Лабораторные работы по курсу

Моделирование ч.3 «Компьютерное зрение»

Лабораторная работа 1

Калибровка камеры и измерение геометрических фигур

2021

Содержание

Введение

Целью лабораторной работы является ознакомление учащихся с базовыми методами разработки алгоритмов компьютерного зрения. Первая часть лабораторной работы заключается в демонстрации процедуры калибровки камеры для закрепления ранее полученного теоретического материала. Вторая часть лабораторной работы представляет собой тестовые задачи по измерению параметров простых геометрических фигур на фотографии с учетом ранее полученных результатов калибровки камеры.

Лабораторная работа содержит два задания для самостоятельного выполнения. Первое задание обязательно для выполнения. Второе задание является заданием повышенной сложности.

Лабораторная работа рассчитана на 4 академических часа.

1. Калибровка камеры

1.1. Задача калибровки

Модель камеры представляет собой установленную зависимость между положением точки в трехмерном пространстве в мировой системе координат и ее положением на изображении в координатах изображения. Данная зависимость может быть описана в линейной форме [1] как

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \cdot \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

где:

- $(u, v, 1)$ – положение точки в координатах изображения (пикселы), записанное в однородной форме,
- (x, y, z) – положение точки в пространстве в мировой системе координат (X, Y, Z),
- $\mathbf{K}$ – матрица внутренних параметров камеры размером 3×3,
- $\begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix}$ – матрица поворота и переноса мировой системы координат (X, Y, Z) в систему координат камеры (X_c, Y_c, Z_c).

Матрица внутренних параметров камеры $\mathbf{K}$ представляет собой:

$$\mathbf{K} = \begin{bmatrix} \dfrac{f}{z_c \cdot \Delta_u} & 0 & \dfrac{u_{center}}{z_c} \\ 0 & \dfrac{f}{z_c \cdot \Delta_v} & \dfrac{v_{center}}{z_c} \\ 0 & 0 & \dfrac{1}{z_c} \end{bmatrix},$$

где:

- f – фокусное расстояние (в миллиметрах),
- Δ_u, Δ_v – размеры пикселя светочувствительной матрицы по горизонтали и вертикали соответственно (в миллиметрах),
- z_c – координата точки по оси Z (удаленность) в системе координат камеры,
- (u_{center}, v_{center}) – положение оптической оси в координатах изображения.

Матрица $[\mathbf{R} \quad \mathbf{T}]$ часто называется матрицей внешних параметров. Она состоит из матрицы $\mathbf{R}$ размером 3×3 для поворота мировой системы координат (X, Y, Z) в систему координат камеры (X_c, Y_c, Z_c) и матрицы $\mathbf{T}$ размером 3×1 для переноса мировой системы координат (X, Y, Z) в систему координат камеры (X_c, Y_c, Z_c). Совокупно матрица $[\mathbf{R} \quad \mathbf{T}]$ определяется шестью неизвестными: тремя углами поворота (каждый угол поворота вокруг своей оси) и тремя неизвестными для линейного перемещения вдоль каждой оси.

Помимо линейной зависимости преобразования координат зачастую модель камеры так же учитывает дисторсию оптической системы, за счет которой положение точки на фокальной плоскости отклоняется от предполагаемого:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = Dist(\begin{bmatrix} u \\ v \end{bmatrix}, \bar{D}),$$

где (u_d, v_d) – смещенные за счет дисторсии координаты точки (u, v) в плоскости изображения, $Dist$ – общая функциональная зависимость смещения координат, $\bar{D}$ – некоторый вектор параметров, присущий конкретной камере.

Задача калибровки камеры заключается в определении значений матрицы внутренних параметров камеры $\mathbf{K}$ и вектора параметров дисторсии $\bar{D}$. Их учет позволяет оперировать реальными физическими размерами проекций объектов на изображении и нивелировать особенности конкретного экземпляра камеры на формирование изображения. При этом для процесса калибровки камеры матрица

$[\mathbf{R} \quad \mathbf{T}]$, которая является как правило неизвестной, выступает в качестве помехового воздействия, снижая точность определения значений $\mathbf{K}$ и $\bar{D}$ или усложняя методику калибровки.

