

Санкт-Петербургский Государственный Университет
Математико-механический факультет
Кафедра системного программирования

3D Registration

Курсовая работа студентки 445 группы
Фоменко Екатерины Сергеевны

Научный руководитель:
Пименов А.

Оглавление

1. Введение.....	3
1.1. Постановка и актуальность задачи.....	3
1.2. Средства разработки(языкC++).	3
1.3. Существующие исследования.....	4
2. Решение поставленных задач.....	5
2.1. Инструментарий.....	5
2.2. Библиотека PCL.....	5
2.3. Алгоритм ICP.....	5
2.3.1. Поиск ближайшего соседа.....	5
2.3.2. Оценка параметров преобразования с помощью функции среднеквадратичной стоимости	6
2.4. Проблемы ICP и их решение.....	7
3. Тестирование на последовательностях изображений.....	8
4. Заключение.....	12
5. Возможности для развития.....	12
6. Список литературы.....	13

1. Введение

Наземные лазерные сканеры позволяют получать точную трехмерную модель объекта в виде облака точек. На практике объекты необходимо сканировать с различных точек зрения, а затем результаты объединяются в единую систему координат (данную процедуру часто называют регистрацией или взаимным ориентированием). То, регистрация — это совмещения различных 3D сцен (облаков точек) в одну целостную сцену. Её целью является найти взаимное расположение и ориентацию одного изображения сцены относительно другого, и наилучшим образом совместить перекрывающиеся области точек.

Нам необходимо было найти алгоритм, который бы для каждого двух облаков точек, полученных при помощи сканирования изображения с разных сторон, находил оптимальное совмещение.

В качестве такого алгоритма был использован алгоритм ICP (iterative closest point).

1.1 Постановка и актуальность задачи

В рамках данной курсовой работы необходимо было найти алгоритм, который для каждой пары облаков точек, полученных в результате сканирования изображения с разных сторон, находил оптимальное совмещение.

Эффективность написанной мной программы, использующую этот алгоритм, нужно протестировать на определённых последовательностях изображений.

Задача регистрации облаков актуальна в области построения трёхмерных карт городов, построения трёхмерного окружающего пространства для определения местоположения роботов, регистрация модели кости(в медицине), планирование оптимального пути.

1.2. Средства разработки (язык C++)

В качестве операционной системы был выбран Windows 7. В качестве среды разработки – Microsoft Visual Studio 2010. Данная среда предоставляет множество различных средств для работы с языком C++.

Язык C++ - это язык программирования, который позволяет разрабатывать программы в соответствии с разными парадигмами: процедурным программированием, объектно-ориентированным, параметрическим.

Разработчиком языка C++ является Бьерн Страуструп. В своей работе он опирался на опыт создателей языков Симула, Модула 2, абстрактных типов данных. Основные работы велись в исследовательском центре компании Bell Labs.

C++ в обработке изображений используют за хорошую скорость, возможность низкоуровневых оптимизаций и в то же время поддержку продвинутых высокоуровневых конструкций.

1.3. Обзор существующих исследований

Похожая задача исследуется в Технологическом институте Карлсруэ. Они используют библиотеку LibVisio2 для совмещения 3D сцен. Их целью является построение *трёхмерных карт города* (См. Рис.1).

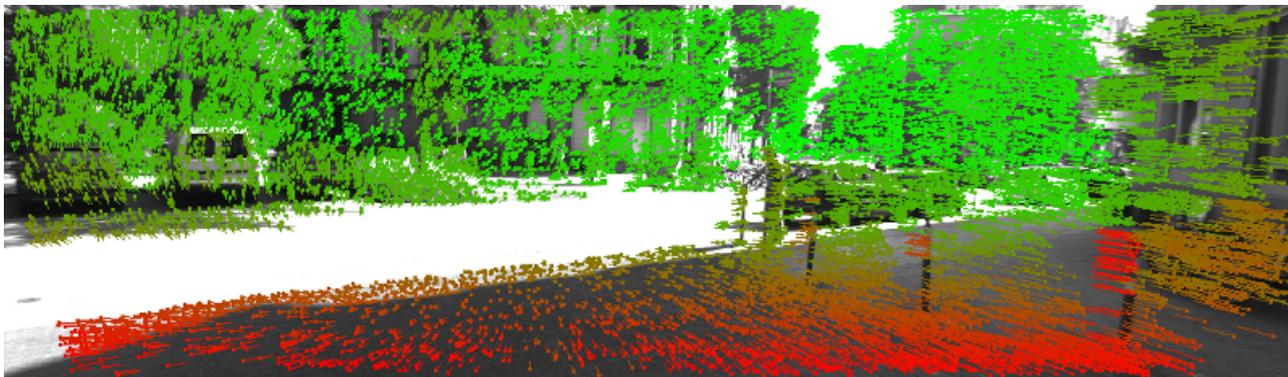


Рис.1

Стоит рассмотреть ещё одно очень важное исследование, основные идеи которого были высказаны в сентябре 2011 года. Проект называется «*Беспилотный автомобиль*», созданный компанией Google.

