1 ОБЗОР ЛИТЕРАТУРЫ

Прежде чем перейти к практической реализации необходимых алгоритмов для решения задачи фотограмметрии, необходимо ознакомится с теоретическими материалами в данной области. Прежде всего, необходимо изучить основные подходы к решению таких задач, ознакомится с основными понятиями компьютерного зрения, проанализировать уже существующие решения, выделить их недостатки и преимущества.

1.1 Обзор существующих решений

Приложений-аналогов, которые бы удовлетворяли поставленным задачам, в данный момент не существует. В открытом доступе можно найти лишь приложения и наборы библиотек фотограмметрии, которые взаимодействуют с файловой системой и которые не являются сервисами. Тем не менее, данные аналоги являются одними из лучших программ для построения 3D-модели объекта по изображениям.

1.1.1 Meshroom

Приложение Meshroom (см. рисунок 1.1) позволяет выполнять фотограмметрию по набору изображений, которые получаются через взаимодействие с файловой системой. Этапы, которые будут выполняться для построения модели, описаны графом, что находится в нижней части главного окна приложения. Основано на библиотеках AliceVision и Qt.

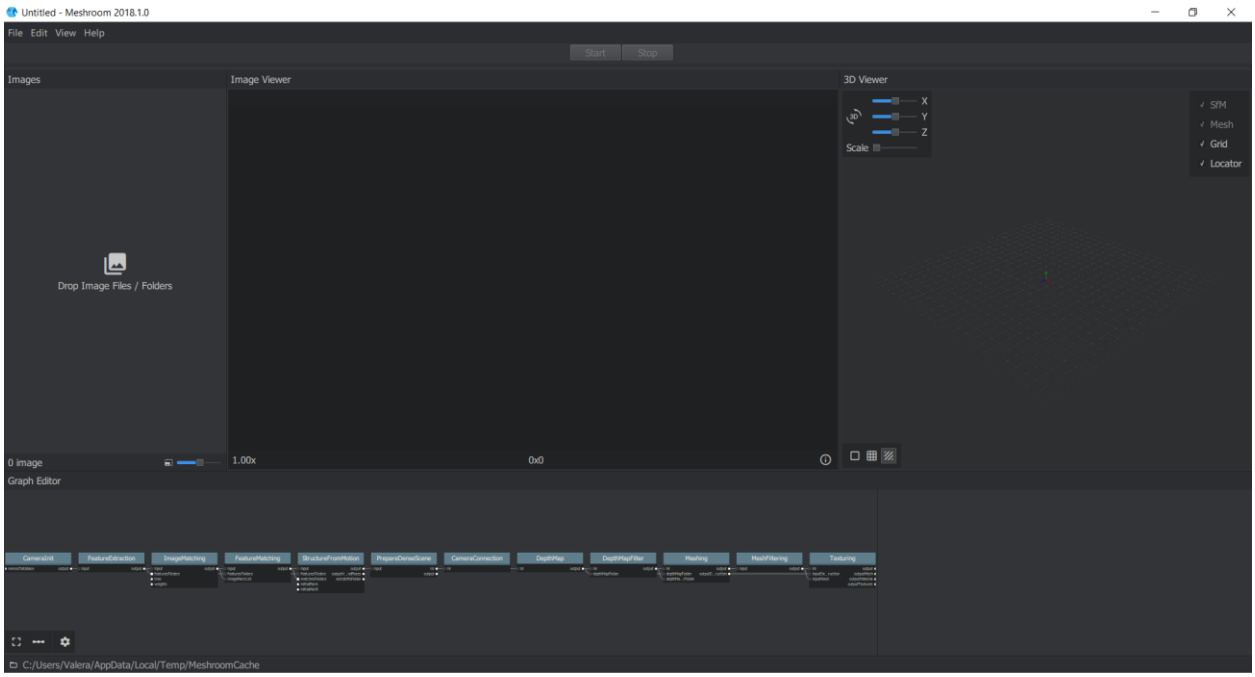

Рисунок 1.1 – Приложение «Meshroom»

1.1.2 Autodesk ReCap

Autodesk ReCap (см. рисунок 1.2) – приложение для фотограмметрии, которое считывает данные из файловой системы, загружает их на удалённый сервер, где производятся все этапы фотограмметрии, а затем высылает выходную модель обратно клиенту.

