



Universidad Autónoma de San Luis Potosí

Facultad de ingeniería

Tratamiento de Imágenes

Práctica 9



FACULTAD DE
INGENIERÍA

Nombre Práctica: Corrección de Distorsión de la

Cámara y calibración con OpenCV

Nombre del Alumno: Manuel Ramírez Galván

Fecha: 01/04/2025

Procedimiento

9.1. Diseñar un programa en Python en el cual se cumplan los siguientes requisitos:

- Acceder a una cámara web.
- Encontrar las esquinas de un tablero de ajedrez y mostrar el patrón encontrado mediante el uso de la función cv2.findChessboardCorners().
- Si se encuentra el patrón de tablero de ajedrez, permitir guardar las imágenes para la calibración de cámara.

Resultados

```
● ● ●

import cv2

cam = cv2.VideoCapture(0)

while cam.isOpened():
    ret, frame = cam.read()
    if not ret:
        break

    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
    CHESS_BOARD_DIM = (6, 4)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Encontrar las esquinas del tablero de ajedrez
    ret, corners = cv2.findChessboardCorners(gray, CHESS_BOARD_DIM)

    # Si son encontradas, refinrar las esquinas y dibujarlas en la imagen a color
    if ret:
        corners1 = cv2.cornerSubPix(gray, corners, (11, 11), (-1,-1), criteria)
        cv2.drawChessboardCorners(frame, CHESS_BOARD_DIM, corners1, ret)

    cv2.imshow('Ejercicio 1', frame)
    key = cv2.waitKey(1)
    if key == ord('q'):
        break

cam.release()
cv2.destroyAllWindows()
```

Imagen 1.- Código 1 Ejercicio 1

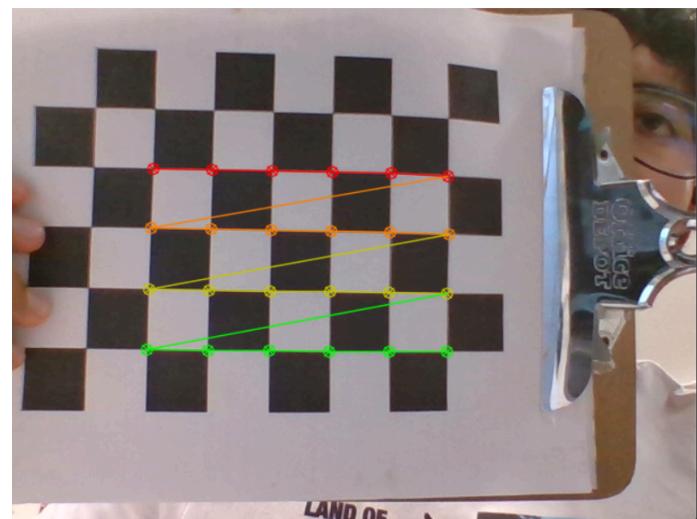


Imagen 2.- Detección de Esquinas de Tablero de Ajedrez

Procedimiento

9.2. Diseñe un programa en Python en el cual se cumplan los siguientes requisitos:

- Usando las imágenes anteriormente capturadas, encontrar los parámetros extrínsecos de la cámara web.
- Encontrar los tipos de distorsiones en la cámara.
- Guardar los parámetros para su corrección.

Resultados



```
● ● ●

import os
import numpy as np

CHESS_BOARD_DIM = (6, 4)
SQUARE_SIZE = 20
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)

calib_data_path = "calib_data"

CHECK_DIR = os.path.isdir(calib_data_path)
if not CHECK_DIR:
    os.mkdir(calib_data_path)
    print(f"Directorio {calib_data_path} creado.")

else:
    print(f"Directorio {calib_data_path} ya existe.")

obj_3D = np.zeros((CHESS_BOARD_DIM[0] * CHESS_BOARD_DIM[1], 3), np.float32)
obj_3D[:, :, 2] = np.mgrid[0:CHESS_BOARD_DIM[0], 0:CHESS_BOARD_DIM[1]].T.reshape(-1, 2)
obj_3D *= SQUARE_SIZE

obj_points_3D = []
img_points_2D = []

image_dir_path = "Fotos"

files = os.listdir(image_dir_path)
for file in files:
    print(file)
    imagePath = os.path.join(image_dir_path, file)

    image = cv2.imread(imagePath)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    ret, corners = cv2.findChessboardCorners(gray, CHESS_BOARD_DIM, None)
    if ret:
        obj_points_3D.append(obj_3D)
        corners1 = cv2.cornerSubPix(gray, corners, (3, 3), (-1, -1), criteria)
        img_points_2D.append(corners1)

ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(obj_points_3D, img_points_2D, gray.shape[:-1], None, None)
print("Calibrada")

print("Guardando datos de calibración")
np.savez(
    f"{calib_data_path}/MulMatrix",
    camMatrix=mtx,
    distCoef=dist,
    rVector=rvecs,
    tVector=tvecs,
)
```