1.2. Модель дисторсии

Общая функциональная зависимость смещения координат $Dist$ в результате действия дисторсии оптической системы является достаточно сложной функцией. Она практически незаметна вблизи оптической оси и сильно проявляется при удалении от нее. Разделяют два типа дисторсий: радиальную и тангенциальную (Рисунок Рисунок 1).

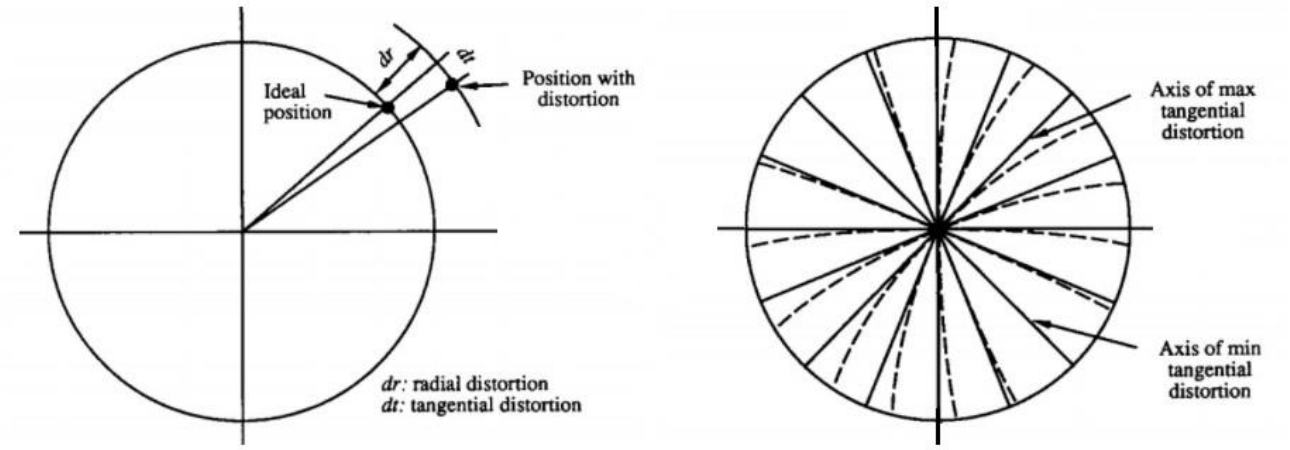

Рисунок 1. Радиальная (слева) и тангенциальная (справа) дисторсии

Радиальная дисторсия возникает в результате не идеальности формы линзы и сильно проявляется по краям изображения. Тангенциальная дисторсия возникает за счет погрешности в параллельности установки линзы и светочувствительной матрицы. Как правило, радиальная дисторсия вносит больший вклад в искажение изображения, чем тангенциальная дисторсия.

Наиболее часто для описания радиальной и тангенциальной дисторсий используется модель *Brown–Conrady* [2]:

$$\begin{cases} x_d = x_u(1+k_1r^2+k_2r^4+k_3r^6+\ldots)+(p_2(r^2+2x_u^2)+2p_1x_uy_u)(1+p_3r^4+p_4r^4+\ldots) \\ y_d = y_u(1+k_1r^2+k_2r^4+k_3r^6+\ldots)+(p_1(r^2+2y_u^2)+2p_2x_uy_u)(1+p_3r^4+p_4r^4+\ldots) \end{cases}$$

где (x_d, y_d) – искаженные дисторсией координаты точки (x_u, y_u), r – расстояние от оптического центра (c_x, c_y) до точки (x_u, y_u):

$$r = \sqrt{(x_u - c_x)^2 + (y_u - c_y)^2},$$

k_i – коэффициенты радиальной дисторсии, p_i – коэффициенты тангенциальной дисторсии.