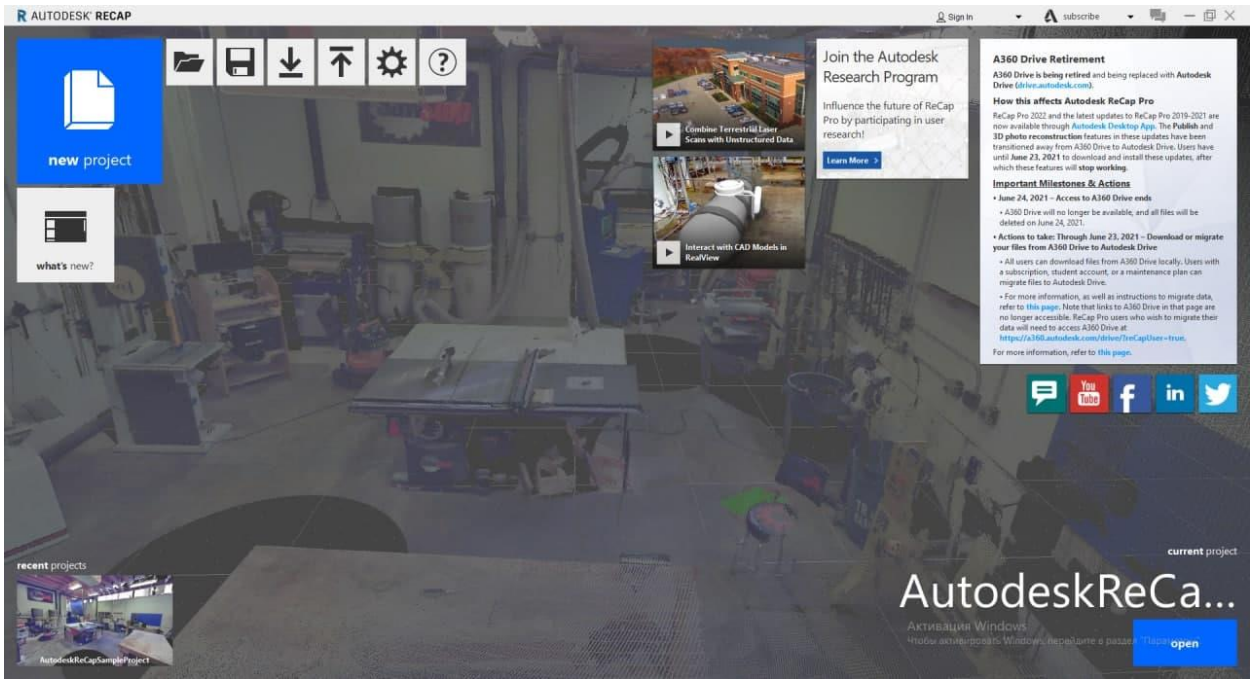

Рисунок 1.2 – Приложение «Autodesk ReCap»

ReCap хорош тем, что строит модель с отличной детализацией достаточно быстро. Так же данное приложение не требует большой вычислительной мощности от системы. Минусы данного приложения в том, что оно имеет закрытый исходный код, в нём нет возможности узнать, какие именно этапы построения модели выполнялись, и какие алгоритмы при этом использовались, нет возможности узнать промежуточные результаты.

1.1.3 OpenMVG & OpenMVS

OpenMVG(Open Multiple View Geometry) – библиотека фотограмметрии, для построения рассеянного облака точек. Данная библиотека выполняет такие этапы построение моделей, как извлечение ключевых точек изображения, сопоставление ключевых точек и нахождение пар изображений с максимальным числом совпадений, эпиполярная фильтрация точек, калибровка изображений, построение рассеянного облака точек.

Однако стоит отметить, что рассеянное облако точек не является конечной моделью объекта, поэтому результат работы данной библиотеки

необходимо дополнительно обрабатывать, чтобы получить конечную модель. Для этих целей в паре с OpenMVG часто используют библиотеку OpenMVS(Open Multiple View Stereovision). Данная библиотека выполняет такие этапы, как построение плотного облака точек, построение трёхмерной сетки объекта, полигональная минимизация, текстурирование.

