

Санкт-Петербургский политехнический университет Петра Великого  
Институт машиностроения, материалов и транспорта  
Высшая школа автоматизации и робототехники

# Отчёт

по лабораторной работе №5

Дисциплина: Техническое зрение

Тема: Распознавание образов на изображении при помощи контурного  
анализа

Студент гр. 3331506/70401

Преподаватель

Ляпцев И.А.

Варлашин В. В.

«    » \_\_\_\_\_ 2021 г.

Санкт-Петербург

2021

## **Задание**

1. Откалибровать собственную камеру при помощи шахматной доски.
2. Откалибровать собственную камеру при помощи доски Aruco.
3. Откалибровать собственную камеру при помощи Calibration Toolbox (Matlab).
4. Сгенерировать маркер Aruco из понравившегося словаря
5. Написать программу, которая бы детектировала маркер на изображении и рисовать куб с основанием в виде маркера (куб должен быть спроецирован на плоскость изображения и иметь различные цвета ребер).
6. Написать программу, которая красила бы видимые стороны куба в произвольный цвет.

## **Ход работы**

## 1. Калибровка при помощи шахматной доски

Алгоритм работы функции `CameraCalib()`:

- 1) Создание и заполнение вектора изображений.
- 2) Создание вектора из векторов 3D точек для каждого изображения и определение мировых координат 3D точек.
- 3) Нахождение углов на шахматной доске.
- 4) Калибровка камеры при помощи встроенной функции `calibrateCamera()`.
- 5) Сохранение результатов калибровки в файл *cameraMatrixChessBoard.txt*.

## 2. Калибровка при помощи доски aruco

Калибровка при помощи доски Aruco для удобства производится в функции *main()*.

Алгоритм калибровки:

- 1) Создание доски аруко.
- 2) Создание и заполнение вектора изображений.
- 3) Создать экземпляр встроенного класса `Board`, который хранит известные (так как размеры aruco доски известны) 3D-координаты маркеров.
- 4) Найти 2D-координаты маркеров при помощи встроенной функции `detectMarkers()`.
- 5) Откалибровать камеру при помощи встроенной функции `calibrateCameraAruco()`.
- 6) Сохранение результатов калибровки в файл *cameraMatrixAruco.txt*.

## 3. Калибровка при помощи Calibration Toolbox (Matlab)

Для калибровки в Calibration Toolbox (Matlab) необходимо загрузить калибровочные сэмплы с шахматной доской, настроить необходимые параметры, провести калибровку.

На рисунке 1 изображен калибровочный сэмпл с отмеченными внутренними углами шахматной доски.

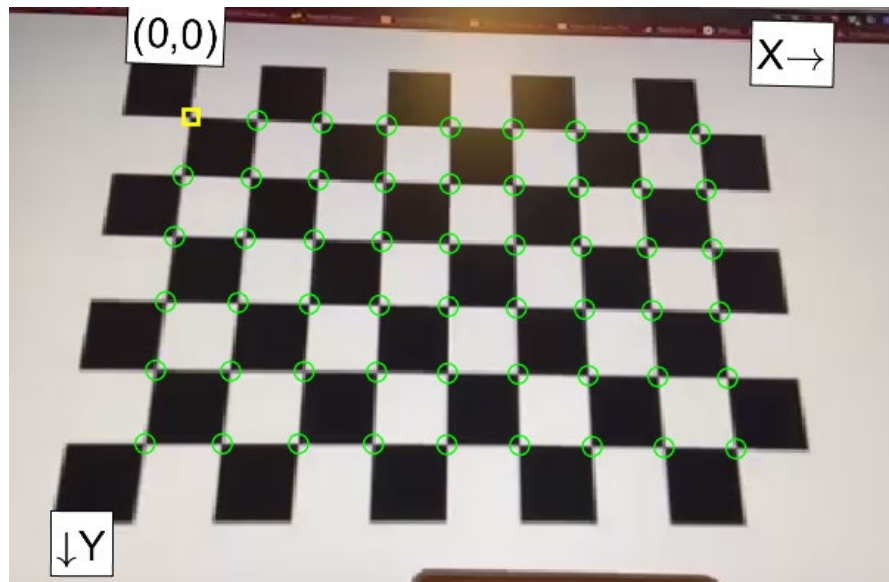


Рисунок 1 – Калибровка в Calibration Toolbox (Matlab)

В данной программе имеется возможность просмотра ошибки перепроекции каждого калибровочного сэмпла и положений камеры в пространстве (см. рисунок 2).