На практике коэффициенты k_i используют до k_3. Коэффициенты p_i используются до p_2 или принимают равными нулю ввиду малости итогового вклада тангенциальной дисторсии. В результате в общем виде вектор дисторсии $\bar{D}$ состоит из:

$$\bar{D} = \begin{bmatrix} k_1 & k_2 & k_3 & p_1 & p_2 \end{bmatrix}$$

1.3. Методы калибровки камеры

Общий метод калибровки камеры заключается в получении одной или нескольких фотографий известных шаблонов при полностью или частично известных условиях съемки. В качестве шаблона, как правило, выступает черно-белое изображение шахматной доски или массива кругов на черном фоне (Рисунок Рисунок 2) ввиду легкости их алгоритмической обработки.

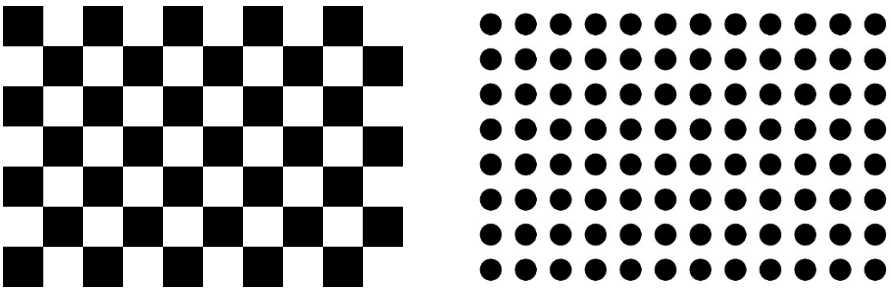

Рисунок 2. Шаблоны для калибровки камеры

Пересечения клеток шахматной доски или черные точки выступают в качестве особых точек на изображении, по которым в дальнейшем и проводится калибровка.

За счет того, что исходно известны расстояния между особыми точками (или они могут быть измерены с помощью линейки при необходимости), матрица внутренних параметров $\mathbf{K}$ может быть вычислена с помощью решения переопределенной системы уравнений (количество точек значительно превосходит количество неизвестных параметров матрицы). За счет расположения особых точек по всему полю изображения, по данным шаблонам так же могут быть найдены коэффициенты вектора дисторсии $\bar{D}$.

Для исключения влияния матрицы внешних параметров $\begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix}$ рассмотрим два способа калибровки: по одиночному снимку при известном взаимном расположении камеры и шаблона или по серии снимков при неизвестном взаимном расположении камеры и шаблона.

1.3.1. Калибровка камеры по одиночному снимку

При калибровке камеры по одиночному снимку целесообразно делать фотографию шаблона в условиях, когда плоскость камеры располагается параллельно плоскости изображения шаблона, оптическая ось камеры направлена в центр шаблона и изображение шаблона на фотографии имеет максимально возможный размер (Рисунок Рисунок 3).