Данные библиотеки так же работают с файловой системой для получения изображения. Точность моделей, построенных таким подходом, довольно высокая, но стоит отметить, что почти все этапы фотограмметрии в данных библиотеках выполняются на CPU, в результате чего они довольно медленные. Графического интерфейса данные библиотеки не имеют.

1.2 Анализ теоретической части

Перед тем, как приступать к реализации алгоритмов фотограмметрии, необходимо рассмотреть определённый теоретический материал.

1.2.1 Матрицы внутренней и внешней калибровки камеры

Для построения трёхмерной модели объекта с помощью фотограмметрии необходим набор изображений одного объекта, полученных съёмкой объекта с различных ракурсов. Точность построенной модели напрямую зависит от изображений, над которыми будет происходить обработка. Качество изображения во многом определяется камерой, которой эти изображения были получены.

Каждая камера как оптическая система имеет параметров, по которым можно сказать, насколько изображение, полученное данной камерой, точно передаёт пропорции объекта в реальном мире. Такими параметрами являются матрицы внутренней и внешней калибровки камеры.

Матрица внутренней калибровки – набор физических параметров камеры, характеризующих отдельно взятую камеру как оптическую систему. Матрица внутренней калибровки (1.1) содержит 5 параметров.

$$K = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1.1)$$

где f_x – фокусное расстояние, измеренное в ширине пикселя;

f_y – фокусное расстояние, измеренное в высоте пикселя;

γ – угловой коэффициент;

u_0 – координата x принципиальной точки;

v_0 – координата y принципиальной точки.

В формуле 1.1 фокусное расстояние измеряется в размерах пикселя. Однако в реальных камерах фокусное расстояние измеряется в миллиметрах. Определить фокусное расстояние в размерах пикселя через фокусное расстояние в миллиметрах можно по формулам 1.2 и 1.3:

$$f_x = \frac{f}{p_x}, \tag{1.2}$$

где f_x – фокусное расстояние, измеренное в ширине пикселя;
f – фокусное расстояние в миллиметрах;
p_x – ширина пикселя в миллиметрах.

$$f_y = \frac{f}{p_y}, \tag{1.3}$$

где f_y – фокусное расстояние, измеренное в высоте пикселя;
f – фокусное расстояние в миллиметрах;
p_y – высота пикселя в миллиметрах.

Угловой коэффициент γ в формуле 1.1 – коэффициент, равный:

$$\gamma = f_y * tg\ \varphi, \tag{1.4}$$

где γ – угловой коэффициент;
f_y – фокусное расстояние, измеренное в высоте пикселя;
φ – угол наклона пикселя.

Коэффициенты u_0 и v_0 – координаты принципиальной точки. Принципиальная точка – точка пересечения главной оптической оси системы с плоскостью изображения.

Стоит отметить, что все коэффициенты в матрице внутренней калибровки являются постоянными величинами для отдельно взятой камеры и определяются физическими параметрами камеры.

Матрица внешней калибровки – матрица, которая необходима для связи системы координаты камеры с системой координат реального мира. Данная матрица состоит из двух компонентов: матрицы поворота и вектора переноса.

$$E = [R \mid T], \tag{1.5}$$

где E – матрица внешней калибровки;
R – матрица поворота;

T – вектор переноса.

Размерность матрицы поворота – 3 x 3, размерность вектора поворота – 3 x 1.

Зная коэффициенты матриц внутренней и внешней калибровки, можно определить зависимость координат реального мира и координат изображения по следующей формуле:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \text{ x } Dist(\begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix} \text{ x } \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}), \quad (1.6)$$

где u – координата x точки на изображении;

v – координата y точки на изображении;

K – матрица внутренней калибровки камеры;

$Dist$ – функция оптической дисторсии;

R – матрица поворота;

T – вектор переноса;

x_w – координата x точки в реальном мире;

y_w – координата y точки в реальном мире;

z_w – координата z точки в реальном мире.

Стоит отметить, что матрица внешней калибровки заранее не известна. Таким образом, одной из подзадач фотограмметрии является задача определения коэффициентов матрицы внешней калибровки.

