Лабораторная работа №1

Калибровка камеры телефона на основе шахматной доски

1. ЦЕЛЬ РАБОТЫ:

Цель данной лабораторной работы – провести калибровку камеры телефона с использованием шахматной доски. В рамках работы необходимо:

- Снять видео с шахматной доской, распечатанной на листе (размер доски 8×8).
- В процессе съемки перемещать телефон относительно шахматной доски (поднимать, поворачивать, сдвигать влево/вправо), чтобы получить кадры с разными ракурсами.
- Исправить скрипт calibration согласно своим данным.
- Полученные параметры и скриншоты результата направить на проверку.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

2.1. Проекция 3D-точек в 2D-изображение

Камера фиксирует двумерное изображение, но реальная сцена существует в трёхмерном пространстве. Для корректного отображения виртуальных объектов необходимо выполнить проекцию 3D-точек в 2D-пространство.

Матрица K имеет вид:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

где:

- f_x, f_y – фокусные расстояния (определяют «зум» или масштаб проекции);
- c_x, c_y – координаты главной точки (центр изображения).

При наличии 3D-точки P=(X,Y,Z) в мировой системе координат её проекция в пиксельные координаты (u, v) получается по формуле:

$$\begin{bmatrix} \mathrm{u} \\ \mathrm{v} \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ Z \\ 1 \end{bmatrix}, \quad (2)$$

2.2. Калибровка камеры

Калибровка камеры – это процесс вычисления внутренних параметров камеры и коррекции искажений объектива. Для этого используется шаблон (например, шахматная доска), на котором расположены легко обнаружимые узлы.

Параметры шаблона:

1. *CHECKERBOARD* - для внутреннего узлового паттерна выбирается размер 7×7, так как доска 8×8 дает 7 внутренних узлов по каждой оси.
2. *CHECKERBOARDSIZE* - размер ячейки доски (например, 100 мм).

```
# Defining the dimensions of checkerboard
CHECKERBOARD = (7, 7)
CHECKERBOARDSIZE = 100
```

Алгоритм калибровки:

1. С помощью *cv2.VideoCapture* открывается видеофайл (AVI) с шахматной доской.
2. Пробегаемся по кадрам видео с помощью цикла (например, используя *for path in fs*: для всех файлов в папке /data/calib). На свое усмотрение выбираем каждый (50-й) кадр и добавляем его в список.

```
for path in fs:
    cap = cv2.VideoCapture(os.path.join(main_path, path))
    f = True
    while f:
        f, im = cap.read()
        if count % 50 == 0:
            images.append(im)
        count = count + 1
```

3. Каждый выбранный кадр преобразуется в изображение в оттенках серого с помощью *cv2.cvtColor(img, cv2.COLOR_BGR2GRAY).*
4. Функция *cv2.findChessboardCorners(gray, CHECKERBOARD, flags)* ищет заданный шаблон (7×7) внутренних узлов на изображении. Если узлы найдены, возвращается *ret=True* и массив координат углов.

```
for img in images:
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    ret, corners = cv2.findChessboardCorners(gray, CHECKERBOARD, cv2.CALIB_CB_ADAPTIVE_THRESH + cv2.CALIB_CB_FAST_CHECK +
                                             cv2.CALIB_CB_NORMALIZE_IMAGE)
```

5. Для повышения точности координаты уточняются с помощью *cv2.cornerSubPix.*

6. Формируются два списка:

   - *objpoints* – список 3D-точек, соответствующих узлам шахматной доски в мировой системе (создается с использованием *np.mgrid* и масштабируется *CHECKERBOARDSIZE*).
   - *imgpoints* – список 2D-точек, найденных на изображениях.

```
    if ret is True:
        objpoints.append(objp)
        # refining pixel coordinates for given 2d points.
        corners2 = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1), criteria)
        imgpoints.append(corners2)
        # Draw and display the corners
        img = cv2.drawChessboardCorners(img, CHECKERBOARD, corners2, ret)
    cv2.imshow('img', img)
    cv2.waitKey(0)
cv2.destroyAllWindows()
h, w = img.shape[:2]
```