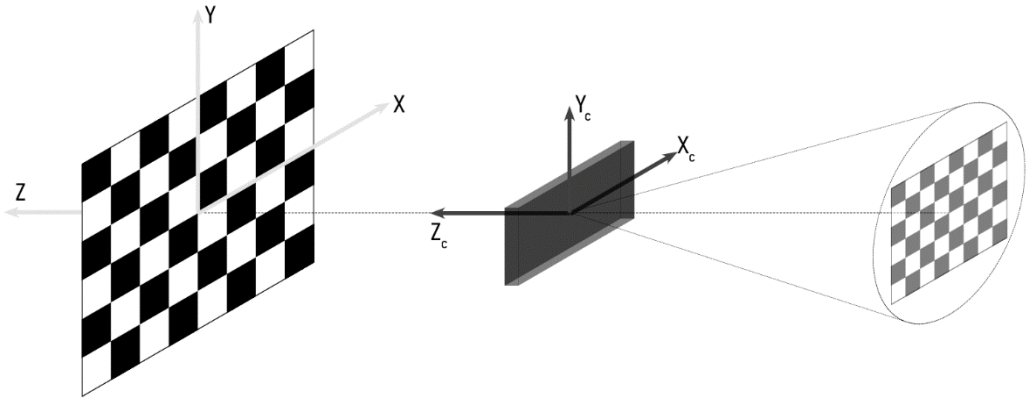

Рисунок 3. Расположение шаблона и камеры

На Рисунке Рисунок 4 приведен пример снимка шаблона, сделанный в данных условиях. Изображение шаблона было выведено на монитор.

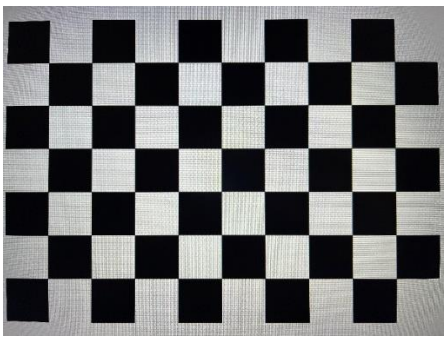

Рисунок 4. Пример фотографии шаблона

За счет такого расположения система координат камеры совмещается с мировой системой координат по трем углам и двум параметрам смещения. Оставшийся параметр смещения представляет собой расстояние межу камерой и плоскостью изображения шаблона. В результате матрица внешних параметров $[\mathbf{R} \quad \mathbf{T}]$ приобретает вид

$$[\mathbf{R} \quad \mathbf{T}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \end{bmatrix},$$

где d - расстояние между камерой и плоскостью изображения, которое может быть измерено с помощью линейки.

Недостатками такого способа является практическая сложность ручного взаимного расположения камеры и шаблона и измерения расстояния d. Погрешности будут негативно сказываться на получаемых параметрах внутренней матрицы калибровки $\mathbf{K}$.

1.3.2. Калибровка камеры по серии снимков

Данный способ был описан Zhengyou Zhang в [3]. Он заключается в создании серии снимков шаблона с разных ракурсов. Каждый ракурс снимка будет описываться своей матрицей внешних параметров $[\mathbf{R}\ \mathbf{T}]$, что позволяет с помощью алгоритмической обработки оценить значения не только внутренней матрицы $\mathbf{K}$, но и матрицы внешних параметров $[\mathbf{R}\ \mathbf{T}]$. Для работы алгоритма необходимо как минимум два снимка с разных ракурсов, однако на практике желательно использовать 5-7 снимков.

Данный алгоритм реализован в качестве штатного средства калибровки камеры в библиотеке компьютерного зрения OpenCV [4].

2. Измерение объектов

2.1. Последовательность действий

При алгоритмической обработке изображений с целью измерения параметров известных объектов (геометрических фигур) общая последовательность действий состоит из этапов, изображенных на Рисунке Рисунок 5.

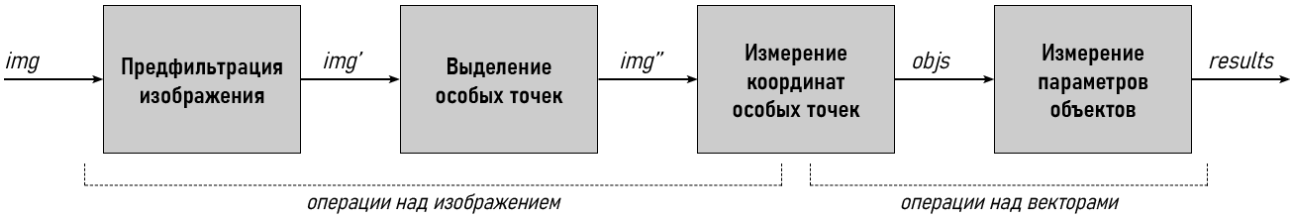

Рисунок 5. Последовательность действий для измерения параметров объектов

Первое действие заключается в фильтрации шумов на входном изображении *img* с целью подавления высокочастотных или низкочастотных помех, мешающих поиску особых точек. При удовлетворительном качестве входного изображения данный этап может быть пропущен.

