

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
GRADUATION THESIS**

**Автоматизация калибровки камеры для использования в задачах компьютерного
зрения и дополненной реальности**

Обучающийся / Student Съестов Дмитрий Вячеславович

Факультет/институт/кластер/ Faculty/Institute/Cluster школа разработки видеоигр

Группа/Group J4222

Направление подготовки/ Subject area 09.04.03 Прикладная информатика

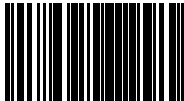
Образовательная программа / Educational program Технологии разработки
компьютерных игр 2022

Язык реализации ОП / Language of the educational program Русский

Квалификация/ Degree level Магистр

Руководитель ВКР/ Thesis supervisor Карсаков Андрей Сергеевич, кандидат технических
наук, Университет ИТМО, школа разработки видеоигр, доцент (квалификационная
категория "ординарный доцент")

Обучающийся/Student

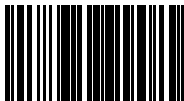
Документ подписан	
Съестов Дмитрий Вячеславович	
24.05.2024	

(эл. подпись/ signature)

Съестов
Дмитрий
Вячеславович

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Карсаков Андрей Сергеевич	
23.05.2024	

(эл. подпись/ signature)

Карсаков
Андрей
Сергеевич

(Фамилия И.О./ name
and surname)

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /
OBJECTIVES FOR A GRADUATION THESIS**

Обучающийся / Student Съестов Дмитрий Вячеславович

Факультет/институт/кластер/ Faculty/Institute/Cluster школа разработки видеоигр

Группа/Group J4222

Направление подготовки/ Subject area 09.04.03 Прикладная информатика

Образовательная программа / Educational program Технологии разработки компьютерных игр 2022

Язык реализации ОП / Language of the educational program Русский

Квалификация/ Degree level Магистр

Тема ВКР/ Thesis topic Автоматизация калибровки камеры для использования в задачах компьютерного зрения и дополненной реальности

Руководитель ВКР/ Thesis supervisor Карсаков Андрей Сергеевич, кандидат технических наук, Университет ИТМО, школа разработки видеоигр, доцент (квалификационная категория "ординарный доцент")

Характеристика темы ВКР / Description of thesis subject (topic)

Тема в области фундаментальных исследований / Subject of fundamental research: нет / not

Тема в области прикладных исследований / Subject of applied research: да / yes

Основные вопросы, подлежащие разработке / Key issues to be analyzed

Требуется разработать алгоритм, позволяющий автоматически калибровать камеру с минимальным участием человека. Алгоритм должен определять как внутренние, так и внешние параметры камеры, не требуя её перемещения и использования физических калибровочных объектов.

Форма представления материалов ВКР / Format(s) of thesis materials:

Отчёт, презентация, программный код

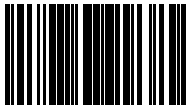
Дата выдачи задания / Assignment issued on: 22.02.2024

Срок представления готовой ВКР / Deadline for final edition of the thesis 24.05.2024

СОГЛАСОВАНО / AGREED:

Руководитель ВКР/
Thesis supervisor

Документ подписан	
----------------------	--

		Карсаков Андрей Сергеевич
Карсаков Андрей Сергеевич		
08.05.2024		

(эл. подпись)

Задание принял к
исполнению/ Objectives
assumed BY

(эл. подпись)

Руководитель ОП/ Head
of educational program

(эл. подпись)

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
SUMMARY OF A GRADUATION THESIS**

Обучающийся / Student Съестов Дмитрий Вячеславович
Факультет/институт/кластер/ Faculty/Institute/Cluster школа разработки видеоигр
Группа/Group J4222
Направление подготовки/ Subject area 09.04.03 Прикладная информатика
Образовательная программа / Educational program Технологии разработки компьютерных игр 2022
Язык реализации ОП / Language of the educational program Русский
Квалификация/ Degree level Магистр
Тема ВКР/ Thesis topic Автоматизация калибровки камеры для использования в задачах компьютерного зрения и дополненной реальности
Руководитель ВКР/ Thesis supervisor Карсаков Андрей Сергеевич, кандидат технических наук, Университет ИТМО, школа разработки видеоигр, доцент (квалификационная категория "ординарный доцент")

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
DESCRIPTION OF THE GRADUATION THESIS**

Цель исследования / Research goal

Разработка алгоритма, позволяющего калибровать внутренние и внешние параметры камеры в автоматическом режиме

Задачи, решаемые в ВКР / Research tasks

Анализ существующих алгоритмов калибровки, формализация предложенного алгоритма, разработка прототипной реализации, экспериментальное исследование прототипа, оценка результатов экспериментов и определение потенциала для дальнейшего улучшения разработки

Краткая характеристика полученных результатов / Short summary of results/findings

Предложен алгоритм, позволяющий с минимальным участием человека и без использования физического шаблона калибровать внутренние параметры камеры, а также nodal offset для синхронизации перспектив физической и виртуальной камер в сфере virtual production. Для алгоритма разработана тестовая реализация, проведено её экспериментальное исследование.

Обучающийся/Student

Документ подписан	
Съестов Дмитрий Вячеславович	

Съестов
Дмитрий

24.05.2024	
------------	--

(эл. подпись/ signature)

Вячеславович

(Фамилия И.О./ name and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Карсаков Андрей Сергеевич	
23.05.2024	

(эл. подпись/ signature)

Карсаков
Андрей
Сергеевич

(Фамилия И.О./ name and surname)

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	7
ВВЕДЕНИЕ.....	9
1 ОБЗОР СУЩЕСТВУЮЩИХ АЛГОРИТМОВ КАЛИБРОВКИ.....	11
1.1 Математическая модель камеры.....	11
1.2 Классические алгоритмы калибровки камеры.....	15
1.3 Ошибка калибровки и способы её уменьшения.....	17
2 ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ.....	20
2.1 Калибровка внутренних параметров.....	20
2.2 Калибровка nodal offset.....	23
3 РАЗРАБОТКА РЕАЛИЗАЦИИ АЛГОРИТМА.....	28
3.1 Описание архитектуры прототипа.....	28
3.2 Разработка прототипа.....	29
3.3 Описание API.....	34
4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМА.....	39
4.1 Описание проведённых экспериментов.....	39
4.2 Результаты экспериментов.....	40
4.3 Анализ результатов.....	41
4.4 Потенциальные улучшения.....	43
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	46

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Виртуальная камера – программный объект, симулирующий процесс получения снимков объектов с помощью физической камеры

Внешние параметры камеры – параметры, связанные с положением камеры в пространстве

Внутренние параметры камеры – параметры, связанные со внутренним устройством камеры

Входной зрачок – точка, в которой фокусируются лучи, попадающие в объектив

Дисторсия – искажение снимка, вызванное объективом и проявляющееся в искривлении прямых линий

Калибровка – вычисление параметров камеры по её снимкам

Калибровочный шаблон – изображение с характерной геометрией, используемое для калибровки камеры

Компьютерное зрение – дисциплина, связанная с принятием решений об объектах по их изображениям

Однородные координаты – система координат, дополненная одним фиктивным измерением

Ошибка репроекции – разница между реальной и идеальной проекциями объекта

Принципиальная точка – точка, в которой оптическая ось камеры пересекает плоскость изображения

Трекер – устройство, транслирующее в игровой движок свою физическую позицию

Фокусное расстояние – внутренний параметр камеры, определяющий увеличение и угол обзора

Nodal offset – смещение между виртуальной камерой и входным зрачком физической камеры

OpenCV – свободная библиотека, реализующая алгоритмы компьютерного зрения

Virtual production – технология, основанная на совмещении физических объектов и виртуального фона, отрисованного в реальном времени с помощью игрового движка

ВВЕДЕНИЕ

В широком смысле *компьютерное зрение* – это технологии, позволяющие принимать решения о физических объектах, основываясь на их изображениях. Дисциплина компьютерного зрения применяется в робототехнике, при анализе изображений, в системах видеонаблюдения и т.п.

Одним из частных случаев применения компьютерного зрения является дополненная реальность – технологии, позволяющие внедрять дополнительную информацию в зрительное поле человека, тем самым смешивая виртуальные и реально существующие объекты. В современном кинематографе находит всё более широкое распространение технология *virtual production*, которая позволяет в реальном времени отрисовывать виртуальную сцену за актёрами, тем самым устраняя необходимость в добавлении фона на этапе монтажа. Для решения подобных задач в первую очередь необходимо выполнить калибровку камеры.

Калибровкой камеры называется задача получения внутренних и внешних параметров камеры по отснятым ей кадрам. Калибровка позволяет определить преобразования, связывающие мировую систему координат с системой координат на плоском изображении. Иными словами, она определяет геометрические характеристики процесса создания изображения. В ходе калибровки вычисляются коэффициенты, позволяющие исправить привносимые линзой искажения и синхронизировать движение физической камеры и движение виртуальной камеры, работающей внутри игрового движка.

Существует ряд алгоритмов для калибровки камеры, как с участием человека, так и полуавтоматической. Реализации подобных алгоритмов можно найти в OpenCV – библиотеке, широко используемой в задачах компьютерного зрения. OpenCV распространяется по лицензии BSD, таким образом, она доступна для свободного использования как в академических, так и в коммерческих целях [1]

Цель данной работы заключается в разработке метода автоматической калибровки камеры с минимальным участием человека, который может быть использован для задач virtual production в Unreal Engine или другом игровом движке. Актуальность работы обоснована тем, что существующие инструменты работают только в полуавтоматическом режиме и требуют использования специального оборудования и участия человека в процессе калибровки.

1 ОБЗОР СУЩЕСТВУЮЩИХ АЛГОРИТМОВ КАЛИБРОВКИ

1.1 Математическая модель камеры

Прежде чем рассматривать алгоритмы калибровки камеры, проанализируем её модель и параметры, определяемые процессом калибровки.

Первой камерой в истории была камера-обскура (pinhole camera) – простейшее устройство в виде светонепроницаемого ящика с маленьким отверстием и экраном напротив него, известное с V-IV веков до н.э. Проходя через отверстие, лучи света, отражённые от внешнего объекта, создают на экране его перевёрнутое изображение (рис. 1). Хотя современные камеры устроены намного сложнее и оснащены линзами, модель камеры-обскуры успешно применяется в задачах компьютерного зрения [2]

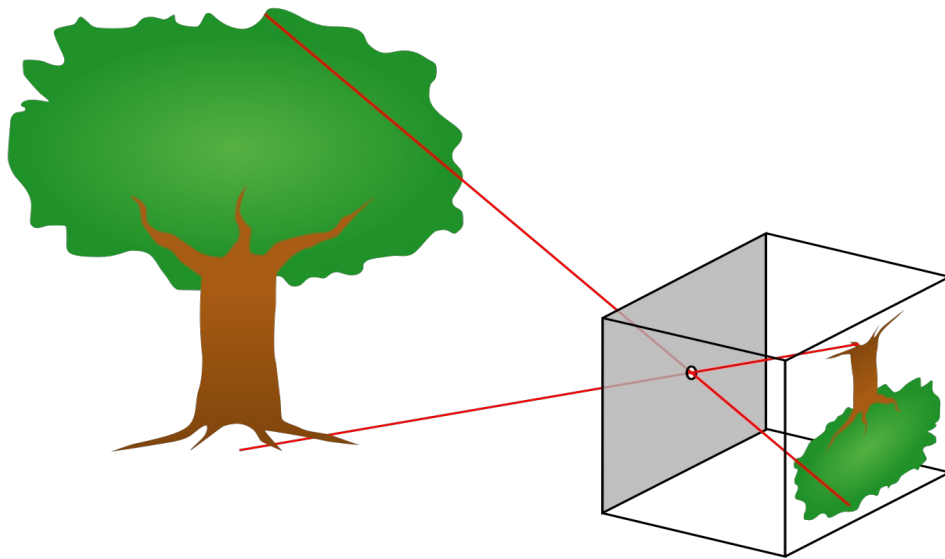


Рисунок 1 – Принцип работы камеры-обскуры

Проекция точки в пространстве соответствует точке пересечения луча, соединяющего её с центром проекции, и плоскости изображения. Геометрия проекции трёхмерного пространства на плоскость изображения в модели камеры-обскуры показана на рисунке 2.