Ядром системы является 64-лучевой лазерный светодальномер, установленный на крыше автомобиля. Прибор помогает сгенерировать детальную объемную карту окружающего пространства. Затем машина комбинирует измерения лазера с высокоточными картами мира и производит различные типы моделей данных, которые позволяют вести движение, избегая препятствия и проблемы с законом. Также на борту установлены другие сенсоры, включая четыре радара на переднем и заднем бамперах, камеру рядом с зеркалом заднего вида; датчик системы Джи-Пи-Эс, блок инерциальных измерений и колесный датчик, которые определяют положение транспортного средства и отслеживают движение.

Такая разработка «Гугла» сильно зависит от тщательно проработанных карт поверхности Земли, что, является неотъемлемым условием для точного определения местоположения машины. Использование системы глобальной навигации может дать погрешность в несколько метров.

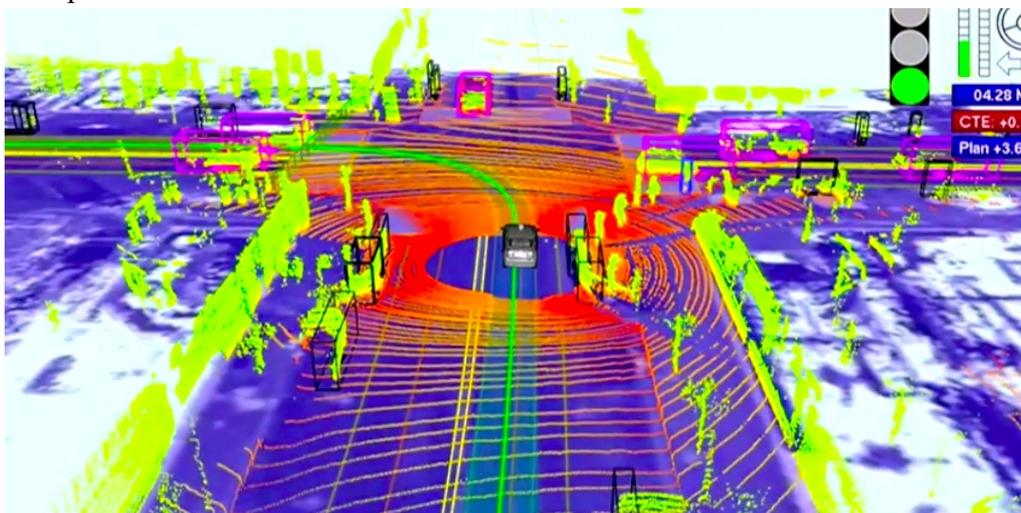


Рис.2. Беспилотный автомобиль от Google

В основе данной разработки лежит регистрация облаков точек для точного построения окружающего пространства.

2. Решение поставленных задач

2.1. Инструментарий

В качестве инструментария в своей задаче я использовала следующее:

- Библиотека PCL(Point Cloud Library)
- Формат облака точек pcd
- Визуализатор для отображения облака точек pcd_viewer
- Алгоритм ICP(Iterative Closest Point)

2.2. Библиотека PCL

Point Cloud Library- библиотека для работы с облаком точек, предназначенная для обработки 3D геометрии и n-мерного облака точек.

Она включает в себя также различные реализации алгоритма ICP, использующие подход с пересекающимися и не сильно повернутыми друг относительно друга облаками точек.

2.3. Алгоритм ICP

Данный алгоритм, предложенный Besl and McKay, является классическим решением задачи регистрации множеств. Этот итеративный алгоритм состоит из трех основных шагов:

1. Связка точек по критерию ближайшего соседа (поиск для каждой точки множества P пары из множества M).
2. Оценка параметров преобразования с помощью функции среднеквадратичной стоимости (вычисление преобразования (смещение + поворот), минимизирующего среднеквадратичное расстояние (MSE) между парными точками).
3. Применение рассчитанного преобразования ко множеству P и обновление среднеквадратичной ошибки.

Эти три шага повторяются. Доказано, что они обеспечивают равномерную сходимость MSE(гл. 2.3.2).

Рассмотрим каждый шаг подробно.