Второе действие заключается в выделении особых точек на входном изображении *img'*. При обработки геометрических фигур особыми точками выступают углы фигуры. Задачей даного этапа является максимизация сигнала в пикселах особых точек по отношению к общему фону.

Измерение координат особых точек заключается в получении набора векторов *objs*, описывающих координаты (u,v) положений особых точек в плоскости изображения *img''*. После данного этапа последующие операции производятся над векторами.

Заключительное действие заключается в измерении требуемых параметров фигуры (физический размер, периметр, площадь и т.д.) на основе расположения особых точек и параметров матриц $\mathbf{K}$, $[\mathbf{R} \quad \mathbf{T}]$ и вектора дисторсии $\bar{D}$.

2.2. Операции над изображением

Первичная фильтрация изображения производится с помощью двумерной свертки с ядром согласованного фильтра. Для реализации фильтра низких частот в качестве ядра должна быть использована выпуклая функция, повторяющая форму сигнала, например

$$h = \frac{1}{35} \cdot \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Сумма всех элементов ядра фильтра должна быть равна единицы для сохранения уровня яркости изображения.

Для реализации фильтра высоких частот применяют свертку с ядром, в котором центральный элемент больше нуля, остальные элементы меньше нуля, а общая сумма элементов ядра равна единице. Например,

$$h = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix}.$$

Как правило, выпуклая функция для ядра свертки выбирается симметричной по вертикальной и горизонтальной осям. Размер апертуры фильтра выбирается исходя из характерного размера особых точек. Так же размер апертуры выбирается нечетным по вертикальной и горизонтальной осям для отсутствия смещения изображений объектов. Характерные размеры апертуры 3×3, 5×5, 7×7 и т. д.

Для выделения особых точек (для геометрической фигуры это угол) применяют дифференциальные операторы, которые максимизируют сигнал от резкого перепада яркости на границе между углом фигуры (особой точкой) и фоном. Для простого бинарного изображения, в котором фон и заливка фигуры представлены константной (квазиконстантной) яркостью дифференциальный оператор может иметь вид,

$$h = \frac{1}{S_h} \cdot \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & 1 \\ 1 & 1 & 1 & 1 & \dots & 1 \\ -1 & \dots & -1 & 1 & \dots & 1 \\ -1 & \dots & \dots & 1 & \dots & 1 \\ -1 & -1 & -1 & 1 & \dots & 1 \end{vmatrix},$$

где S_h – общая сумма элементов h, учет которой необходим для сохранения яркости изображения. Расположению области «-1» соответствует область угла фигуры, область «1» соответствует фону, при условии, что заливка фигуры темнее фона. Если заливка фигуры светлее фона, то знак у элементов h необходимо поменять.

Размер апертуры дифференциального оператора h выбирается исходя из характерного размера угла фигуры.

2.3. Операции над объектами

После обнаружения особых точек входные данные представлены массивом объектов

$$obj_i = (u_i, y_i), \qquad i = 0,1,\dots, N,$$

где (u_i, y_i) координаты особой точки в плоскости изображения, N - общее количество особых точек.

Углы направления на точку (u_i, y_i) в градусах относительно оптической оси камеры рассчитывается как

$$\begin{cases} \alpha_i = \frac{180^\circ}{\pi} \cdot arctg\left(\frac{(u_i - u_{center}) \cdot \Delta_u}{f}\right) \\ \beta_i = \frac{180^\circ}{\pi} \cdot arctg\left(\frac{(v_i - v_{center}) \cdot \Delta_v}{f}\right) \end{cases},$$

где Δ_u, Δ_v – размеры пикселя светочувствительной матрицы по горизонтали и вертикали соответственно (в миллиметрах), f - фокусное расстояния по горизонтальной и вертикальной осям (в миллиметрах).