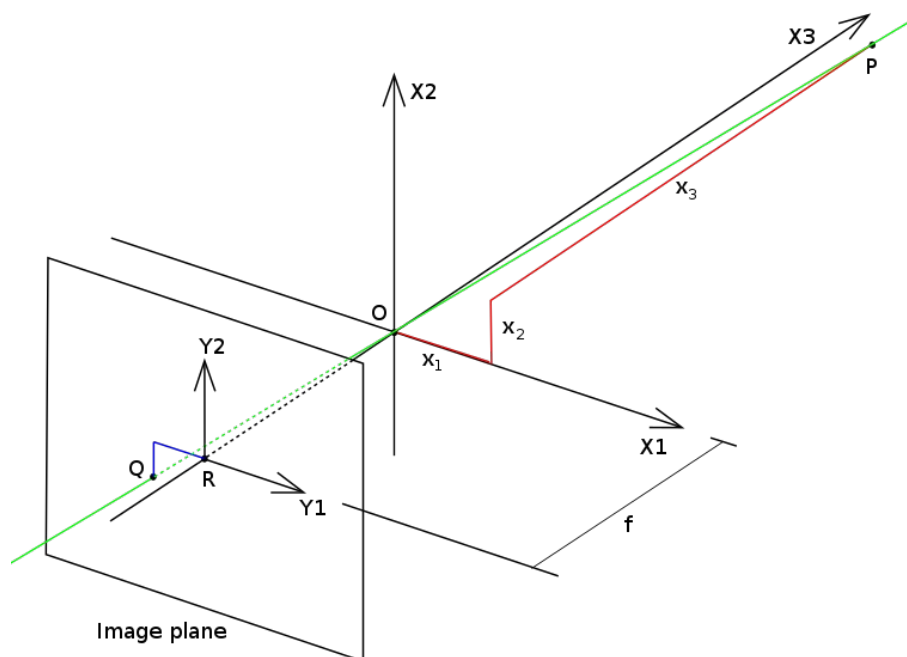


Рисунок 2 – Геометрическое устройство камеры-обскуры

В компьютерной графике используются *однородные координаты* – система координат, дополненная одним фиктивным измерением сверх декартовых координат точки. Так, координаты точки на плоскости изображения задаются вектором $(u, v, 1)$, а мировые координаты – $(x, y, z, 1)$. Однородные координаты удобны тем, что упрощают некоторые формулы и позволяют выражать бесконечно удалённые точки в конечных координатах.

В модели камеры-обскуры преобразование координат можно описать следующей формулой:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = K [R T] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (1)$$

где z_c – коэффициент масштабирования, M – матрица камеры, K – матрица *внутренних параметров*, R и T – *внешние параметры камеры*.

К внутренним параметрам относятся те параметры, которые связаны только с устройством самой камеры. Поэтому внутренние параметры не

меняются во время съёмки, а их калибровка производится только при изменении настроек камеры.

Матрица внутренних параметров имеет вид [2,3]

$$K = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

где f_x и f_y – фокусные расстояния по каждой координате, c_x и c_y – координаты *принципиальной точки* (т.е. точки, в которой оптическая ось пересекает плоскость изображения. Принципиальная точка совпадает с центром кадра, но может быть смещена из-за искажений), а γ – угол наклона пикселя.

Как правило, в реальных камерах угол наклона пикселя равен нулю, а высота и ширина пикселя одинаковы. Исходя из этого, в алгоритмах калибровки матрица (2) часто упрощается до следующего вида: [4]

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Внешние параметры камеры – это параметры, задающие положение камеры сцены. К внешним параметрам относятся матрица вращения R и вектор переноса T . В отличие от внутренних параметров, внешние параметры требуют калибровки для каждого отдельного снимка.

Как было сказано выше, камера-обскура – это идеальная модель, в которой не учитываются различные искажения, внесённые объективом. Наиболее значительное искажение, при котором коэффициент линейного увеличения изменяется по мере удаления от оптической оси, называется *дисторсией*. Поскольку дисторсия нарушает подобие между трёхмерным объектом и его изображением, её необходимо исправить, прежде чем выполнять калибровку внешних параметров.

Самой распространённой моделью дисторсии является модель Брауна-Конради [5,6], по которой выделяется две компоненты дисторсии: *радиальная* и *тангенциальная*.

Радиальная дисторсия вызвана сферической формой линзы и проявляется в искривлении прямых линий, в результате чего они перестают быть параллельными. Различают *подушкообразную дисторсию*, при которой расстояние от оптического центра увеличивается по мере удаления, и *бочкообразную дисторсию*, при которой оно уменьшается (рис. 3).

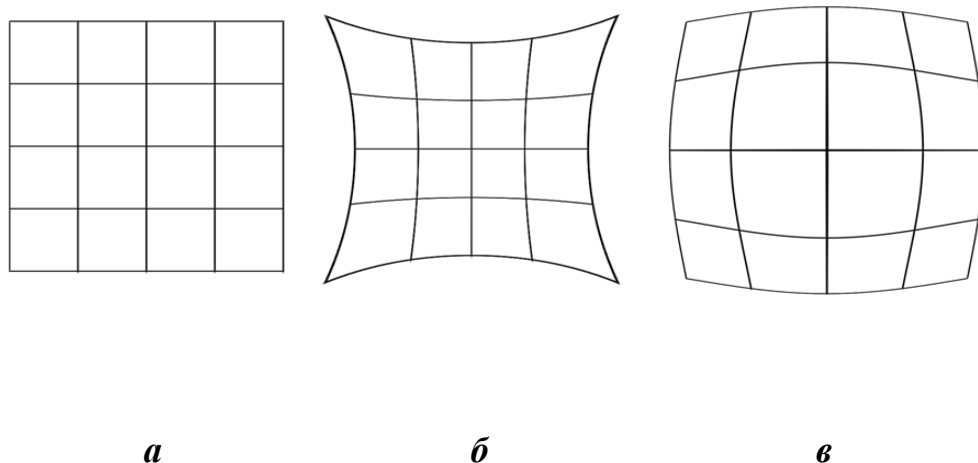


Рисунок 3 – Виды радиальной дисторсии: а – изображение без дисторсии; б – подушкообразная (положительная) дисторсия; в – бочкообразная (отрицательная) дисторсия

Тангенциальная дисторсия обусловлена неправильным положением объектива относительно плоскости изображения. Она проявляется в наклоне прямых линий, вследствие которого они кажутся искривлёнными. Как правило, радиальная дисторсия превосходит тангенциальную.

Дисторсия изображения в модели Брауна-Конради описывается следующей формулой: [7]

$$\begin{cases} x_d = x_u(1 + k_1 r^2 + k_2 r^4 + \dots) + (p_2(r^2 + 2x_u^2) + 2p_1 x_u y_u)(1 + p_3 r^2 + p_4 r^4 + \dots) \\ y_d = y_u(1 + k_1 r^2 + k_2 r^4 + \dots) + (p_1(r^2 + 2y_u^2) + 2p_2 x_u y_u)(1 + p_3 r^2 + p_4 r^4 + \dots) \end{cases} \quad (4)$$

где (x_d, y_d) , (x_u, y_u) – координаты точки с дисторсией и без неё соответственно, r – расстояние от точки (x_u, y_u) до оптического центра изображения (c_x, c_y) , k_i и p_i – коэффициенты радиальной и тангенциальной дисторсии соответственно.

На практике при решении задачи калибровки вычисляются только 2 коэффициента тангенциальной дисторсии и от 3 до 6 коэффициентов радиальной. Таким образом, формула дисторсии (4) приобретает следующий упрощённый вид:

$$\begin{cases} x_d = x_u(1 + k_1 r^2 + k_2 r^4 + \dots) + (p_2(r^2 + 2x_u^2) + 2p_1 x_u y_u) \\ y_d = y_u(1 + k_1 r^2 + k_2 r^4 + \dots) + (p_1(r^2 + 2y_u^2) + 2p_2 x_u y_u) \end{cases} \quad (5)$$

1.2 Классические алгоритмы калибровки камеры

Теперь, когда мы проанализировали цели калибровки и модель камеры, можно рассмотреть некоторые традиционные подходы к решению задачи калибровки. Существует два наиболее распространённых метода калибровки: классический двухэтапный алгоритм Roger Y. Tsai [8] и “Новая гибкая технология калибровки камеры”, предложенная Zhengyou Zhang [9]. Рассмотрим каждый из них поподробнее.

Классический алгоритм Roger Y. Tsai состоит из двух этапов: на первом определяются параметры внешней калибровки, а на втором – параметры внутренней. В качестве шаблона калибровки обычно используются две или три ортогональные плоскости (рис. 4).

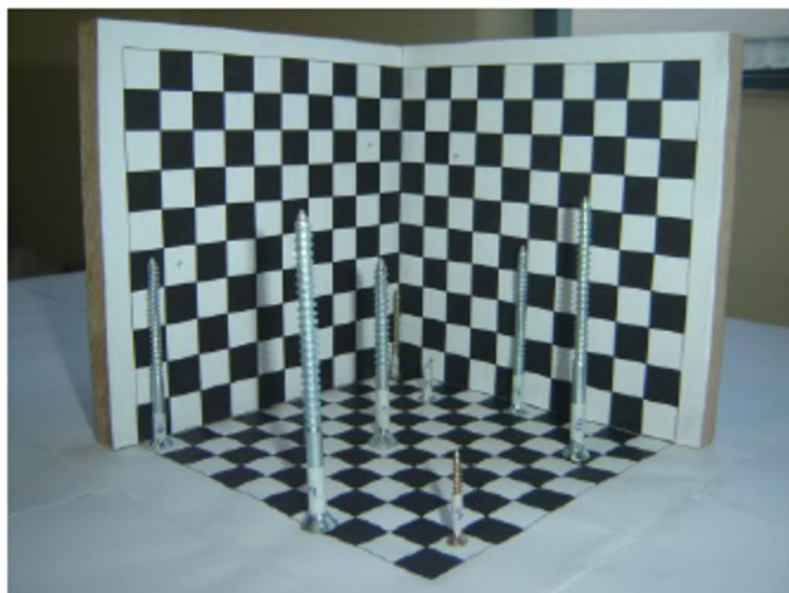


Рисунок 4 – Шаблон калибровки в алгоритме Roger Y. Tsai

На первом этапе алгоритма выполняется вычисление параметра R и первых двух компонент вектора смещения: T_x и T_y . Кроме того, получается оценка фокусного расстояния f и третьего компонента вектора смещения T_z . Для этого необходимо решить систему линейных уравнений, связывающую координаты точек калибровочного шаблона на изображении и в реальном мире. На втором этапе выполняется градиентный поиск. Это позволяет определить фактор радиальной дисторсии, а также скорректировать оценку фокусного расстояния f и смещение вдоль оси z T_z , оценённые на первом этапе. Ограничение этого метода заключается в том, что современные камеры имеют линзы, способные автоматически корректировать фокус. Вследствие этого внутренние параметры камеры могут меняться во время съёмки, на что алгоритм Tsai не рассчитан [10]

Другой популярный подход, предложенный в 1998 году Zhengyou Zhang, использует модель камеры-обскуры и несколько (не менее трёх) изображений калибровочного шаблона, в качестве которого обычно используется шахматная доска (рис. 5). Важной особенностью такой доски является разность длин её сторон, позволяющая однозначно определить

положение доски в пространстве. По изображениям калибровочного шаблона составляется система уравнений, решив которую, можно вывести внутренние параметры [11]

Алгоритм Zhengyou Zhang получил широкое распространение за счёт надёжности и простоты создания шаблона калибровки, который представляет собой обычное плоское изображение. В частности, он применяется при калибровке камер марсоходов NASA и сенсоров Kinect [12] Кроме того, именно этот подход используется в реализации алгоритма калибровки в библиотеке OpenCV, которая является стандартом в сфере компьютерного зрения [13].

