

Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего профессионального
образования

ДАЛЬНЕВОСТОЧНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Кафедра информатики

ОТЧЕТ

о прохождении преддипломной практики

Отчет защищен:
с оценкой

М.А.Гузов
«_____» марта 2010г.

Регистрационный № _____

«_____» марта 2010г.

Выполнил студент гр. 258

А.Е.Антонов

Руководитель практики
заведующий лабораторией машинной графики
ИАПУ ДВО РАН, д.т.н.

В.А.Бобков

Практика пройдена в срок

с «07» декабря 2009г.

по «13» марта 2010г.

на предприятии

ДВГУ, Кафедра информатики

г. Владивосток

2010

Оглавление

1	Введение	3
1.1	Глоссарий	3
1.2	Описание предметной области	3
1.2.1	Методы построения моделей объектов	3
1.3	Неформальная постановка задачи	4
1.4	Математические методы	4
1.5	Обзор существующих методов решения	4
2	Требования к окружению	4
2.1	Требования к аппаратному обеспечению	4
2.2	Требования к программному обеспечению	5
3	Спецификация данных	5
3.1	Формат метафайла	5
4	Функциональные требования	5
5	Требования к интерфейсу	5
6	Проект	6
6.1	Средства реализации	6
6.2	Модули и алгоритмы	6
6.2.1	Классы приложения	6
6.2.2	Класс MainWindow	6
6.2.3	Класс ImagePreview	6
6.2.4	Класс ImageScene	6
6.3	Проект интерфейса	7

1 Введение

1.1 Глоссарий

Воксел – элемент объёмного изображения, содержащий значение элемента растра в трёхмерном пространстве. Воксели являются аналогами пикселей для трёхмерного пространства. [13]

Кроссплатформенное программное обеспечение – программное обеспечение, работающее более чем на одной аппаратной платформе и/или операционной системе. [14]

ПК – персональный компьютер.

Реконструкция – построение объемной модели реального объекта.

Фреймворк – в информационных системах структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. В его состав могут входить вспомогательные программы, библиотеки кода, язык сценариев и проч. [16]

1.2 Описание предметной области

Потребность в реалистичном отображении окружающего мира увеличивает значимость трехмерного (3D) моделирования – построения компьютерных моделей объектов. 3D модели облегчают планирование, контроль и принятие решений во многих отраслях. Например, если в ходе эксплуатации модели требуется внести коррективы, компьютерный объект позволит это сделать максимально быстро.

Особенно сложной проблемой является создание точных моделей объектов из реального мира. [15] Как быстро выяснилось, человеческий глаз очень легко определяет погрешности в синтезированных изображениях хорошо известных ему в повседневной жизни объектов.

Вот несколько примеров объектов, требующих реконструкции:

1. Здания и строения
2. Предприятия со сложной структурой (нефтегазоперерабатывающие комплексы, химические предприятия и т.д.)
3. Дороги и дорожные объекты (мосты, путепроводы, прилегающая зона)
4. Открытые и закрытые горные разработки
5. Рельефы местности

1.2.1 Методы построения моделей объектов

На данный момент существует три основных способа получения модели объекта:

- Использование активных методов (например, лазерных сканеров, позволяющих с высокой точностью определить расстояние от лазера до поверхности объекта)
- Вручную с помощью программ трехмерного моделирования (например, Blender, 3D Studio Max)
- Автоматическая реконструкция объекта по набору изображений и другой известной о нем информации

Сканеры – современное средство измерения, позволяющее производить исполнительную съемку сложных объектов, там, где проходит множество пересечений на различных уровнях. Причем вступать в контакт с самим объектом не требуется. Сканер формирует модель по десяткам тысяч точек, при этом положение каждой из них определяется с точностью от 2–3 мм до 2 см. Наземные сканеры используются для создания моделей зданий, заводов, кварталов и других больших сооружений. Дальность сканера составляет порядка 200 метров, на таком расстоянии он может определять объекты с точностью шага до 0.5 см. Современные

сканеры в состоянии сканировать окружающую действительность с поворотом на 360 градусов, что позволяет сканировать сразу несколько зданий. Сшивка отдельных сканов выполняется посредством специальных марок или характерных точек рельефа. Такой метод обеспечивает высокое качество проектирования, приближая его к реальности. Недостатками использования сканера являются высокая стоимость оборудования (от 50 тыс. долл. [5]), необходимость в высококвалифицированных специалистах, способных это оборудование обслуживать, а так же невозможность реконструкции динамических сцен. [15]

Второй подход требует больших затрат усилий и времени, что вытекает в высокую стоимость и длительные сроки разработки модели.

В третьем подходе в качестве исходных данных используют только фотографии или последовательности изображений объектов, полученные в естественном освещении, или приближенном к нему (в свете обычных ламп). Такой подход считается наиболее перспективными, так как позволяет строить модели динамических сцен, в которых могут участвовать люди.