Расстояние d_{ij} рассчитывается между двумя точками obj_i и obj_j как:

$$d_{ij} = \sqrt{\left(u_i - u_j\right)^2 + \left(v_i - v_j\right)^2}$$

Физический размер D_{ij} расстояния d_{ij} между двумя равно удаленными от камеры точками по оси *Z* для квадратного пиксела может быть найдено с помощью

$$D_{ij} = \frac{z_c}{f} d_{ij} \cdot \Delta,$$

где Δ – размер пиксела, z_c – расстояние до точек в системе координат камеры.

3. Программные инструменты

3.1. Язык программирования *python*

В лабораторной работе используется язык программирования *python* [5] версии 3.9.7. Официальная документация на данный язык программирования доступна через Интернет и расположена по адресу [6]. Среди документации так же содержится официальный учебник по программированию [7].

Для разработки программного кода в лабораторной работе используется среда разработки *PyCharm IDE Community Edition* версии 2021.2. Данная версия среды распространяется бесплатно и может быть загружена с официального сайта [8]. Документация по обучению работе в *PyCharm IDE* доступна на официальном сайте по адресу [9].

3.2. Библиотеки

Для разработки программного кода в лабораторной работе используются следующие основные библиотеки.

3.2.1. *OpenCV*

OpenCV (*Open Source Computer Vision Library*) является широко применяемой бесплатной библиотекой с открытым исходным кодом, содержащей в себе множество алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения [4]. Документация для данной библиотеки доступна на ее официальном сайте по адресу [10]. В лабораторной работе будет использована версия 4.5.3.56 данной библиотеки.

Для работы с библиотекой *OpenCV* на языке *python* 3.9.7 необходимо установить пакет «*opencv-python*».

3.2.2. *NumPy*

NumPy является базовой библиотекой с открытым исходным кодом языка *python* для поддержки работы с многомерными массивами и реализации высокоуровневых

математических операций [11]. Документация для данной библиотеки доступна на ее официальном сайте по адресу [12]. В лабораторной работе будет использована версия 1.21.2 данной библиотеки.

Для работы с библиотекой *NumPy* на языке *python* 3.9.7 необходимо установить пакет «*numpy*».

3.2.3. *SciPy*

SciPy является свободно распространяемой библиотекой для языка *python*, предназначенной для выполнения научных и инженерных расчетов [13]. Документация для данной библиотеки доступна на ее официальном сайте по адресу [14]. В лабораторной работе будет использована версия 1.7.1 данной библиотеки.

Для работы с библиотекой *SciPy* на языке *python* 3.9.7 необходимо установить пакет «*scipy*».

3.2.4. *Matplotlib*

Matplotlib является свободно распространяемой библиотекой для языка *python*, предназначенной для визуализации данных [15]. Документация для данной библиотеки доступна на ее официальном сайте по адресу [16]. В лабораторной работе будет использована версия 3.4.3 данной библиотеки.

Для работы с библиотекой *Matplotlib* на языке *python* 3.9.7 необходимо установить пакет «*matplotlib*».

4. Упражнения

4.1. Подготовка данных для калибровки камеры

4.1.1. Откройте файл «*data/Chess board original.png*» с изображением шахматной доски.

4.1.2. Разверните изображение во весь экран (F11).

4.1.3. С помощью линейки измерьте длины ребер клетки шахматной доски SQ_y и SQ_x на изображении.

4.1.4. Сделайте фотографию изображения шахматной доски, стараясь соблюдать параллельность между плоскостью изображения и камерой. Расположите изображение на всей плоскости фотографии по аналогии с Рисунком Рисунок 4.

