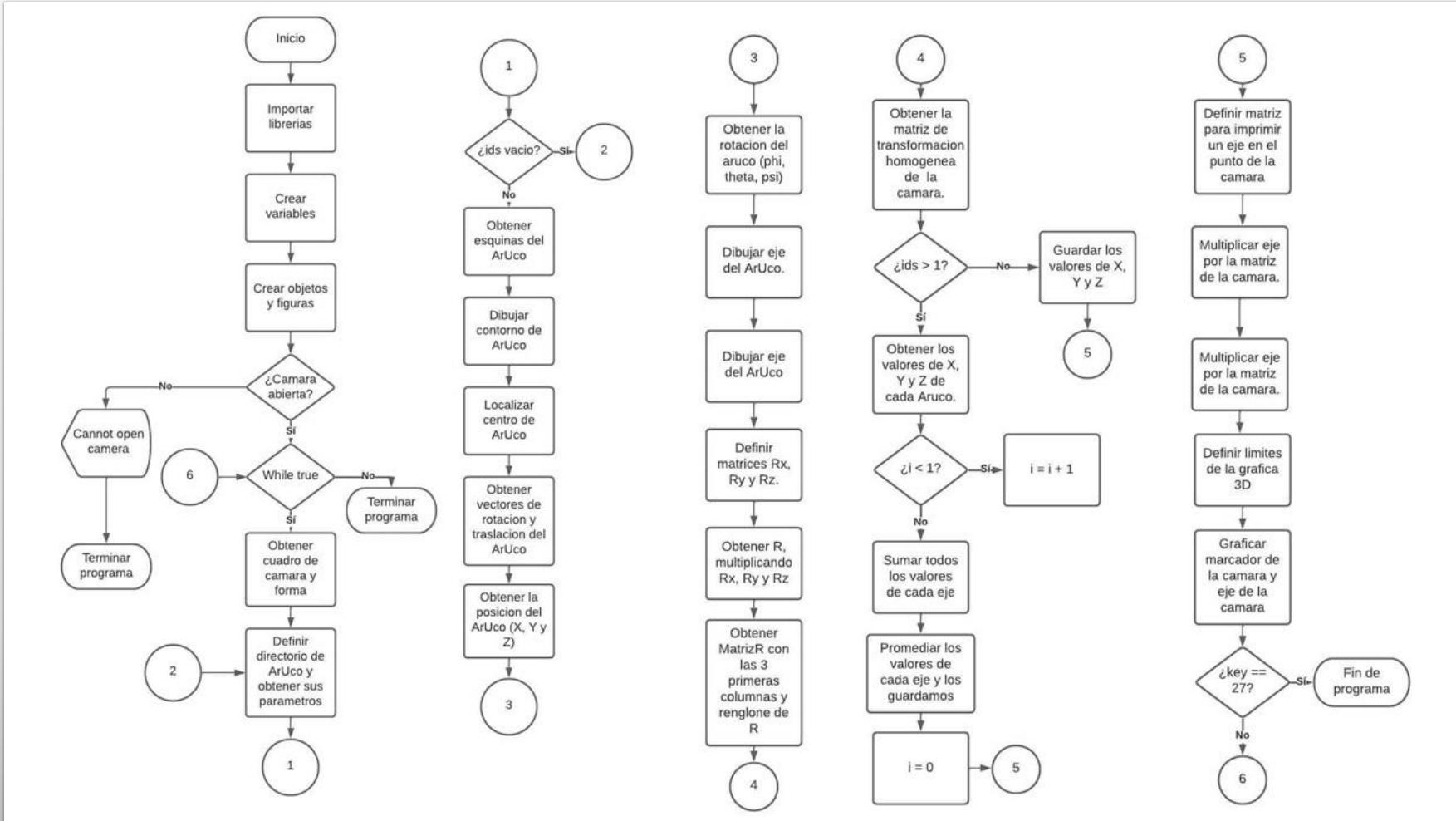
## Matriz de transformación inversa:

Una matriz inversa puede expresarse como una matriz de transformación homogénea.

Como se sabe que la inversa de una matriz de rotación es su transpuesta y tomando como referencia el vector inicial de A a B nos queda el siguiente término para la inversa de la matriz:

$$T_A^B = \begin{bmatrix} R_B^{A^T} & -R_B^A(T_{r_B}^A) \\ 0 & 1 \end{bmatrix}$$

# Diagrama de flujo:



## Conclusión:

El sistema cumple con la tarea que fue definida, pero la precisión con la que trabaja el sistema depende de muchos factores, como la iluminación, la calidad de la cámara y la posición en donde se encuentra el ArUco. Es por ello que un trabajo a futuro, se buscaría de trabajar en los factores que afectan al sistema para que ofrezca una mejor precisión.

# Referencias:

- [1] R. Jiménez, “Sistema de seguimiento de objetos usando OpenCv, ArUco y Filtro de Kalman extendido”, tesis, Universidad de Sevilla, SE, España, 2018.
- [2] J. N. A. Rivero, «Robot Dirigido Por Voz,» Universidad de las Américas Puebla, Puebla, 2007.
- [3] E. A. Prieto y P. A. G. Quintero, «Diseño e implementación de un robot móvil autónomo con detección de color,» Universidad De San Buenaventura, Bogotá, 2006.
- [4] G. Viera-Maza, «Procesamiento de imágenes usando OpenCV aplicado en Raspberry PI para la clasificación del cacao,» Universidad De Piura, Piura, 2017.