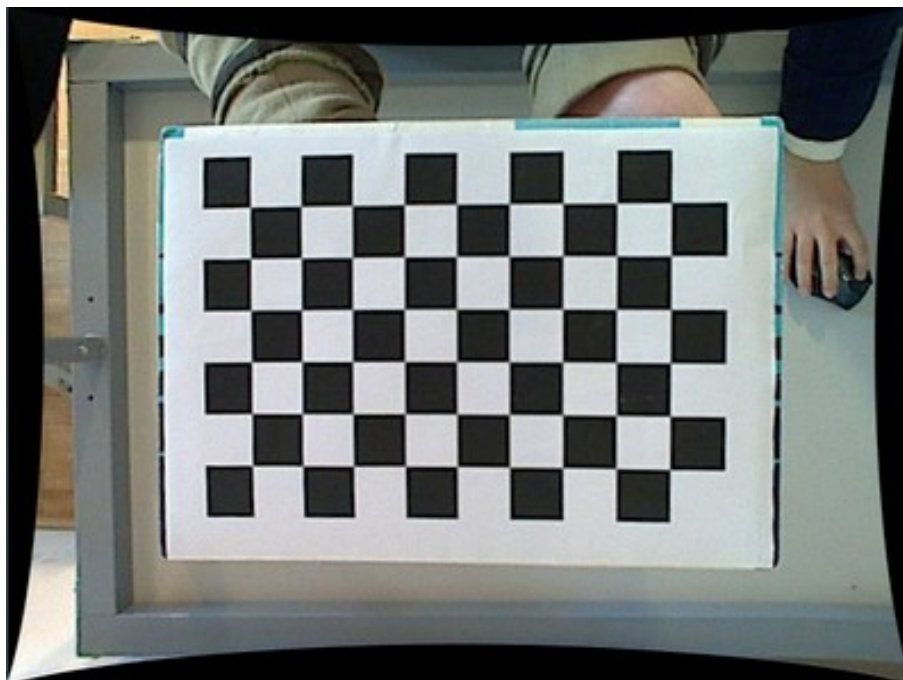


Рисунок 5 – Калибровочный шаблон, используемый в алгоритме Zhang

Недостаток алгоритма Zhang заключается в его более низкой производительности по сравнению с предыдущим алгоритмом. С другой стороны, он также более точен, чем подход, предложенный Roger Y. Tsai [14].

1.3 Ошибка калибровки и способы её уменьшения

При калибровке имеет место *ошибка репроекции* – разница между идеальной и реальной проекциями [15]. Она вычисляется по следующей формуле:

$$\sqrt{\frac{\sum_i (dist(x_i, \hat{x}_i))^2}{n}}, \quad (6)$$

где *dist* - функция Евклидова расстояния, x_i и \hat{x}_i - координаты точки на исходном снимке и репроекции соответственно.

Самый простой способ снизить ошибку репроекции – увеличить количество снимков и/или калибровочных точек. Если на некоторых снимках ошибка получается слишком высокой, то их можно отбросить и использовать для калибровки более удачные снимки. Однако этого может быть недостаточно для достижения приемлемо низкой ошибки. Для дальнейшего снижения ошибки репроекции оптимизировать положение камеры [16], применить методы оптимизации [17] или рассчитывать дисторсию в центре изображения и по его краям независимо [18]. Кроме того, существуют алгоритмы, снижающие ошибку за счёт оптимизации угла поляризации света [19].

Помимо традиционной шахматной доски, при калибровке камер могут использоваться другие шаблоны. Так, в OpenCV поддерживаются шаблоны в виде массива кругов, которые могут дать лучшую точность за счёт простоты поиска особых точек. Это связано с тем, что координаты центра круга можно найти просто как среднее от всех его пикселей, тогда как при поиске углов используются более сложные алгоритмы [20,21].

Существуют системы специализированных калибровочных шаблонов, в которых каждый элемент обладает уникальными особенностями геометрии. На рисунке 6 показано несколько маркеров из широко используемой библиотеки ArUco, которая также поддерживается OpenCV. Существует также шаблон ChArUco, совмещающий эти маркеры с шахматной доской.

Такие шаблоны более сложны для обработки, но могут быть использованы для более точной калибровки.

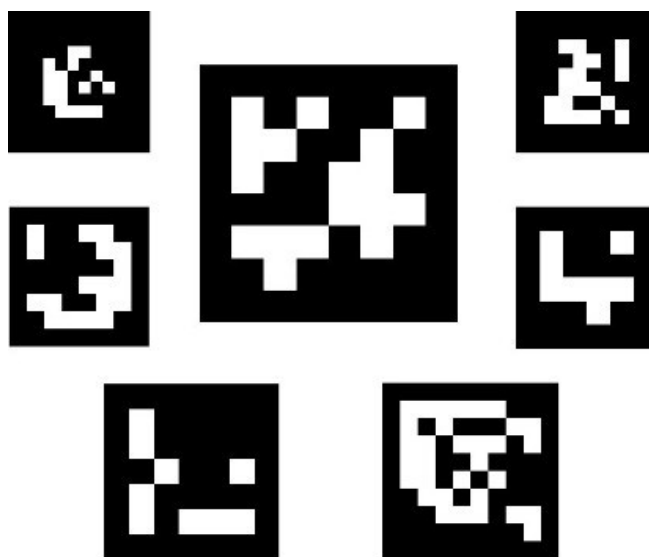


Рисунок 6 – Маркеры ArUco

2 ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ

2.1 Калибровка внутренних параметров

Вышеперечисленные алгоритмы различаются по своей точности и трудоёмкости, но все они в некоторой степени требуют участия человека, поскольку необходимо вручную перемещать калибровочный шаблон относительно камеры. Алгоритмы, не требующие перемещения шаблона (например, при калибровке камеры мобильного устройства по специальному маркеру), подразумевают перемещение самой камеры. При калибровке камеры на мобильном устройстве также могут использоваться данные МЭМС-датчиков [22].

Из этого следует, что для дальнейшего снижения роли человека в процессе калибровки необходимо автоматизировать перемещение калибровочного шаблона. При разработке алгоритма было решено использовать одну неподвижную камеру и экран, по которому с помощью случайных аффинных преобразований будет перемещаться изображение шаблона. Это позволит избежать проблем, связанных с использованием двух камер или одной движущейся, а также ручным перемещением шаблона. Таким образом, от пользователя потребуются только правильно расположить камеру и запустить программу.

Для корректной калибровки экран не должен быть изогнутым, чтобы не вносить на изображение дополнительных искажений. Кроме того, желательно расположить камеру на расстоянии, примерно равном фокусному, чтобы экран занимал как можно большую часть кадра. Это связано с тем, что для точной калибровки необходимо покрыть изображениями шаблона как можно большую часть поля зрения камеры.

При калибровке требуется не только перемещать шаблон параллельно плоскости изображения, но также масштабировать (то есть перемещать вдоль оптической оси камеры, тем самым меняя размер шаблона на проекции) и вращать, поскольку использование параллельных шаблонов в алгоритме

Zhang приводит к вырожденной конфигурации [9]. Исходя из этого, было предложено применить к исходному изображению следующие преобразования:

- а) смещение вдоль оси z , задающее масштаб и позволяющее повернуть изображение;
- б) поворот вокруг случайной оси;
- в) смещение по плоскости проекции.

Аффинные преобразования выполняются через умножение матрицы преобразования M на координаты изображения. Данная матрица вычисляется за несколько шагов [23]. Сначала вычисляется матрица для перехода в трёхмерное пространство A_1 :

$$A_1 = \begin{vmatrix} 1 & 0 & \frac{-w}{2} \\ 0 & 1 & \frac{-h}{2} \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{vmatrix}, \quad (7)$$

где w и h – ширина и высота изображения соответственно.

Далее, когда мы перешли к трёхмерному пространству, необходимо вычислить матрицу поворота R :

$$R_x = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(r_x) & -\sin(r_x) & 0 \\ 0 & \sin(r_x) & \cos(r_x) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad (8)$$

$$R_y = \begin{vmatrix} \cos(r_y) & 0 & -\sin(r_y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(r_y) & 0 & \cos(r_x) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad (9)$$

$$R_z = \begin{vmatrix} \cos(r_z) & -\sin(r_z) & 0 & 0 \\ \sin(r_z) & \cos(r_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad (10)$$

$$R = (R_x \cdot R_y) \cdot R_z, \quad (11)$$

где r_x , r_y и r_z – случайно выбранные углы поворота вокруг каждой из осей. Максимальные значения r_x и r_y , при которых возможно определение углов, необходимо определить экспериментальным путём, тогда как r_z может принимать любые значения.

Затем вычисляется матрица смещения T для удаления изображения шаблона от плоскости экрана на некоторое расстояние:

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad (12)$$

где t_z – расстояние между плоскостью экрана и изображением шаблона, выраженное в пикселях.

После этих преобразований вычисляется матрица A_2 для перехода обратно в двухмерное пространство:

$$A_2 = \begin{vmatrix} f & 0 & \frac{w}{2} & 0 \\ 0 & f & \frac{h}{2} & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix}, \quad (13)$$

где $f = \sqrt{w^2 + h^2}$ – фокусное расстояние, равное диагонали экрана.

Затем необходимо применить к изображению случайное смещение по осям x и y . Было решено делать это после перехода в двухмерное пространство, поскольку таким образом проще найти максимально возможные значения смещения, при которых изображение не выходит за

пределы экрана. Математически данное условие может быть описано следующей формулой:

$$\forall C_i(x_i, y_i) \in C: \begin{cases} 0 \leq x_i + t_x \leq w \\ 0 \leq y_i + t_y \leq h \end{cases}, \quad (14)$$

где C – множество углов изображения после умножения на матрицы (7), (11), (12) и (13), а t_x и t_y – смещения по соответствующим осям.