4.1.5. Измерьте расстояние *D* между камерой и изображением шахматной доски при фотографировании.

4.1.6. Сделайте 5-6 дополнительных снимков изображения шахматной доски с разных ракурсов.

4.1.7. Сохраните сделанные снимки в каталоге «*data/chessboard/*»

4.2. Калибровка камеры

4.2.1. Калибровка камеры по одиночному снимку

4.2.1.1. Откройте в *PyCharm IDE* проект «*camera_meas*»

4.2.1.2. В файле «*main.py*» укажите:

- путь к файлу фотографии, сделанной перпендикулярно плоскости изображения (переменная IMG_FILE);
- длины ребер клетки шахматной доски по вертикали и горизонтали (переменная SQUARE_SIZES_MM)
- расстояние D (переменная DISTANCE_MM)
- размеры пикселя камеры в миллиметрах (переменная PIXEL_SIZES_MM).

4.2.1.3. Изучите исходный код проекта и запустите на исполнение файл *main.py* (*Ctrl+F10*)

4.2.1.4. В ходе выполнения промежуточные результаты будут выводиться в консоль, а обработка данных останавливаться. Для продолжения обработки необходимо нажимать клавишу «*n*».

4.2.1.5. По завершению работы на основе входных данных для камеры будут измерены фокусные расстояния и углы обзора по вертикальной и горизонтальной осям и коэффициенты радиальной дисторсии объектива.

4.2.1.6. Полученные параметры камеры будут сохранены в файл «*/data/camera.txt*»

4.2.2. Калибровка камеры по серии снимков

4.2.2.1. Откройте в *PyCharm IDE* проект «*CameraCalib_OpenCV*», использующий возможности библиотеки *OpenCV* для калибровки камеры.

4.2.2.2. В файле «*main.py*» укажите:

- путь к каталогу «*data/chessboard/*» со снимками шахматной доски в (переменная *IMGs_PATH*);
- размеры пикселя камеры в миллиметрах (переменная PIXEL_SIZES_MM).

4.2.2.3. Изучите исходный код проекта и запустите на исполнение файл *main.py* (*Ctrl+F10*)

4.2.2.4. В ходе работы на экран будут выводиться промежуточные изображения, а обработка данных останавливаться. Для продолжения необходимо нажать любую клавишу.

4.2.2.5. По завершению работы на основе входных данных для камеры будут измерены фокусные расстояния и углы обзора по вертикальной и горизонтальной осям и коэффициенты радиальной дисторсии объектива.

4.2.2.6. Сравните полученные результаты с результатами, полученными в ходе калибровке камеры по одиночному снимку (п. 3.2.1)

4.2.2.7. Полученные параметры камеры будут сохранены в файл «*/data/camera_OpenCV.txt*»

4.3. Пример алгоритма для измерения объекта

4.3.1. Откройте во весь экран изображение с тестовой фигурой «*data/Figures/Figure1.png*» (Рисунок Рисунок 6).

