

VISIÓN ARTIFICIAL : COMENTARIOS CALIBRACIÓN

- Entre 10 y 20 imágenes.
- Imprimir plantilla1 sin modificar relación de aspecto.
- Hacer varias fotos cambiando posición y orientación sin cambiar zoom.
- Cubrir bien la parte de los bordes (hacer varias fotos).
- Eje x , eje y vienen cambiados por error en la plantilla, el programa los va a poner bien.
- Las esquinas son dentro del tablero, que hagan forma de cruz. Resultado es un cuadrado verde que alberga la zona completa.
- dX es la medida en la realidad, hay que medirlo con una regla.
- Las primeras aspas rojas que salen es la predicción de donde van a estar las esquinas, aún no ha empezado a buscarlas.
- Si hay mucha distorsión se le avisa, aunque hemos puesto en todos que no hay.
- Cuando ya hemos marcado las esquinas en todas las fotos, le damos a calibrar.
- Los parámetros obtenidos son:
 1. $fc = [fx \ fy]$
 2. $cc = [u0 \ v0]$
 3. $\alpha_c = s \rightarrow skew$
 4. $kc = [kr1 \ , \ kr2 \ , \ kt1 \ , \ kt2 \ , \ kr3]$
 5. $err = \text{error de reproyección} \rightarrow$ Si es menor a 1 píxel se considera buena la calibración.
- Show extrinsic = predice como se han tomado las fotografías \rightarrow Muestra los parámetros extrínsecos.
- Reproject on images = Reproyecta para permitirme evaluar como de buenas son las proyecciones en la imagen. Mientras más grande es la flecha más desviado estaba en rotación u orientación con respecto a la cámara.

- **Analyse error** = Pulsamos un punto de la nube de puntos de error para ver a qué imagen pertenece y podemos decidir quitar esa imagen si mete mucho error. Puede ser que no haya imágenes erróneas, sino que el sensor esté mal calibrado de verdad, o que el skew no sea nulo, o que el k_r3 despreciado no sea nulo.
- **est_alpha** a 0 indica que no usa skew, 1 que sí.
- **est_dist** a 1 aquellos parámetros de distorsión que queremos, a 0 los que no.
- **RESULTADOS:**

Calibration results after optimization (with uncertainties):

Focal Length: $fc = [2653.31253 \ 2655.78055] \pm [7.74914 \ 7.53238]$
Principal point: $cc = [1612.83788 \ 1222.18881] \pm [6.65732 \ 5.70268]$
Skew: $\alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: $kc = [0.14443 \ -0.30713 \ -0.00321 \ 0.00171 \ 0.00000] \pm [0.00973 \ 0.04967 \ 0.00092 \ 0.00116 \ 0.00000]$
Pixel error: $err = [0.60692 \ 0.60174]$

Note: The numerical errors are approximately three times the standard deviations (for reference).

- Los \pm indican una desviación para una normal de $3 \cdot \sigma$ (qué implica un 99% de probabilidad de encontrar ahí).
- Para usar el otro Toolbox usar **cameraCalibrator**. Una vez obtenida la variable **cameraParams**, obtenemos la matriz **A** llamando a **cameraParams.IntrinsicMatrix**, pero nos da el resultado traspuesto.

PRÁCTICA 1

1 DEFINICIÓN DEL OBJETO

Crear una matriz de puntos con un bucle para que genere los puntos (unos 70).

Se hace un programa para los cálculos (ejercicio 5 de clase es buena guía), primero que calcule sin distorsión y luego con ella. Cambiar también los valores de distorsión para obtener más resultados. El punto de partida debe ser fácil, cámara y objeto enfrentados como viene en la memoria.