1.2.2 Ключевые точки изображения

Одним из этапов фотограмметрии является нахождение ключевых точек изображения. Ключевая точка – точка на изображении, которая обладает следующими свойствами:

1. Определённость – ключевая точка должна выделяться на фоне других точек.
2. Устойчивость – изменение яркости, контрастности и цветовой гаммы изображения не влияют на положение особой точки на объекте.
3. Инвариантность – ключевая точка обладает устойчивостью к повороту, изменению масштаба изображения и смены ракурса съёмки.
4. Стабильность – зашумлённость изображения, которая не превышает определённый порог, не должна влиять на положение особой точки на объекте.
5. Интерпретируемость – особые точки должны быть представлены в формате, пригодном для обработки.

6. Количество – число ключевых точек должно быть достаточным для обнаружения объекта.

Каждая ключевая точка обладает собственным дескриптором. Дескриптор ключевой точки – набор параметров, который определяет окрестность точки. Количество таких параметров и значение каждого из них определяются конкретными алгоритмами нахождения ключевых точек.

Дескриптор позволяет выделить ключевую точку на фоне остальных точек изображения. Так же дескрипторы ключевых точек используются для ещё одного этапа фотограмметрии – нахождение пар ключевых точек, принадлежащих одному и тому же объекту на разных изображениях.

Таким образом, появляются ещё две задачи фотограмметрии: нахождение ключевых точек на входных изображениях, вычисление их дескрипторов, и нахождение пар точек, которые соответствуют одинаковым объектам на различных изображениях.

1.2.3 Метод съёмки изображений

Как отмечалось ранее, точность модели, построенной с помощью фотограмметрии, напрямую зависит от качества входных данных. Чтобы достичь высокой точности модели, крайне желательно, чтобы все поверхности объекта хорошо просматривались и имели хорошее освещение. Освещение в данном случае играет очень важную роль, поэтому следует по возможности избегать рассеянного освещения и наличия резких теней.

Для построения высококачественной модели рекомендуется сначала снять объект по кругу, а затем фокусироваться на деталях, постепенно увеличивая масштаб.

Существует два способа равномерно снять объект по кругу.

Первый способ – зафиксировать положение объекта и передвигать камеру равномерно вокруг объекта. При таком методе рекомендуется передвигать камеру без резких движений, чтобы избегать нечётких кадров и размытости при движении. Желательно при съёмке сохранять постоянным угол поворота камеры. Пример съёмки изображён на рисунке 1.3.

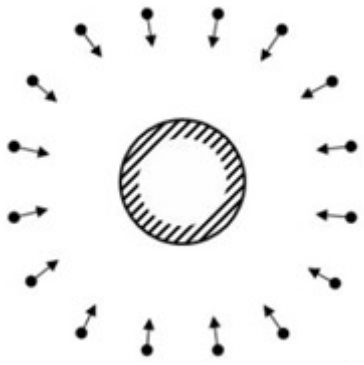

Рисунок 1.3 – Съёмка объекта при равномерном перемещении камеры

Второй способ – оставить камеру неподвижной, при этом поворачивая объект съёмки на постоянный угол. В некоторых случаях этот метод проще в реализации, чем первый, однако он не всегда возможен, например, в случае здания. При съёмке данным методом рекомендуется поместить объект на поворотной платформе, при этом обеспечив хорошее освещение и просматриваемость всех поверхностей объекта, и выполнять съёмку, поворачивая объект на определённый угол.

Для обоих случаев рекомендуется выдерживать угол поворота между соседними изображениями до 30°.

Данный этап является очень важным для построения модели. От качества выполнения этого этапа будет зависеть точность построенной модели и качество текстур.

1.3 Используемые технологии

Так как одной из поставленных задач является достижение высокой скорости работы, то необходимо выбрать язык программирования, код на котором будет исполняться с максимальной производительностью. Исходя из этого, для реализации данного программного модуля был выбран язык C++ (см. рисунок 1.4).

Рисунок 1.4 – Логотип языка C++

В качестве среды разработки выбор пал на IDE Clion от компании JetBrains (см. рисунок 1.5). Clion является одной из наиболее удобных IDE для C++. Стоит отметить, что в данной IDE имеются встроенные лексический и синтаксический анализатор, что повысит качество кода и упростит разработку.