Исходя из условия (14), мы можем определить интервал возможных значений смещения по осям x и y :

$$t_x \in [-\min(x_1 \dots x_4); w - \max(x_1 \dots x_4)], \quad (15)$$

$$t_y \in [-\min(y_1 \dots y_4); h - \max(y_1 \dots y_4)], \quad (16)$$

Значения смещения генерируются дважды и берётся большее по модулю, чтобы изображения с большей вероятностью сдвигались к краям экрана. Это сделано из тех соображений, что на краях экрана наиболее сильно проявляется дисторсия, и по этой причине необходимо покрыть их калибровочными точками. По полученным значениям строится матрица случайного смещения T_{random} :

$$T_{random} = \begin{vmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{vmatrix}. \quad (17)$$

Наконец, мы можем перемножить матрицы (7), (11), (12), (13) и (17) для получения финальной матрицы преобразования M :

$$M = T_{random} \cdot A_2 \cdot T \cdot R \cdot A_1. \quad (18)$$

Теперь мы можем симулировать вращение и сдвиг калибровочного шаблона, умножив его изображение на матрицу (18). Преобразованное таким способом изображение необходимо вывести на экран и снять на камеру. Затем процедуру необходимо повторить ещё несколько раз, чтобы получить достаточное количество кадров для калибровки внутренних параметров камеры методом Zhang. Блок-схема полного алгоритма генерации калибровочного изображения показана на рисунке 7.

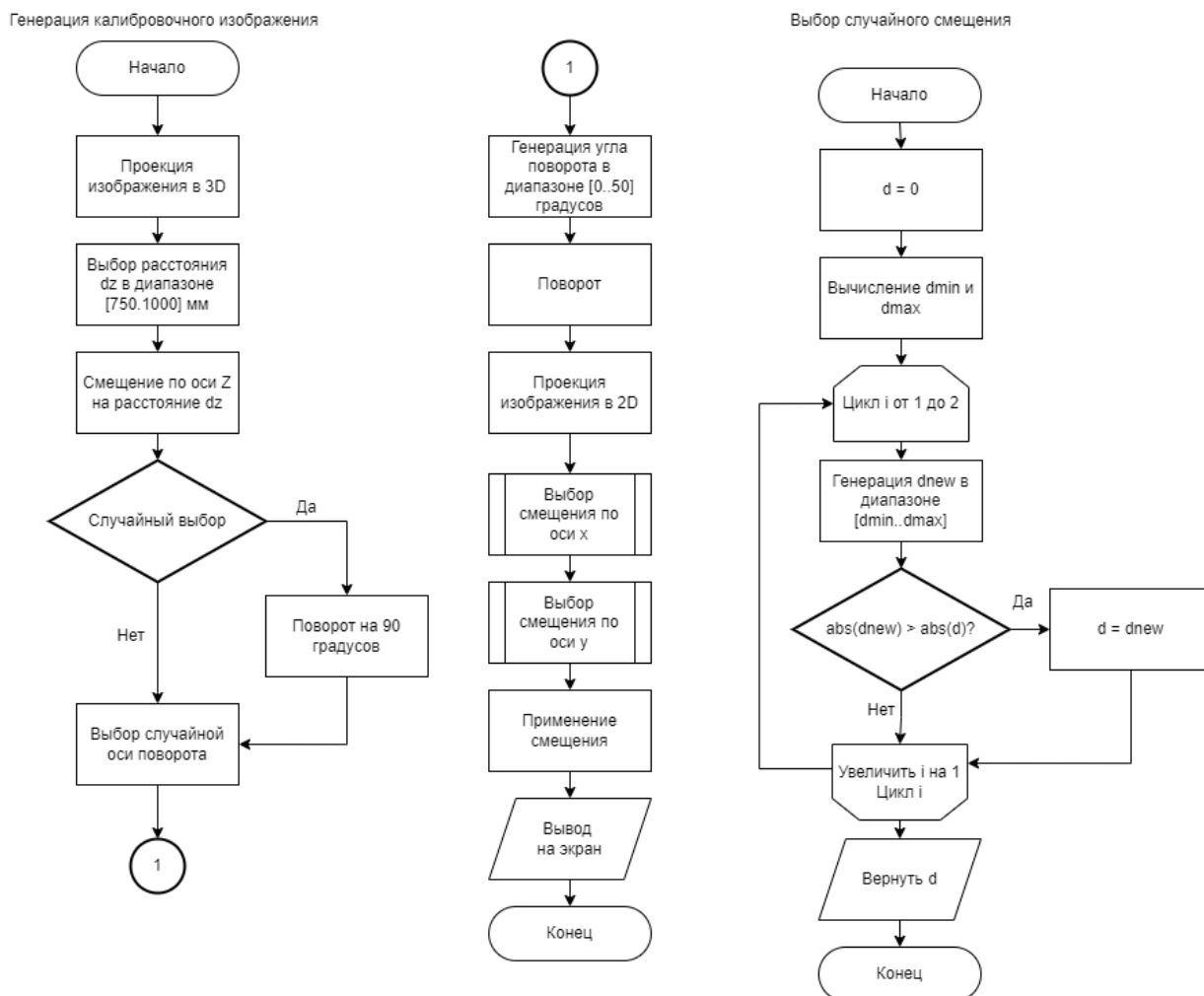


Рисунок 7 – Блок-схема алгоритма генерации калибровочного изображения

2.2 Калибровка nodal offset с помощью виртуальной камеры

В зависимости от выполняемой задачи может понадобиться определить также внутренние параметры камеры. Алгоритм Zhang позволяет вычислить поворот и смещение камеры относительно шаблона калибровки для каждого кадра. Однако при использовании изображения на экране эти значения можно легко получить и без выполнения калибровки, зная параметры генератора изображений и расстояние между камерой и экраном.

В сфере virtual production необходимо также определять позицию входного зрачка физической камеры, чтобы скорректировать перспективу виртуальной камеры. Для этого существует ещё один важный параметр –

nodal offset, то есть смещение входного зрачка камеры (nodal point) относительно её положения в игровом движке (рис. 8). Положение виртуальной камеры часто определяют с помощью прикреплённого к физической камере трекера, который транслирует свою позицию в игровой движок.

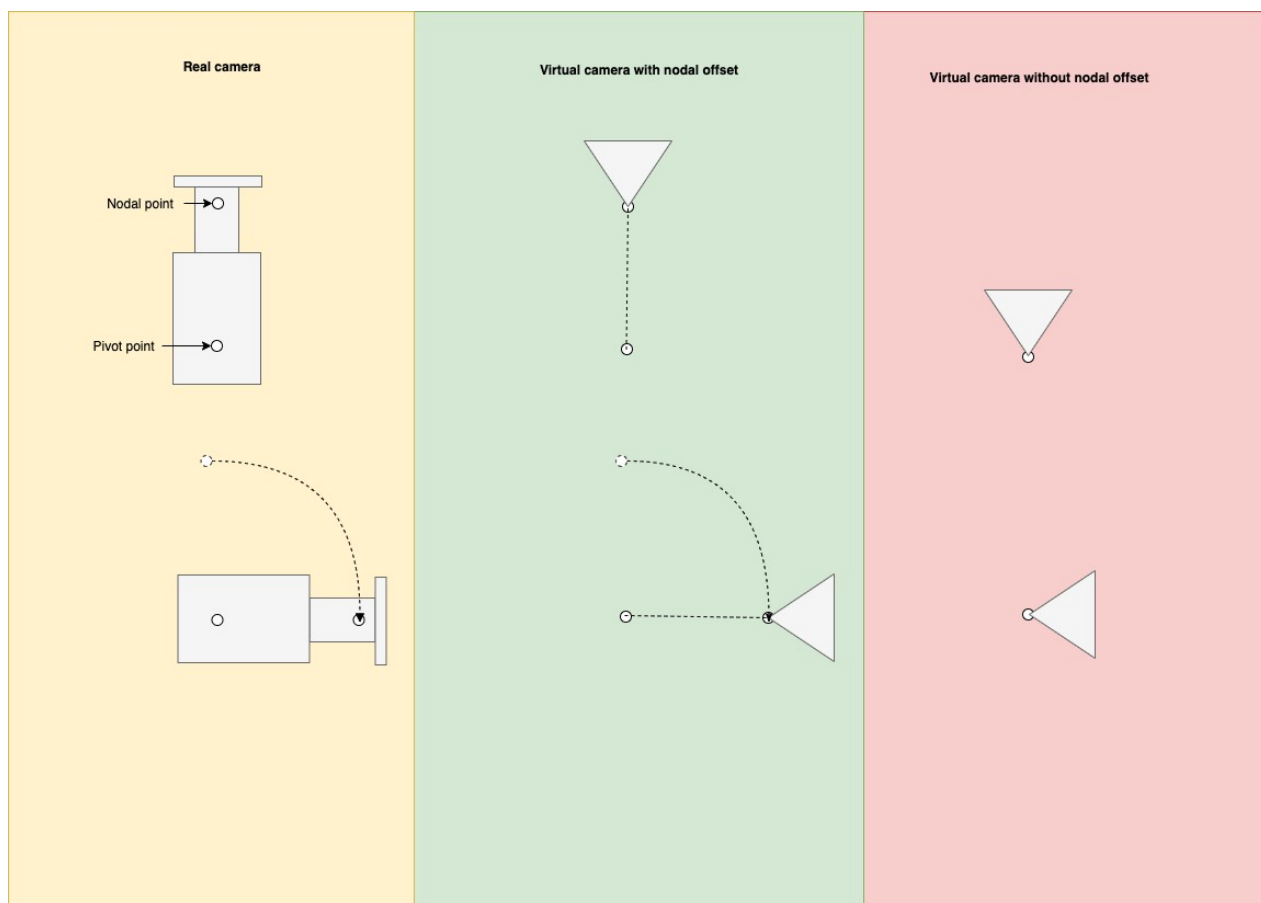


Рисунок 8 – Демонстрация nodal offset

Калибровка nodal offset может выполняться с помощью дополнительных трекеров или по снимкам калибровочного шаблона с разных ракурсов [24]. Так или иначе, для неё необходимо использовать методы стереозрения. Сопоставляя снимки физической и виртуальной камер, можно вычислить расстояние между ними и определить преобразования, которые позволят синхронизировать их перспективу. Полученные таким образом преобразования – это и есть искомый nodal offset.