Imagen 3.- Código Ejercicio 2

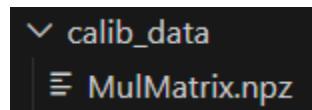


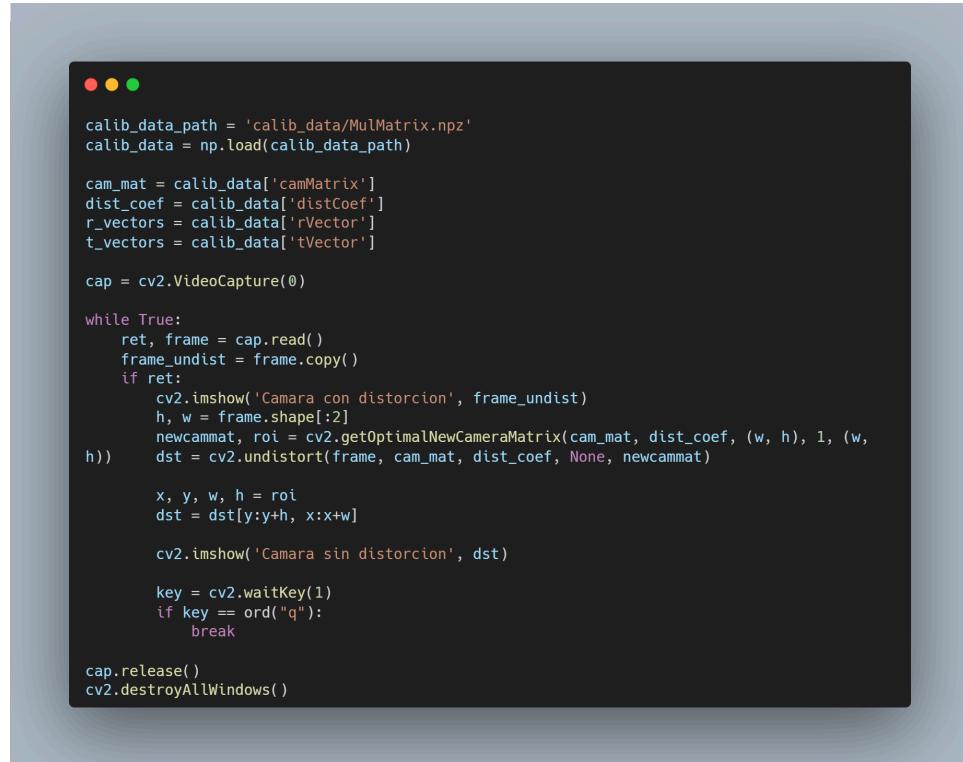
Imagen 4.- Archivo de Datos de Calibración

Procedimiento

9.3. Diseñe un programa en Python en el cual se cumplan los siguientes requisitos:

- Acceder a la cámara web.
- Con los parámetros encontrados y guardados, realizar la corrección de distorsión.
- Mostrar el video con la distorsión y resultado al aplicar la corrección.

