

## INFORME

### TALLER 5: Cámara y Calibración

Erick Steven Badillo Vargas [e-badillo@gmail.com](mailto:e-badillo@gmail.com)

## I. Calibración de la cámara

Se desarrollo un script en Python que calibre la cámara en base a una galería de fotos de un tablero de Ajedrez, con ayuda de la función de Open-Cv *calibrateCamera* obteniendo de este los parámetros intrínsecos de la cámara y de distorsión.

### a. Cámara Web

Matriz de parámetros intrínsecos obtenidos de una cámara para computador:

$$\begin{bmatrix} 1.13317478e^3 & 0 & 6.86927967e^2 \\ 0 & 1.13252480e^3 & 3.33668062e^2 \\ 0 & 0 & 1 \end{bmatrix}$$

Vector de distorsión:

$$[0.07481802 \quad 1.73958873 \quad -0.00237835 \quad 0.02768936 \quad 0.37167247]$$

### b. Cámara celular

Matriz de parámetros intrínsecos obtenidos de una cámara de celular:

$$\begin{bmatrix} 680.85876544 & 0 & 316.01566085 \\ 0 & 676.7749807 & 261.99565367 \\ 0 & 0 & 1 \end{bmatrix}$$

Vector de distorsión para cámara de celular:

$$[1.94204627e^{-1} \quad -1.45411189 \quad -4.70953186e^{-3} \quad 2.73203996e^{-3} \quad 3.38187987]$$

La matriz de parámetros intrínsecos se guardo en un archivo “.JSON” con nombre de “K” para luego ser leída, además se guardaron parámetros extrínsecos como la altura, la distancia, el ángulo “tilt” y el ángulo “pan”. Estos fueron guardados para los dos tipos de cámara (*ver archivos adjuntos: calibration.json y calibrationweb.json*)

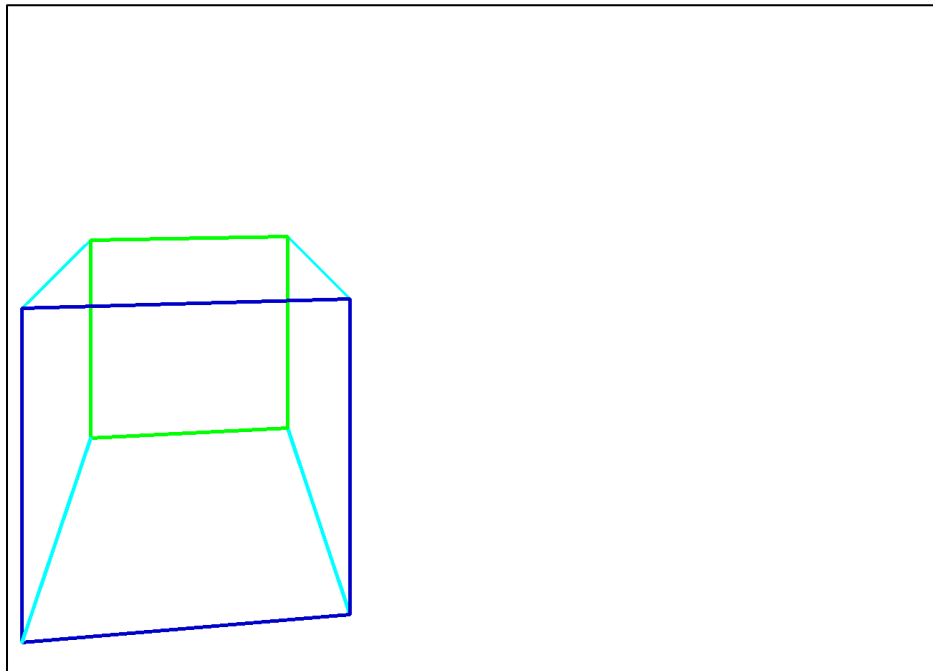
## II. Cámara proyectiva

Luego se extrae del archivo “.JSON” los parámetros guardados anteriormente y se crean puntos coincidentes a un cubo en 3D (centrado en el origen), que con los parámetros obtenidos de la calibración se crea “un modelo de cámara” y se realizan proyecciones 3D de

los puntos para ciertos parámetros extrínsecos. Y se unen líneas creando un aspecto de “cubo 3D”:

**a.  $d = 2$ ,  $h = 1$ ,  $tilt = 0$ ,  $pan = 5$**

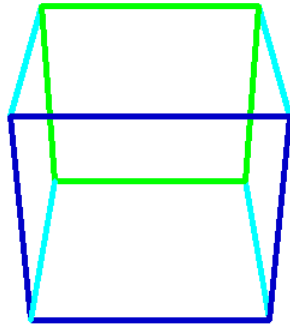
Para una distancia de 2, una altura de 1 un ángulo tilt de 0, un pan de 5 y el modelo de una cámara de celular, el cubo con una re-proyección se ilustra así:



*Figura 1  $d=2$ ,  $h=1$ ,  $tilt=0$ ,  $pan=5$*

**b.  $d = 3$ ,  $h = 2$ ,  $tilt = 30$ ,  $pan = 0$**

Para una distancia de 3, una altura de 2 un ángulo tilt de 30, un pan de 0 y el modelo de una cámara de celular, el cubo con una re-proyección se ilustra así:



*Figura 2  $d=3$ ,  $h=2$ ,  $tilt=30$ ,  $pan=0$*

Los cubos 3D se pueden observar con una re-proyección aproximada a la ideal, el modelo de la cámara puede afectar, y las fotos deben ser tomadas con buena calidad para que el modelo sea mas acertado. (Para mas detalle del algoritmo utilizado puede consultar los archivos adjuntos, *main.py*, *second.py*)