В стереозрении используется *эпиполярная геометрия* (рис. 9), позволяющая определять *стереопары*, т.е. проекции одной и той же точки на два изображения. Две точки являются x и \hat{x} являются стереопарой, если существует такая матрица F , что

$$\hat{x}^T \cdot F \cdot x = 0. \quad (19)$$

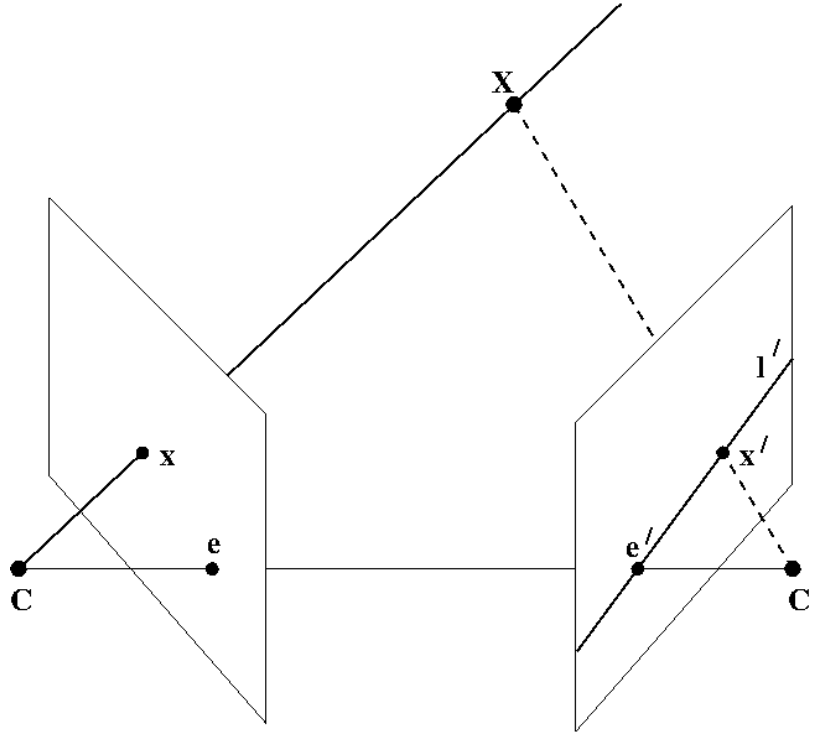


Рисунок 9 – Эпиполярная геометрия

Матрица, удовлетворяющая равенству (19), называется *фундаментальной матрицей* [25,26]. Для вычисления фундаментальной матрицы не требуется предварительная калибровка внутренних параметров камеры. Если же матрицы обеих камер известны, то мы можем вычислить *существенную матрицу* (essential matrix): [26]

$$E = K'^T \cdot F \cdot K, \quad (20)$$

где K и K' – матрицы первой и второй камеры соответственно. Поскольку мы используем виртуальную камеру с такой же матрицей, как у физической, то в нашем случае $K = K'$. Следовательно, уравнение (20) принимает вид

$$E = K^T \cdot F \cdot K. \quad (21)$$

Матрицу E можно разложить на вектор смещения t и матрицу поворота R следующим образом:

$$E = [t]_x \cdot R, \quad (22)$$

где $[t]_x$ обозначает кососимметричную матрицу, соответствующую вектору из трёх элементов t , которая определяется так:

$$[t]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \quad (23)$$

Наконец, мы можем объединить формулы (21) и (22), чтобы выразить смещение и поворот через фундаментальную матрицу и матрицу камеры:

$$[t]_x \cdot R = K^T \cdot F \cdot K. \quad (24)$$

Для нахождения фундаментальной матрицы F необходимо найти на физическом и виртуальном снимках стереопары калибровочных точек и решить систему уравнений. После калибровки внутренних параметров камеры и вычисления фундаментальной матрицы мы можем найти вектор смещения и матрицу поворота, совокупность которых является nodal offset.

2.3 Критерии оценки эффективности алгоритма

Поскольку для калибровки nodal offset необходимо сначала определить матрицу камеры, погрешность при калибровке внутренних параметров может оказать существенное влияние на результат. Таким образом, необходимо обеспечить максимально точную калибровку внутренних параметров, чтобы можно было правильно оценить точность второй части алгоритма.

Как было сказано выше, для этого в первую очередь требуется правильно расположить камеру. Затем необходимо определить количество снимков, при которых ошибка снижается до минимума. Для вычисления ошибки репроекции используется формула (6). Откалибровав внутренние параметры в оптимальных условиях, мы сможем дать оценку эффективности калибровки nodal offset.

В отличие от калибровки внутренних параметров, на втором этапе координаты калибровочных точек должны быть выражены в пикселях. Это необходимо для корректного вычисления смещения. Виртуальная камера должна использовать полученную на первом этапе алгоритма матрицу камеры и то же исходное изображение. После этого мы сможем определить преобразования, позволяющие перейти из системы координат виртуальной камеры в систему координат физической.

Для оценки точности можно также использовать оценку репроекции. После вычисления nodal offset и корректировки перспективы изображения калибровочных шаблонов на двух снимках должны совпасть. Построив график зависимости ошибки от количества снимков, мы сможем установить, скольких снимков достаточно для калибровки.

3 РАЗРАБОТКА РЕАЛИЗАЦИИ АЛГОРИТМА

3.1 Описание архитектуры прототипа

В реализации алгоритма было решено использовать библиотеку OpenCV, которая включает в себя инструменты для калибровки, генерации калибровочных шаблонов и вывода изображений на экран. Данная библиотека написана на C++, но существуют привязки к Python, Java, JavaScript и другим языкам. Для разработки прототипа был выбран Python исходя из соображений, что он является динамическим языком, хорошо подходящим для прототипирования, и часто используется для работы с OpenCV. Кроме того, редактор Unreal Engine поддерживает использование скриптов на Python, что позволит в будущем модифицировать прототип для интеграции с этим движком.

Для решения была разработана архитектура (рис. 10), включающая в себя три модуля:

- а) генератор изображений,
- б) драйвер камеры,
- в) калибровщик.

Первый модуль, помеченный на диаграмме как Generator – это генератор изображений, отвечающий за выполнение аффинных преобразований над исходным изображением шахматной доски и вывод результирующих изображений на экран с использованием формулы (18). Перемещение и вращение изображения позволяют симулировать ручное передвижение калибровочного шаблона. После отображения очередного снимка генератор подаёт сигнал драйверу камеры.

Второй модуль, помеченный как CameraDriver – драйвер камеры. Этот модуль отвечает за управление камерой и делает снимки изображений, выведенных предыдущим модулем. Полученные снимки затем передаются калибровщику.

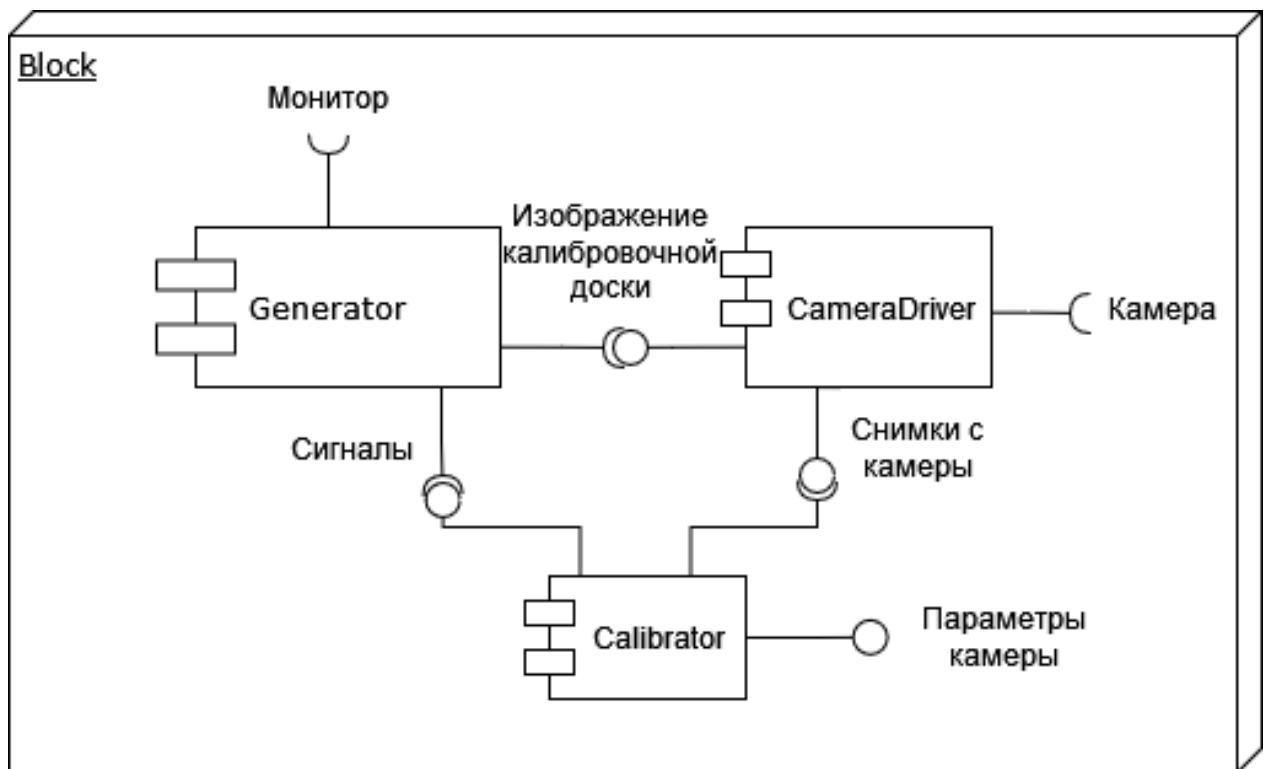


Рисунок 10 – Диаграмма архитектуры предложенного решения

Наконец, последний модуль, на диаграмме помеченный как Calibrator – это калибровщик, который отвечает за обработку полученных снимков и определение параметров камеры. На первом этапе алгоритма он осуществляет калибровку линзы, а после этого создаёт виртуальную камеру и с помощью стереокалибровки определяет nodal offset. Все полученные данные выводятся в файлы, а после калибровки внешних параметров также выполняется коррекция перспективы для снимков виртуальной камеры и оценка ошибки репроекции.

3.2 Разработка прототипа

В реализации генератора изображений было решено использовать шаблон в виде шахматной доски прямоугольной формы (рис. 11). Шахматная доска является самым распространённым и универсальным шаблоном для калибровки, поэтому для разработки прототипа было решено использовать именно её. Прямоугольная форма была выбрана исходя из соображений, что

по сравнению с квадратной она упрощает определение положения доски в пространстве за счёт разных длин сторон. Изображение было сгенерировано с помощью скрипта из библиотеки OpenCV.

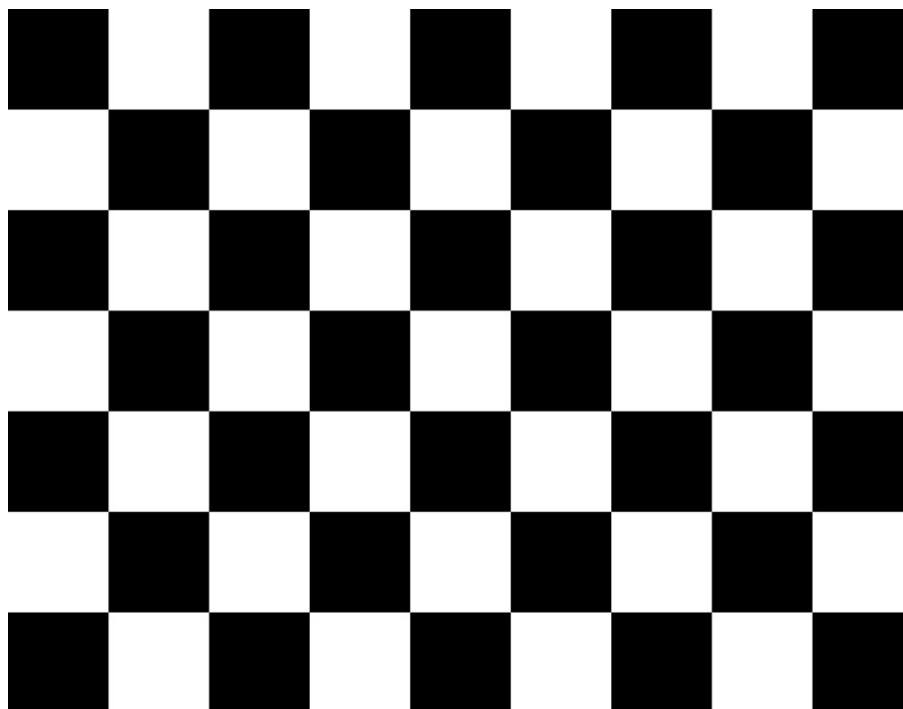


Рисунок 11 – Шаблон калибровки, использованный в тестовой реализации алгоритма

Изначально предполагалось, что изображение доски будет генерироваться «на лету», и таким образом, её размер можно будет настраивать. Однако во время разработки выяснилось, что скрипт создаёт векторные изображения в формате SVG, которые не могут быть напрямую отображены средствами OpenCV. Таким образом, чтобы отобразить доску, её необходимо сперва перевести в растровый формат. Чтобы не добавлять в проект дополнительную зависимость в виде библиотеки для работы с векторной графикой, было решено заранее сгенерировать векторное изображение, конвертировать в растровое и использовать его в программе. В

будущем этот подход может быть изменён, чтобы сделать алгоритм более гибким, позволяя пользователю настраивать размер доски и использовать другие виды шаблонов.