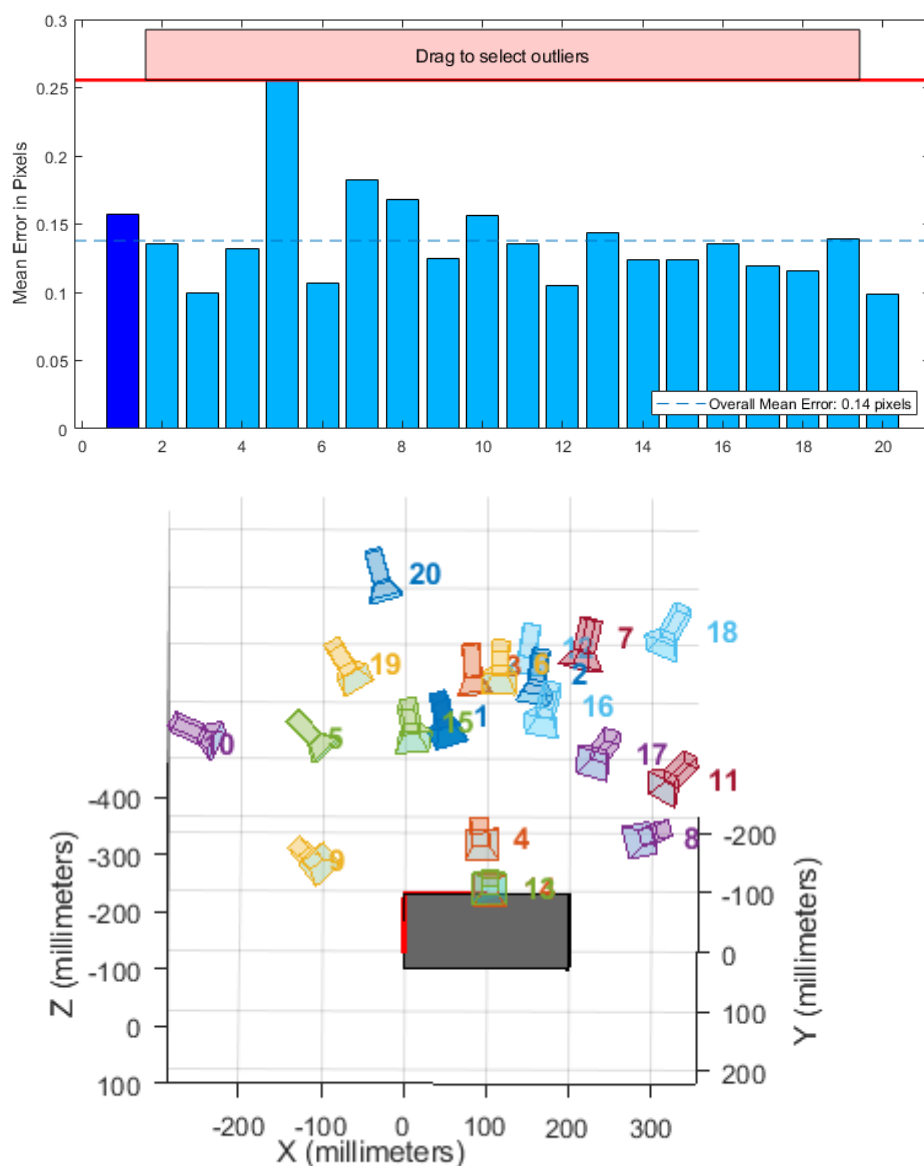


Рисунок 2 – Ошибки проекции и положения камеры

#### 4. Сравнение результатов калибровки разными методами

Количество калибровочных сэмплов равно 20. Для калибровки при помощи шахматной доски использовались одни и те же изображения. В таблицу 1 представлены результаты калибровок, где  $f_x$ ,  $f_y$  – фокусные расстояния в пикселях по оси  $x$  и  $y$  соответственно,  $c_x$ ,  $c_y$  – координаты оптического центра в пикселях,  $k_1$ ,  $k_2$  – коэффициенты радиальной дисторсии,  $p_1$ ,  $p_2$  – коэффициенты тангенциальной дисторсии.

Таблица 1 – Результаты калибровки разными методами

Параметр	Chess OpenCV	Chess Matlab	Aruco OpenCV
$f_x$	486	486	488
$f_y$	485	485	486
$c_x$	318	319	312
$c_y$	230	231	230
$k_1$	0,1303	0,1212	0,0757
$k_2$	-0,2958	-0,1885	-0,2887
$k_3$	0,0001	0	0,0037
$p_1$	-0,0006	0	0,0074
$p_2$	0,2979	0	0,1613
средняя ошибка перепроекции	0,7514	0,1379	4,9906

Судя по средней ошибке перепроекции, наиболее точным методом калибровки является калибровка при помощи шахматной доски в Calibration Toolbox (Matlab).

## 5. Дополненная реальность

Алгоритм метода, рисующего куб на Aruco-маркере:

- 1) Загрузить параметры камеры из файла.
- 2) Задать параметры детекции маркеров.
- 3) Загрузить кадр с камеры.
- 4) Обнаружить маркер при помощи встроенной функции `detectMarkers()`.
- 5) Вычислить положение маркеров относительно камеры при помощи встроенной функции `estimatePoseSingleMarkers()`.
- 6) Задать 3D-координаты куба, учитывая, что система координат маркера находится в его центре.
- 7) Спроецировать 3D-координаты в 2D-координаты на изображении при помощи встроенной функции `projectPoints()`.
- 8) Нарисовать куб при помощи функции `Aruco::drawCube()`.

Также для выделения видимых граней на кубе перед п. 8 определяется положение куба на изображении и, в зависимости от ориентации, окрашиваются те или иные грани куба.

Пример результата выполнения функции представлен на рисунке 3.

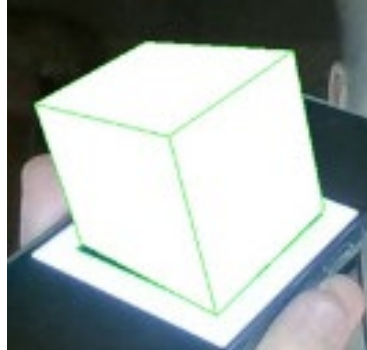


Рисунок 3 – Визуализация куба в зависимости от положения маркера

## **Вывод**

Реализованы и выполнены методы калибровки камеры: при помощи шахматной доски, при помощи доски Aruco. Выполнена калибровка при помощи шахматной доски в Calibration Toolbox (Matlab). Наиболее точным методом оказался Calibration Toolbox (Matlab). Реализован метод генерации Aruco-маркера. Реализован метод дополненной реальности, который рисует куб с основанием в виде маркера, а также окрашивает видимые грани в белый цвет.