7. После сбора данных вызывается *cv2.calibrateCamera(objpoints, imgpoints, gray.shape[::-1], None, None)*, которая возвращает:

   - ***mtx*** - матрица камеры K
   - ***dist*** - коэффициенты дисторсии D
   - *rvecs, tvecs* - векторы вращения и смещения для каждого изображения.

```
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[::-1], None, None)
print("Camera matrix : \n")
print(mtx)
print("dist : \n")
print(dist)
print("rvecs : \n")
print(rvecs)
print("tvecs : \n")
print(tvecs)
```

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ И ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Скачайте архив с данными (содержащий видеофайлы в формате AVI, /data/calib) по ссылке: https://cloud.mail.ru/public/abmx/Li8kWSVrU для наглядного примера итогового варианта.
2. Распечатайте шахматную доску 8×8 на листе бумаги.

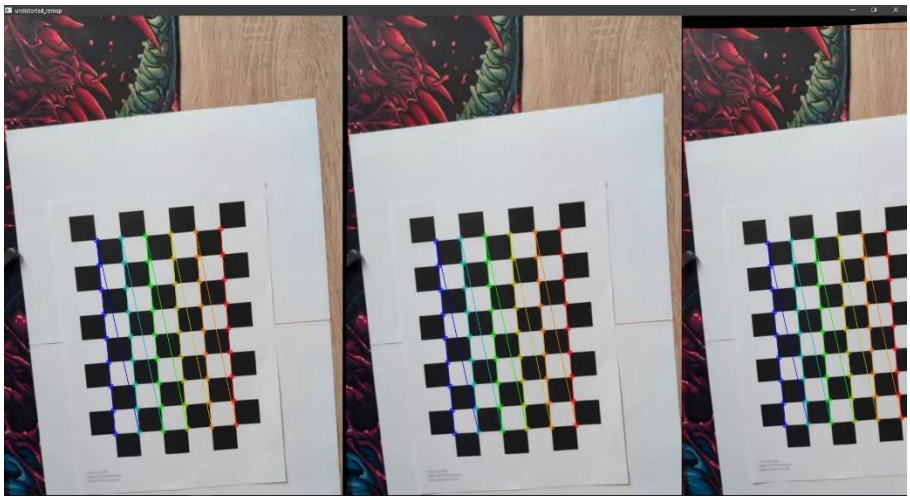
3. Запишите видео, в котором вы перемещаете камеру (телефон) относительно шахматной доски (поднимаете, сдвигаете, поворачиваете). Сохраните видео в формате .avi (в папку data/calib или укажите свои пути к видео).
4. Склонируйте репозиторий cv_book.
5. Перейдите в Module_1b/calibration.py и измените скрипт согласно вашим параметрам шахматной доски.
6. Получите матрицу камеры K (mtx), коэффициенты дисторсии D (dist) и 3 итоговых изображения.

Литература:

- Документация OpenCV – https://opencv.org/
- Учебное пособие по калибровке камеры в OpenCV - https://waksoft.susu.ru/2020/02/29/kalibrovka-kamery-s-ispolzovaniem-s-opencv/
- Дополнительные статьи по компьютерному зрению и калибровке – https://nikolasent.github.io/computervision/opencv/calibration/2024/12/20/Practical-OpenCV-Refinement-Techniques.html

ПРИЛОЖЕНИЕ

Пример итогового результата:



```
PS D:\Code> python calibration.py
Camera matrix :

[[1.31206136e+03 0.00000000e+00 2.83382912e+02]
 [0.00000000e+00 1.31316926e+03 6.70902611e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
dist :

[[ 0.2063038  -1.5844591  -0.00538798 -0.0280344   2.17038779]]
```