Рисунок 1.5 – Логотип IDE Clion

Одной из наиболее популярных систем сборки проекта для C++ является CMake (см рисунок 1.6). CMake – кроссплатформенное средство автоматизированной сборки проекта, которое позволяет эффективно

распределять весь код проекта на отдельные модули, а так же эффективно управлять зависимостями проекта. Безусловным плюсом CMake в данном случая является то, что в подавляющем большинстве библиотек для C++ используется CMake, что упрощает добавление сторонних зависимостей в проект. Зачастую в таких случаях всё сводится к команде `find_package`, которая автоматически выполняет поиск всех необходимых компонентов той или иной зависимости и инициализирует внутренние переменные окружения, которые зачастую используются в команде `target_link_libraries`.

Рисунок 1.6 – Логотип CMake

Практика показывает, что на разных операционных системах код на C++ работает с разной скоростью. Это связано с внутренними особенностями операционных систем. В связи с этим, целевой операционной системой для данного проекта будет являться Ubuntu 20.04 LTS (см. рисунок 1.7), так как на ней C++ показывает хорошие скоростные показатели, в отличие от Windows.

Рисунок 1.7 – Логотип Ubuntu 20.04 LTS

Однако, чтобы не принуждать конечного пользователя менять операционную систему для запуска приложения, будет использоваться средство контейнеризации Docker (см. рисунок 1.8). Docker представляет собой нечто похожее на легковесную виртуальную машину, в которой можно настраивать необходимое окружение для определённого приложения с

помощью специального файла – Dockerfile. При развёртывании приложения таким образом, Docker скачивает на локальную ЭВМ образ нужной операционной системы, устанавливает все необходимые компоненты, указанные в конфигурационном файле, и предоставляет возможность запускать платформозависимое приложение на любой операционной системе.

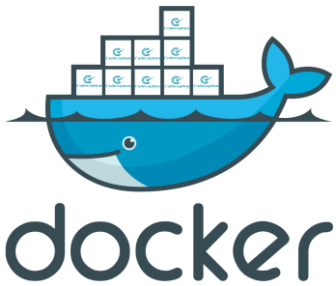

Рисунок 1.8 – Логотип Docker

Для реализации сетевого обмена данными будет использоваться Apache Kafka (см. рисунок 1.9). Kafka представляет собой средство обмена сообщениями между узлами, подключёнными к одному Kafka брокеру. Kafka брокер – сущность, которая отвечает за получение сообщений и гарантированную доставку сообщений адресату. При этом такой способ обмена данными имеет довольно высокую производительность.

Рисунок 1.9 – Логотип Apache Kafka

Так как задача фотограмметрии относится к классу задач компьютерного зрения, а для C++ одной из наиболее популярных библиотек для компьютерного зрения является OpenCV(см. рисунок 1.10). В этой библиотеке реализовано огромное множество алгоритмов и структур данных для всевозможной работы с изображениями. Кроме того, исходный код

библиотеки является открытым, что позволит узнать особенности реализации некоторых методов работы с изображениями.

Рисунок 1.10 – Логотип OpenCV

Несмотря на то, то в OpenCV реализовано много алгоритмов работы с изображениями с использованием GPU, существует библиотека, в которой не так много алгоритмов, как в OpenCV, зато скорость работы у неё в некоторых случаях превосходит OpenCV в 50 – 100 раз. Эта библиотека называется Nvidia NPP (Nvidia Performance Primitives). Данная библиотека является одной из множества библиотек CUDA-X, реализованных на языке C с использованием CUDA. Библиотека состоит из двух больших модулей: NPPI и NPPS. Данные модули отвечают за обработку изображений и обработку сигналов соответственно. Нас интересует модуль NPPI.

Зачастую в больших проектах возможностей стандартной библиотеки C++ оказывается не достаточно. Очень часто эту проблему решают, используя библиотеку Boost (см. рисунок 1.11). Boost – библиотека, состоящая из очень большого количества модулей. Данная библиотека хороша тем, что является прямым кандидатом на стандартизацию. Некоторые модули из Boost уже попали в современные стандарты языка C++. Данная библиотека постоянно развивается, имеет очень высокое качество исходного кода, высокую производительность и огромный набор решаемых задач.

Рисунок 1.11 – Логотип Boost