1.3 Неформальная постановка задачи

Создание кроссплатформенного приложения, использующего для реконструкции объекта алгоритм раскраски вокселей по методу множества гипотез [6] и способного задействовать все возможности современного ПК. Из поставленных условий вытекает еще одна задача – распараллеливание существующего алгоритма.

1.4 Математические методы

В качестве базового, для построения модели объекта, используется алгоритм раскраски вокселей по методу множества гипотез [6].

1.5 Обзор существующих методов решения

На текущий момент была найдена только одна, находящаяся в открытом доступе, реализация алгоритма раскраски вокселей – Voxel Coloring Framework [8]. К её достоинствам можно отнести распространение под академической свободной лицензией [1]. Однако данная реализация обладает следующими недостатками:

- Работает только в операционной системе семейства MS Windows
- Для работы необходим пакет Matlab
- Не использует вычислительные возможности современных графических адаптеров

2 Требования к окружению

2.1 Требования к аппаратному обеспечению

Необходимо наличие в системе устройств с поддержкой стандарта OpenCL [7]. Список поддерживаемых устройств можно найти

- для NVIDIA в [11]
- для ATI в [2]
- для S3 в [12]
- для IBM в [4]

2.2 Требования к программному обеспечению

Для работы приложения необходимо наличие в системе установленной библиотеки Qt версии не ниже 4.6, а так же драйверов для устройств с поддержкой стандарта OpenCL версии 1.0.

3 Спецификация данных

3.1 Формат метафайла

Метафайл – файл, используемый для сохранения и последующего восстановления информации о загруженных пользователем изображениях, введенных матрицах внешней и внутренней калибровки, объемлющем основной объект прямоугольнике. Этот файл записан в текстовом формате. Файл имеет следующую структуру:

- В первой строке находится беззнаковое целое число, обозначающее количество изображений, информация о которых записана в файле.
- Далее идет три строки, содержащие по три элемента, разделенных пробелами. Каждый элемент – число с плавающей точкой. Вместе они описывают матрицу внутренней калибровки камеры.
- Далее идет информация об изображениях:
 - имя файла с изображением, записанное в формате, совместимом с типом QString (см. [10])
 - прямоугольник, объемлющий основной объект, записанный в формате, совместимом с типом QRectF (см. [9])
 - матрица внешней калибровки камеры для данного изображения, записанная в три строки по три числа с плавающей точкой в каждой.

Количество таких блоков должно быть равным числу, записанному в первой строке файла.

4 Функциональные требования

Система должна позволять пользователю:

- загружать изображения
- выделять основной объект
- вводить матрицу внешней калибровки камеры для каждого изображения
- вводить матрицу внутренней калибровки камеры
- запускать выполнение алгоритма построения воксельной модели, на основе загруженных изображений
- просматривать полученную воксельную модель объекта
- сохранять и загружать метафайл
- сохранять построенную воксельную модель
- загружать для просмотра воксельную модель

5 Требования к интерфейсу

Приложение должно иметь графический интерфейс пользователя и из него должна быть доступна вся функциональность приложения.

6 Проект

6.1 Средства реализации

Для реализации приложения был выбран язык программирования C++, т.к. он очень распространен, что упрощает последующую поддержку приложения. К тому же, программы, написанные на этом языке, могут быть скомпилированы для очень большого количества платформ.

Для упрощения переноса приложения на другие платформы, был использован кроссплатформенный инструментарий разработки ПО под названием Qt. Он так же включает в себя инструментарий, необходимый для разработки кроссплатформенных приложений с графическим интерфейсом пользователя. Выбор этого инструментария позволяет уменьшить усилия, требуемые для реализации новой и поддержки имеющейся функциональности.

Для того, чтобы приложение могло задействовать как можно больше доступных ресурсов современных ПК, было решено использовать фреймворк OpenCL. Это позволит приложению задействовать ресурсы всех процессоров в многопроцессорных системах, а так же ресурсы современных видеокарт. Для взаимодействия с OpenCL используется библиотека OpenCLxx [3], которая предоставляет ООП-интерфейс для доступа к OpenCL.