Скрипт генератора изображений при запуске создаёт окно, в котором размещает изображение доски, случайным образом смещённое и повёрнутое. При повороте шаблон ставится в горизонтальное либо вертикальное положение, после чего поворачивается вокруг оси, заданной случайным единичным вектором. Для определения максимального угла поворота был проведён эксперимент, в котором изображение поворачивалось одновременно по осям x и y на одинаковый угол α . Угол увеличивался от 0 до 90 градусов с шагом в 5 градусов. На каждом шаге делался снимок экрана. В ходе данного эксперимента было установлено, что при одновременном повороте доски по осям x и y более чем на 50 градусов OpenCV не может распознать её на снимке. Исходя из этого, за максимальное значение угла поворота по этим осям принято 50 градусов.

Алгоритм подбирает случайные значения смещения, удовлетворяющие условию (14). При этом значения генерируются дважды и используется большее по модулю, чтобы обеспечить лучшее покрытие обзора камеры контрольными точками. Итоговая матрица используется в качестве аргумента для функции `warpPerspective`. Выполнив преобразования над изображением, генератор изображений выводит его на экран. После вывода изображения генератор отправляет драйверу камеры команду сделать снимок (рис. 12). Полученный снимок затем обрабатывается калибровщиком.

Калибровщик реализован в виде скрипта, который сначала запускает два других модуля, а после окончания их работы обрабатывает полученные снимки. На снимках идёт поиск характерных точек доски, определяются их мировые координаты и координаты на плоскости изображения. Затем на основании полученных данных выполняется калибровка. Функция

calibrateCamera позволяет определить внутренние параметры камеры, а также коэффициенты дисторсии.

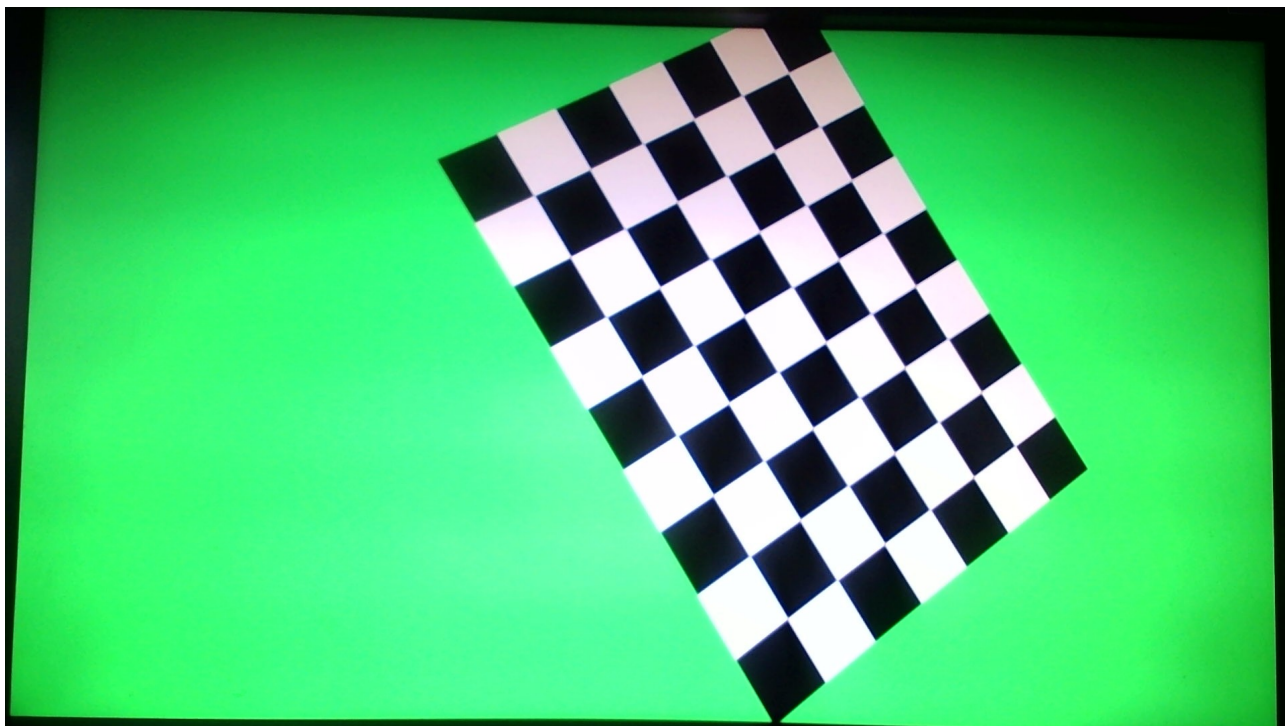


Рисунок 12 – Пример снимка, сделанного драйвером камеры

После калибровки линзы выполняется калибровка nodal offset. Для этого модуль калибровки создаёт виртуальную камеру, положение которой задаётся с помощью трекера. С помощью второго трекера, расположенного на мониторе, определяется позиция центра монитора, которая используется для определения расстояния до плоскости экрана. Виртуальная камера использует это расстояние и внутренние параметры камеры, полученные на предыдущем этапе, для симуляции съёмки изображения.

На рисунке 13 показана схема геометрии калибровки nodal offset. Точки Т и Т' обозначают трекеры, С – центр экрана (вычисленный с помощью трекера Т'), К – позиция физической камеры.

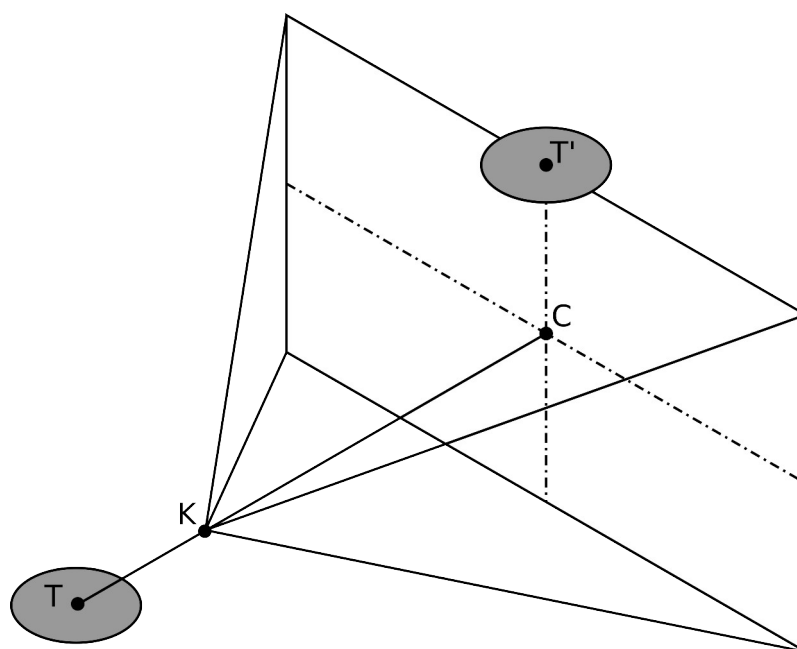


Рисунок 13 – геометрия виртуальной камеры

Оптическая ось виртуальной камеры проходит через центр экрана. Обе камеры снимают шахматную доску, расположенную одинаковым образом на расстоянии 750 мм от плоскости экрана. Таким образом делается несколько пар снимков, по которым программа выполняет калибровку стереосистемы из физической и виртуальной камер. Для этого используется функция `stereoCalibrate`, которая позволяет вычислить матрицы поворота и смещения для перехода между системами координат двух камер по формуле (24). Применяв эти преобразования к виртуальной камере, мы можем привести её в положение, аналогичное положению физической. Таким образом, результат последнего этапа работы алгоритма – это и есть искомый `nodal offset`, выраженный в виде матриц преобразований.

На рисунке 14 показана диаграмма классов разработанного прототипа. Для запуска программы необходима установка пакетов `opencv-python` и `screeninfo`. Библиотека `screeninfo` предоставляет информацию о мониторах. Объект `Monitor`, массив которых возвращается функцией `screeninfo.get_monitors()`, используется в конструкторах некоторых классов.

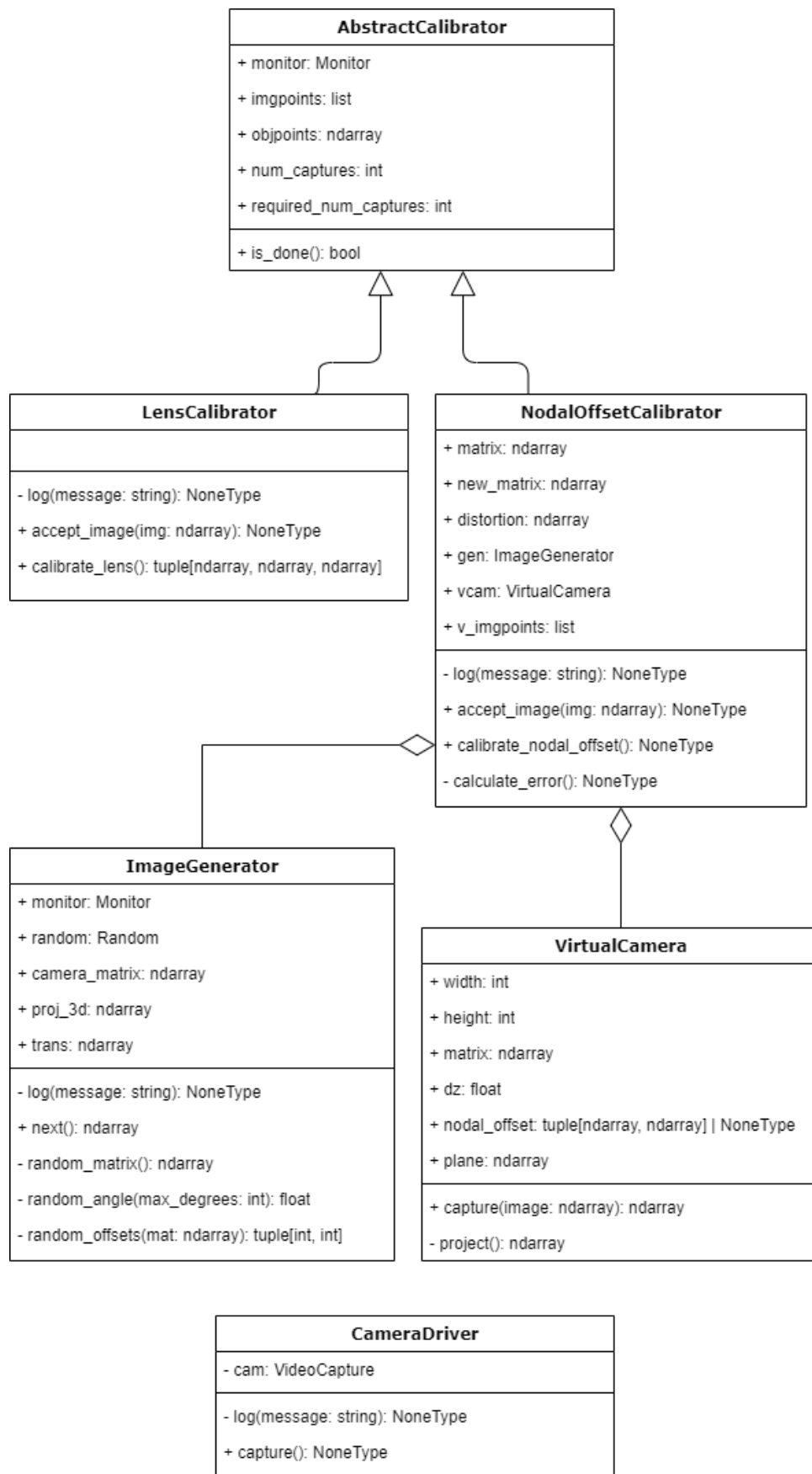


Рисунок 14 – Диаграмма классов прототипной реализации алгоритма

4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМА

4.1 Описание проведённых экспериментов

Для испытания разработанного алгоритма были проведены эксперименты с использованием веб-камеры модели DEXP M100R1. Технические характеристики камеры даны в таблице 1.