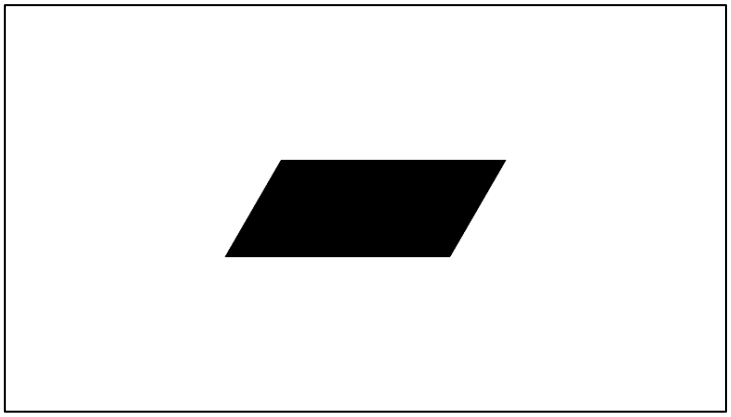

Рисунок 6. Тестовая фигура

4.3.2. Сделайте фотографию изображения, стараясь соблюдать параллельность между плоскостью камеры и плоскостью изображения.
4.3.3. Измерьте расстояние между камерой и изображением при фотографировании.
4.3.4. Откройте в *PyCharm IDE* проект «*obj_meas_example*».
4.3.5. В файле main.py в разделе «0. Параметры» введите необходимые параметры съемки.
4.3.6. Изучите исходный код проекта.
4.3.7. Запустите выполнение проекта (Ctrl+F10).
4.3.8. Сравните выведенные в консоль значения с реальными значениями (для получения реальных значений используйте линейку).

5. Индивидуальные задания

5.1. Задание №1

5.1.1. Подготовка данных

5.1.2. Выберете в директории «*data/Figures*/» файл фигуры с номером *i:*

$$i = (N \,\%\, 5) + 2$$

где *N* – ваш номер в общем списке группы. Например, если *N* равно 17, то *i* будет равно 4.

5.1.3. Откройте файл фигуры в полный экран.
5.1.4. Сделайте фотографию фигуры, соблюдая параллельность между плоскостью камеры и плоскостью изображения.

5.1.5. Подзадание №1

5.1.5.1. Рассчитайте длину каждого ребра фигуры и общий периметр в миллиметрах
5.1.5.2. Рассчитайте угловой размер фигуры по вертикали и горизонтали

5.1.6. Подзадание №2

5.1.6.1. Рассчитайте площадь фигуры.

5.2. Задание №2*

5.2.1. Подготовка данных

5.2.1.1. Откройте файл «*data/Crosses/Crosses.png*» в полный экран.

5.2.2. Сделайте фотографию изображения, соблюдая параллельность между плоскостью камеры и плоскостью изображения.

5.2.2.1. По фотографии вычислите длину кратчайшего пути, включающего в себя все точки.

6. Список литературы

1. Л. Шапиро Д.С. Компьютерное зрение. 3rd ed. Москва: БИНОМ. Лаборатория знаний, 2015.
2. Brown D.C. Close-Range Camera Calibration // Photogramm, No. 8, 1971. pp. 855-866.
3. Zhang Z. A Flexible New Technique for Camera Calibration, December 1998.
4. OpenCV [Электронный ресурс] // OpenCV: [сайт]. URL: https://opencv.org/
5. Python [Электронный ресурс] // Python: [сайт]. URL: https://www.python.org/
6. Python 3.9.7 Documentation [Электронный ресурс] // Python: [сайт]. URL: https://docs.python.org/3.9/
7. The Python Tutorial [Электронный ресурс] // Python: [сайт]. URL: https://docs.python.org/3.9/tutorial/index.html
8. PyCharm IDE [Электронный ресурс] // PyCharm IDE: [сайт]. URL: https://www.jetbrains.com/ru-ru/pycharm/
9. Get started [Электронный ресурс] // PyCharm: [сайт]. URL: https://www.jetbrains.com/help/pycharm/quick-start-guide.html
10. OpenCV modules [Электронный ресурс] // OpenCV: [сайт]. URL: https://docs.opencv.org/4.5.3
11. NumPy [Электронный ресурс] // NumPy: [сайт]. URL: https://numpy.org
12. Overview - NumPy [Электронный ресурс] // NumPy: [сайт]. URL: https://numpy.org/doc/stable
13. SciPy [Электронный ресурс] // SciPy.org: [сайт]. URL: https://www.scipy.org
14. SciPy documentation [Электронный ресурс] // SciPy: [сайт]. URL: https://docs.scipy.org/doc/scipy/reference/
15. Matplotlib: python plotting [Электронный ресурс] URL: https://matplotlib.org/
16. Overview - Matplotlib [Электронный ресурс] // Matplotlib: [сайт]. URL: https://matplotlib.org/stable/contents.html