6.2 Модули и алгоритмы

6.2.1 Классы приложения

- *MainWindow* – главное окно приложения
- *ImageInfo* – информация об изображении
- *ImagePreview* – виджет предпросмотра изображений
- *ImageScene* – сцена для отображения изображения

6.2.2 Класс MainWindow

Методы класса:

- *add_image* – добавление нового изображения
- *load_metafile* – загрузка метафайла
- *save_metafile* – сохранение метафайла

6.2.3 Класс ImagePreview

Методы класса:

- *add_image* – добавление нового изображения в блок предпросмотра
- *clear* – очистка блока предпросмотра

6.2.4 Класс ImageScene

Методы класса:

- *rectangle_changed* – изменяет параметры объемлющего прямоугольника для изображения
- *set_rectangle* – устанавливает объемлющий прямоугольник для отображения
- *set_image* – устанавливает изображение для отображения

6.3 Проект интерфейса

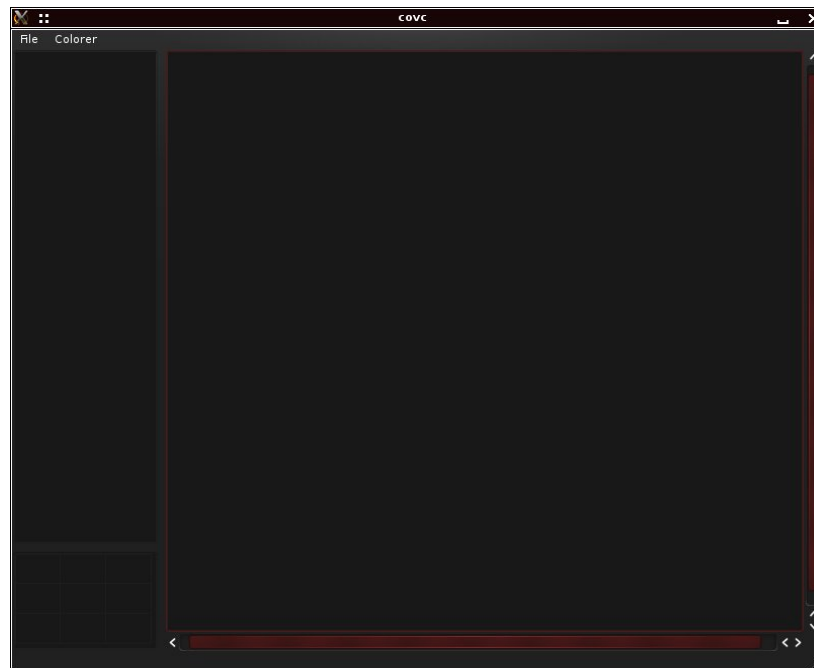


Рис. 1: Проект интерфейса

Список литературы

- [1] Academic Free License ("AFL") v. 3.0 [Электронный ресурс] : электрон. энциклопедия – Режим доступа: <http://www.opensource.org/licenses/academic.php>
- [2] ATI, ATI Stream Software Development Kit [Электронный ресурс] – Режим доступа: <http://developer.amd.com/gpu/ATIStreamSDK/Pages/default.aspx#two>
- [3] Domenico Mangieri, OpenCLxx [Электронный ресурс] – Режим доступа: <http://openclxx.sourceforge.net/>
- [4] IBM, OpenCL Development Kit for Linux on Power [Электронный ресурс] – Режим доступа: <http://www.alphaworks.ibm.com/tech/opencl>
- [5] Leica ScanStation 2 3D Laser Scanner [Электронный ресурс] // FLT Geosystems website – Режим доступа: <http://www.fltgeosystems.com/>
- [6] Eisert, P. Multi-Hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views / Peter Eisert, Eckehard Steinbach, Bernd Girod // ICASSP'99, Phoenix, USA – march 1999. – PP. 3509-3512
- [7] Khronos Group, OpenCL [Электронный ресурс] – Режим доступа: <http://www.khronos.org/opencl/>
- [8] Koen van de Sande, Voxel Coloring Framework [Электронный ресурс] – Режим доступа: <http://voxelcoloring.sourceforge.net/>
- [9] Nokia, QRectF [Электронный ресурс] – Режим доступа: <http://doc.trolltech.com/4.6/qrectf.html>
- [10] Nokia, QString [Электронный ресурс] – Режим доступа: <http://doc.trolltech.com/4.6/qstring.html>
- [11] NVIDIA, CUDA GPUs [Электронный ресурс] – Режим доступа: http://www.nvidia.com/object/cuda_gpus.html
- [12] S3 Graphics [Электронный ресурс] – Режим доступа: <http://www.s3graphics.com/>
- [13] Воксел [Электронный ресурс] : электрон. энциклопедия – Режим доступа: <http://ru.wikipedia.org/w/index.php?title=Voxel&oldid=22172795>
- [14] Кроссплатформенное программное обеспечение [Электронный ресурс] : электрон. энциклопедия – Режим доступа: <http://wikipedia.org/>
- [15] Комарова, Н. Практическая реализация алгоритма раскраски вокселей : курсовая работа / Н. Комарова – 2005. - 23 с.
- [16] Фреймворк [Электронный ресурс] : электрон. энциклопедия – Режим доступа: <http://ru.wikipedia.org/wiki/Framework>