Таблица 1 – характеристики веб-камеры DEXP M100R1

Характеристика	Значение
Тип матрицы	CMOS
Число мегапикселей	2 Мп
Разрешение	1920×1080
Максимальная частота кадров	30/сек
Фокусировка	Ручная

В эксперименте по измерению внутренних параметров использовался экран ноутбука. Камера и экран располагалась на расстоянии 215 мм друг от друга таким образом, чтобы оптическая ось камеры приблизительно проходила через центр экрана. Центрирование оптической оси выполнялось с помощью стандартного приложения «Камера» в Windows 11 со включённым перекрестием. Конфигурация эксперимента показана на рисунке 15.

При калибровке nodal offset использовался комплект HTC Vive, предварительно откалиброванный с помощью SteamVR. Поскольку в наличии имелся только один трекер, сначала с его помощью была найдена позиция точки T' на мониторе (см. рисунок 12), а затем он был размещён на камере. Позиция трекера в обеих точках была выведена в консоль с помощью библиотеки ruorenxr и записана в файл.

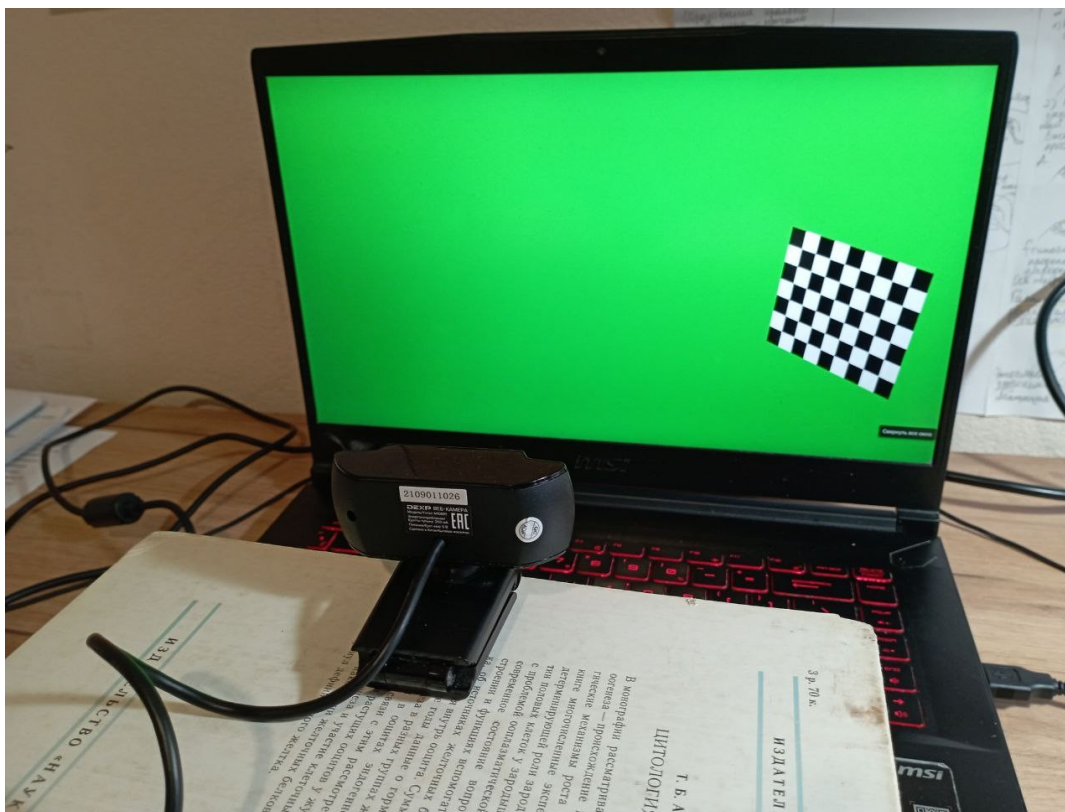


Рисунок 15 – Конфигурация эксперимента

4.2 Результаты экспериментов

В качестве метрики для оценки точности работы алгоритма использовалась ошибка репроекции, возвращаемая функциями `calibrateCamera` и `stereoCalibrate`. OpenCV вычисляет ошибку репроекции по формуле (6).

Как было сказано выше, расстояние между камерой и экраном было фиксированным во всех экспериментах. Таким образом, единственным переменным параметром было количество снимков. На рисунке 16 показан график зависимости ошибки репроекции от количества снимков при калибровке внутренних параметров камеры, а на рисунке 17 – аналогичный график для калибровки `nodal offset`.

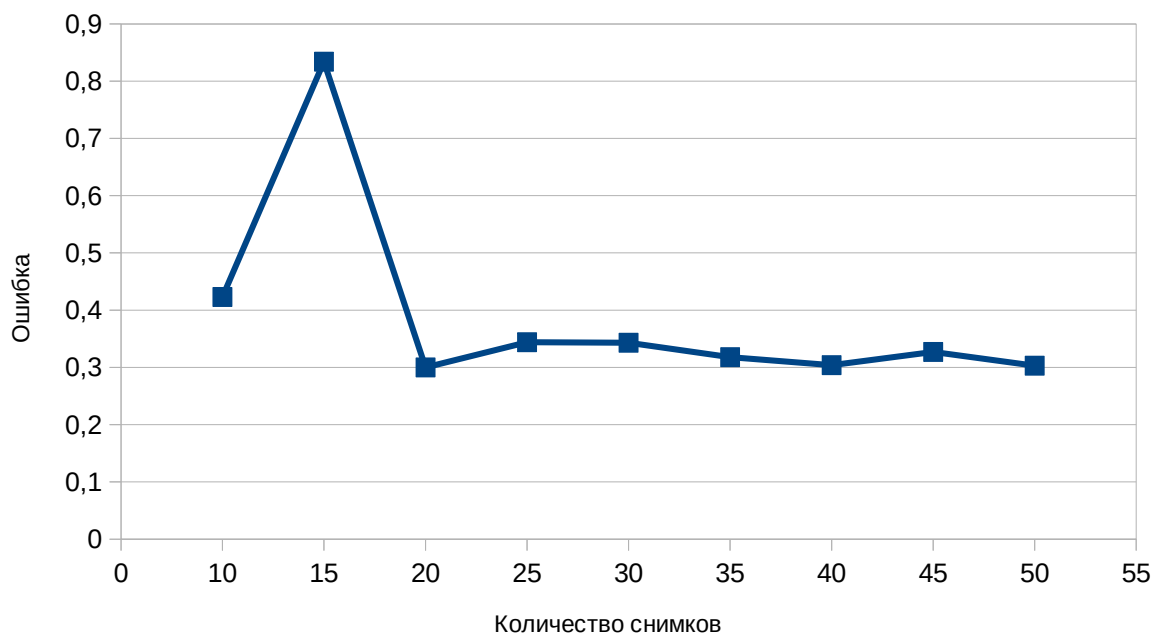


Рисунок 16 – График зависимости RMSE от количества снимков при калибровке внутренних параметров камеры

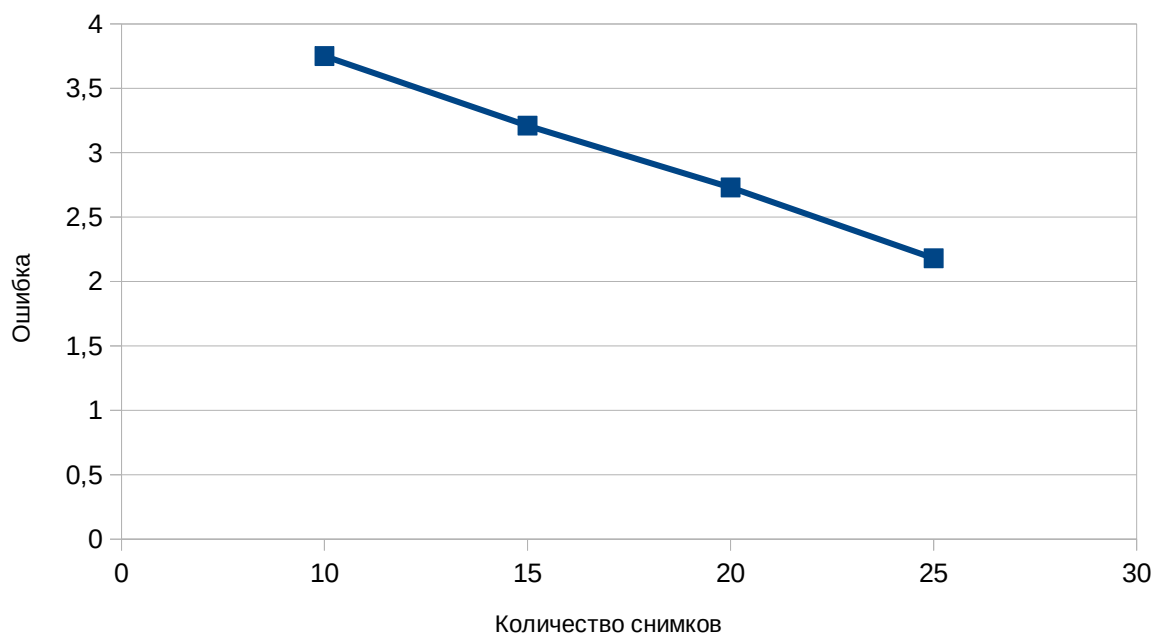


Рисунок 17 – График зависимости RMSE от количества снимков при калибровке nodal offset

4.3 Анализ результатов

В ходе экспериментов было установлено, что разработанный алгоритм успешно справляется с задачей автоматической калибровки камеры. Пример снимков, сделанных камерой во время эксперимента, показан на рисунке 18.

Эксперимент с калибровкой внутренних параметров показал, что максимальная точность алгоритма достигается уже при 20 снимках. Выброс при 15 снимках связан со случайной природой алгоритма и говорит о том, что в этом случае калибровочная доска покрыла недостаточно большую область экрана. Для исправления подобных проблем алгоритм можно модифицировать таким образом, чтобы он гарантировал более равномерное покрытие экрана перемещающимся шаблоном.

