

Práctica 02

Calibración de Cámaras

Visión Computacional Aplicada a la Robótica

UNAM, 2021-2

Resumen

El alumno aprenderá a obtener los parámetros intrínsecos de una cámara, así como los coeficientes de distorsión radial y tangencial. Después, el alumno utilizará los parámetros obtenidos para corregir la distorsión del video de una WebCam.

Duración

2 semanas

Desarrollo

1. Utilizando el tutorial publicado en https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html realizar lo siguiente:
 - a) Identifique el número de esquinas ($r \times c$) que se pueden detectar en los tableros de ajedrez de las imágenes de la carpeta `data`.
 - b) Genere un conjunto de puntos P de la forma $(0, 0, 0), (0, 1, 0), \dots, (i, j, 0) \dots (r, c, 0)$. Estos puntos representarán las coordenadas en 3D de las esquinas detectadas en un tablero. Estas coordenadas serán las mismas para todas las imágenes de muestra.
 - c) Cargar todas las imágenes de muestra de la carpeta `data` y transformarlas a escala de grises.
 - d) Utilizando la función `findChessboardCorners`, encontrar las esquinas del tablero para cada imagen.
 - e) Utilizando la función `cornerSubPix`, refinar las coordenadas de las esquinas encontradas.
 - f) Utilizando el conjunto de puntos P y las esquinas encontradas en cada imagen, determinar los parámetros de calibración utilizando la función `calibrateCamera` de OpenCV.
 - g) Almacenar en un archivo en disco la matriz de parámetros intrínsecos y el arreglo con los coeficientes de distorsión. Se sugiere el uso de la función `savez` de la biblioteca `numpy`.
2. Utilizando el mismo tutorial:

- a) Cargar los parámetros de calibración almacenados en disco. Se sugiere el uso de la función `load` de la biblioteca `numpy`.
 - b) Corregir la distorsión en todas las imágenes de la carpeta `data`.
 - c) Mostrar en pantalla la imagen original y la imagen sin distorsión.
3. Calibración de una WebCam:
- a) Imprimir la imagen `chessboard.png`
 - b) Con una WebCam, tomar varias fotos del tablero desde varios ángulos, cuando menos diez imágenes.
 - c) Realizar un proceso similar al inciso 1 para calibrar la cámara utilizada.
 - d) Hacer un programa que obtenga el video de la cámara y corrija la distorsión. Desplegar tanto el video original como el video sin distorsión.

Funciones de referencia

- `numpy.zeros`
- `numpy.mgrid`
- `numpy.savez`
- `numpy.load`
- `cv2.imread`
- `cv2.findChessboardCorners`
- `cv2.cvtColor`
- `cv2.calibrateCamera`
- `cv2.drawChessboardCorners`
- `cv2.cornerSubPix`
- `cv2.VideoCapture`

Entregables:

Los siguientes entregables se refieren al inciso 3 del desarrollo.

- Archivo zip con el conjunto de imágenes tomadas del tablero de ajedrez.
- Archivo con los parámetros de calibración obtenidos (de preferencia un archivo con extensión `.npz`).
- Código fuente utilizado para la obtención de los parámetros de calibración y para corregir la distorsión del video.