Resultados



```
calib_data_path = 'calib_data/MulMatrix.npz'
calib_data = np.load(calib_data_path)

cam_mat = calib_data['camMatrix']
dist_coef = calib_data['distCoef']
r_vectors = calib_data['rVector']
t_vectors = calib_data['tVector']

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame_undist = frame.copy()
    if ret:
        cv2.imshow('Camara con distorcion', frame_undist)
        h, w = frame.shape[:2]
        newcammat, roi = cv2.getOptimalNewCameraMatrix(cam_mat, dist_coef, (w, h), 1, (w, h))
        dst = cv2.undistort(frame, cam_mat, dist_coef, None, newcammat)

        x, y, w, h = roi
        dst = dst[y:y+h, x:x+w]

        cv2.imshow('Camara sin distorcion', dst)

    key = cv2.waitKey(1)
    if key == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

Imagen 5.- Código Ejercicio 3

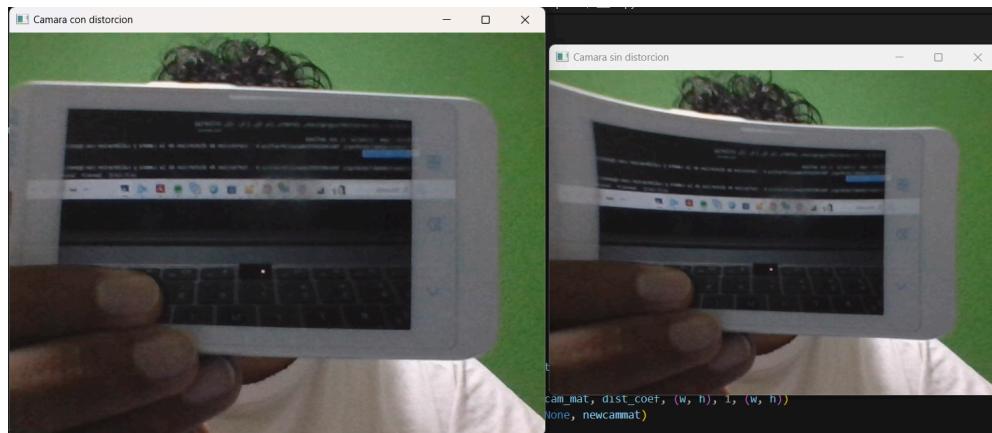


Imagen 6.- Comparación de de Capara con y sin Distorsión

Comprensión

1. ¿Cuáles son los parámetros intrínsecos y describe cada uno?

fx, fy (Focales en píxeles) representan el zoom o ampliación de la imagen en los ejes X y Y.

cx, cy (Coordenadas del centro óptico en la imagen) es donde los rayos que pasan por el centro del lente se cruzan en el sensor.

Matriz intrínseca (K) (Matriz 3×3 que agrupa fx, fy, cx y cy) se usa para transformar coordenadas 3D a 2D.

2. ¿Cuáles son los tipos de distorsión que existen?

Distorsión radial:

Hace que las líneas rectas se curven. Muy común en lentes gran angulares.

- **Barril:** líneas se curvan hacia afuera.
- **Almohada:** líneas se curvan hacia adentro.

Distorsión tangencial:

Se produce cuando el lente **no está perfectamente alineado** con el sensor. Hace que la imagen se vea inclinada o asimétrica.

3. ¿Qué es y para qué se utiliza la calibración de cámaras?

Es el proceso de estimar los parámetros intrínsecos y de distorsión de una cámara usando imágenes de referencia, por ejemplo, un patrón de tablero de ajedrez.

Se utiliza para:

- **Corregir la distorsión de la imagen**
Elimina las curvaturas o deformaciones provocadas por el lente.
- **Obtener medidas precisas**
Permite medir distancias reales y hacer reconstrucciones 3D a partir de imágenes.
- **Aplicaciones de visión artificial**
 - Detección y seguimiento de objetos
 - Estimación de poses
 - Reconstrucción 3D
 - Realidad aumentada (AR)
 - Robótica

Conclusiones

La corrección de distorsión y la calibración de cámaras permiten eliminar las deformaciones ópticas que los lentes agregan en las imágenes y obtener medidas precisas del entorno.

La calibración con permite estimar los parámetros intrínsecos de la cámara, como la distancia focal y el centro óptico, además de los coeficientes de distorsión radial y tangencial. Estos parámetros son importantes para convertir coordenadas del mundo real a coordenadas de imagen.

Una de las formas de realizar la calibración es mediante el uso de un tablero de ajedrez, el cual da un patrón regular y bien definido. Al capturar varias imágenes del tablero desde diferentes ángulos con la cámara de la computadora, se pueden extraer puntos clave que para calcular los parámetros intrínsecos y realizar la corrección de distorsión.