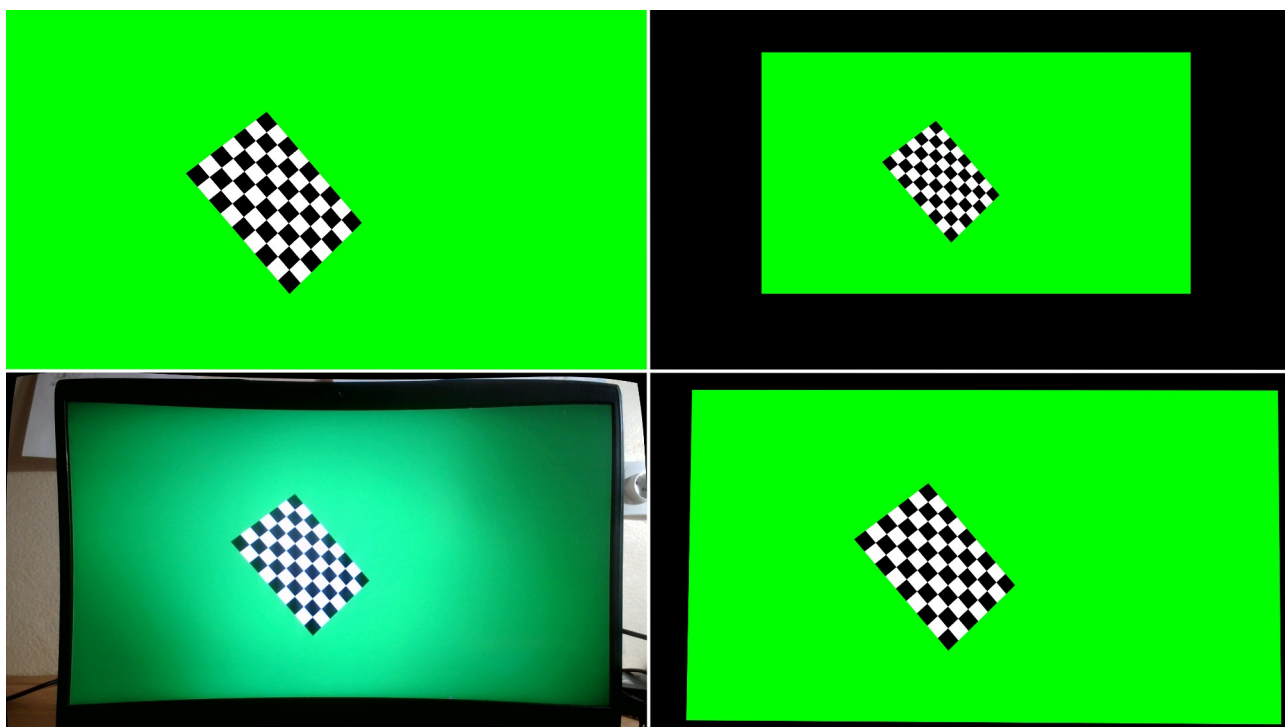


Рисунок 18 – Слева направо, сверху вниз: исходное изображение, его снимок с виртуальной камеры, снимок веб-камеры и снимок виртуальной камеры, скорректированный вычисленными внешними параметрами

В ходе второго эксперимента изменение количества снимков слабо влияло на ошибку, которая оставалась высокой на протяжении всего

эксперимента. Это говорит о том, что ошибка вызвана прежде всего погрешностями, накопившимися на протяжении предыдущего эксперимента, так как стереокалибровка полагается на корректную калибровку внутренних параметров камеры. Погрешности могут быть связаны с проведением эксперимента с помощью веб-камеры. Использование профессиональной видеокамеры позволило бы точнее управлять её положением в пространстве и избежать погрешности, связанной со случайным движением камеры во время калибровки.

4.4 Потенциальные улучшения

Самая заметная проблема в проведённых экспериментах – погрешность в измерении nodal offset. По снимкам заметно, что после калибровки изображения шахматной доски на снимках находятся близко друг к другу, они не совпадают идеально. Для снижения погрешности возможно модифицировать алгоритм таким образом, чтобы процесс калибровки продолжался до достижения приемлемой точности, а некачественные снимки автоматически отбрасывались. Подобные алгоритмы успешно применяются при калибровке камеры в полуавтоматическом режиме [27].

На точность измерений негативно повлияло также использование в экспериментах веб-камеры, которую крайне проблематично расположить правильным образом. Веб-камера также не поддерживает разные уровни зума. Для поддержки камер с регулируемым зумом будет необходимо выполнить калибровку при разных масштабах и с помощью линейной интерполяции установить зависимость параметров камеры от зума. Необходимо проведение экспериментов с использованием профессионального оборудования, чтобы улучшить точность калибровки.

Наконец, реализованный алгоритм калибровки nodal offset наиболее актуален для технологии virtual production. Исходя из этого, могут быть реализованы версии алгоритма для популярных игровых движков,

использующихся в этой технологии, таких как Unreal Engine и Unity. К примеру, для работы с Unreal Engine прототип можно модифицировать таким образом, чтобы он мог запускаться из редактора, а результаты его работы записывались в файл линзы с использованием API движка. Альтернативно, алгоритм может быть реализован на C++ и разделён на два модуля – калибровщик дисторсии и калибровщик nodal offset. Таким образом, реализованные калибровщики можно будет выбирать в настройках файла линзы и удобно сравнивать с ручными алгоритмами калибровки.

ЗАКЛЮЧЕНИЕ

В рамках выполнения данной ВКР были рассмотрены существующие подходы к калибровке камеры и предложен алгоритм, позволяющий выполнять калибровку внутренних параметров и nodal offset в автоматическом режиме, используя только неподвижную камеру и монитор.

Была разработана тестовая реализация предложенного алгоритма с использованием библиотеки OpenCV. Эффективность алгоритма была проверена экспериментально с использованием веб-камеры модели DEXR M100R1. Эксперименты показали, что алгоритм выполняет поставленную задачу. Наконец, были предложены потенциальные улучшения для реализации алгоритма: поддержка различных шаблонов, автоматический отбор полученных кадров и динамический процесс калибровки, продолжающийся до снижения ошибки до требуемого уровня.

Предложенный алгоритм может быть использован для решения задач virtual production на Unreal Engine, Unity или любом другом движке. Вместо реализованного класса виртуальной камеры в таком случае может быть использована виртуальная камера из игрового движка.

Результат выполнения ВКР и его исходный код, примеры и инструкции к запуску опубликованы в репозитории на GitHub по следующей ссылке: <https://github.com/Azmalent/auto-calibration/>

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. opencv/opencv: Open Source Computer Vision Library [Электронный ресурс]. URL: <https://github.com/opencv/opencv> (дата обращения: 16.01.2022).
2. Шуляев А.В. Модель камеры // Международный научно-исследовательский журнал. Индивидуальный предприниматель Соколова Марина Владимировна, 2014. № 1-1 (20).
3. Heikkila J., Silven O. Four-step camera calibration procedure with implicit image correction // Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. IEEE, 1997. С. 1106–1112.
4. Конушин А. Геометрические свойства нескольких изображений [Электронный ресурс] // Компьютерная графика и мультимедиа (сетевой журнал). 2006. № 4. URL: <http://masters.donntu.org/2012/iii/chigarev/library/article2.htm> (дата обращения: 17.01.2022).
5. Brown D. Decentering Distortion of Lenses // Photom. Eng. 1966. Т. 32, № 3. С. 444–462.
6. Conrady A.E. Decentred Lens-Systems // Mon. Not. R. Astron. Soc. Oxford Academic, 1919. Т. 79, № 5. С. 384–390.
7. Глаголев В.М. Описание и программное устранение дисторсии объективов // Известия Тульского государственного университета. Технические науки. Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тульский государственный университет», 2017. № 9–2.
8. Tsai R. Versatile Camera Calibration Techniaue for High-Accuracy 3D Machine Vision Metrology // Robot. Autom. 1987.
9. Zhang Z. A flexible new technique for camera calibration // IEEE Trans. Pattern Anal. Mach. Intell. 2000. Т. 22, № 11. С. 1330–1334.
10. Tapper M., McKerrow P., Abrantes J. Problems encountered in the implementation of Tsai's algorithm for camera calibration // Proc. 2002 Australas. Conf. 2002. № November. С. 27–29.

11. Burger W. Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation // Technical Report HGB16-05. 2016. 56 с.
12. Zhang Z. Camera Calibration: a Personal Retrospective // Mach. Vis. Appl. Springer Verlag, 2016. Т. 27, № 7. С. 963–965.
13. OpenCV: Camera Calibration and 3D Reconstruction [Электронный ресурс]. URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html (дата обращения: 09.05.2024).
14. Li W. и др. A practical comparison between Zhang's and Tsai's calibration approaches // Proceedings of the 29th International Conference on Image and Vision Computing New Zealand. New York, NY, USA: ACM, 2014. Т. 19-21-Nove. С. 166–171.
15. The Reprojection Error? [Электронный ресурс]. URL: <https://www.camcalib.io/post/what-is-the-reprojection-error> (дата обращения: 09.05.2024).
16. Ricolfe-Viala C., Sanchez-Salmeron A.-J. Camera calibration under optimal conditions // Opt. Express, Vol. 19, Issue 11, pp. 10769-10775. Optica Publishing Group, 2011. Т. 19, № 11. С. 10769–10775.
17. Шешин Е.П. и др. Калибровка системы стереоскопического компьютерного зрения // Труды Московского физико-технического института. Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)», 2018. Т. 10, № 2 (38).
18. Huang J. и др. Camera calibration optimization algorithm that uses a step function // Opt. Express, Vol. 32, Issue 11, pp. 18453-18471. Optica Publishing Group, 2024. Т. 32, № 11. С. 18453–18471.
19. Zhu Z. и др. Camera calibration method based on optimal polarization angle // Opt. Lasers Eng. Elsevier, 2019. Т. 112. С. 128–135.
20. Heikkilä J. Geometric camera calibration using circular control points // IEEE Trans. Pattern Anal. Mach. Intell. IEEE, 2000. Т. 22, № 10. С. 1066–1077.
21. Kaehler A., Bradski G. Learning OpenCV 3 // O'Reilly Media, Inc. 2018.

22. Поляков А.Р., Корнилова А.В., Кириленко Я.А. Автоматическая калибровка и синхронизация камеры и МЭМС-датчиков // Труды Института системного программирования РАН. Institute for System Programming of the Russian Academy of Sciences, 2018. Т. 30, № 4. С. 169–182.
23. How to transform a 2D image into a 3D space? | Towards Data Science [Электронный ресурс]. URL: <https://towardsdatascience.com/how-to-transform-a-2d-image-into-a-3d-space-5fc2306e3d36> (дата обращения: 06.04.2023).
24. Camera Lens Calibration Overview | Epic Developer Community [Электронный ресурс]. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/camera-lens-calibration-overview?application_version=5.4 (дата обращения: 25.04.2024).
25. Введение в эпполярную геометрию и стереозрение — Записки преподавателя [Электронный ресурс]. URL: <https://waksoft.susu.ru/2021/01/26/vvedenie-v-epipolyarnuyu-geometriyu-i-stereozrenie/> (дата обращения: 11.05.2024).
26. Andrew A.M. Multiple View Geometry in Computer Vision // Kybernetes. Cambridge University Press, 2001. Т. 30, № 9–10. С. 1333–1341.
27. Sovrasov V. OpenCV Interactive Camera Calibration [Электронный ресурс]. URL: https://docs.opencv.org/4.x/d7/d21/tutorial_interactive_calibration.html (дата обращения: 05.05.2024).