2.3.1. Поиск ближайшего соседа

Задача поиска ближайшего соседа заключается в отыскании среди множества элементов, расположенных в многомерном метрическом пространстве, элементов близких к заданному, согласно некоторой функции близости.

Для решения этой задачи используется модель k-d деревьев. **K-d дерево** представляет собой бинарное дерево, которое строится по следующему правилу:

- начальное множество разбивается по значениям векторов в одной из координат, например, по i , где $i \in \{1, \dots, k\}$, на два подмножества;
- i выбирается таким образом, чтобы разброс значений по данной координате был

максимальным;

- разбиение проводится по медиане m , так что одинаковое количество точек оказывается с одной и с другой стороны;
- в вершине дерева хранятся значения i , m , разброс значений векторов по каждой координате;
- для полученных вершин процесс повторяется.

В итоге получим бинарное дерево глубины $d = \log_2 N$

Для поиска n ближайших соседей к вектору q в построенном дереве:

- сначала дерево обходится вниз до листа содержащего «ближайшую» к q точку. Эта точка не обязана быть ближайшей, это только первое приближение;
- во время спуска по дереву заполняется список поддеревьев, которые еще не обходили. Также запоминаются расстояния до них, которое определяется как минимальное расстояние от точки q до любой точки, находящейся в границах значений поддерева;
- из списка выбираем ближайшее к q поддерево и продолжаем поиск в нём;
- расстояние до каждого нового найденного претендента сравнивается с радиусом сферы найденных точек с центром в точке q . Если данное расстояние меньше, то точку на сфере заменяем этим претендентом.

Алгоритм работает до тех пор, пока в списке есть поддеревья с расстоянием, меньшим радиуса сферы найденных точек.

Среднее время построения kd-дерева $O(n \log n)$. Kd-дерево строится только один раз для каждого облака точек.

Среднее время нахождения ближайшего соседа по kd-дереву $O(\log n)$.

2.3.2. Оценка параметров преобразования с помощью функции среднеквадратичной стоимости

На этом шаге алгоритм производит поиск соответствующих ближайших точек и вычисляет трансформацию (смещение+поворот) для минимизации уравнения:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2, \quad (1)$$

где N_m и N_d - количество точек в соответствующих облаках M и D , w_{ji} - веса пар точек.

Значения весов определяются следующим образом:

$w_{ji} = 1$, если \mathbf{m}_i является ближайшей точкой к \mathbf{d}_j , $w_{ji} = 0$ иначе.

Равенство (1) можно переписать следующим образом:

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})\|^2, \quad (2)$$

где $N = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j}$

Матрица соответствия может быть представлена вектором, содержащим пары точек, т.е.

$$\mathbf{v} = ((\mathbf{d}_1, \mathbf{m}_{f(\mathbf{d}_1)}), (\mathbf{d}_2, \mathbf{m}_{f(\mathbf{d}_2)}), \dots, (\mathbf{d}_{N_d}, \mathbf{m}_{f(\mathbf{d}_{N_d})}))$$

где $f(x)$ -функция поиска, возвращающая ближайшую точку.

Предполагается, что на последнем шаге итерации нахождения соответствующих пар точек, данный вектор окажется верным.

На каждой итерации ИСР трансформация может быть посчитана при помощи любого из четырёх методов:

- Метод SVD (сингулярное разложение матрицы)[1]
- Метод кватернионов [2]
- Метод с использованием ортонормированных матриц [3]
- Метод, связанный с вычислением двойных кватернионов [4]

Итерации заканчиваются, когда будет сведен квадратичная ошибка станет достаточно мала.

В данной реализации алгоритма использовался *метод кватернионов*.

2.4. Проблемы ИСР и их решение

Одной из основных проблем ИСР является ограничение области сходимости: алгоритм работает только при условии, что облака точек не сильно сдвинуты друг относительно друга. Иначе необходимо вручную задавать начальную матрицу трансформаций. Обычно она вычисляется из каких либо внешних параметров (изменение угла обзора камеры).

Для решения этой проблемы Велижевым А.Б. было предложено разбить решение задачи взаимной ориентации точечных моделей на два главных этапа:

- оценка матрицы угловой ориентации R;
- оценка вектора сдвига T.

Для оценки матрицы угловой ориентации для исходного точечного представления объекта выполнялось построение ориентационной гистограммы, обладающей важным свойством инвариантности к преобразованию сдвига. Получение ориентационной гистограммы для точечной модели требует вычисления нормалей в каждой точке. Для эффективного вычисления нормалей произвольно организованная точечная модель преобразовывается в восьмеричное дерево, каждая ячейка которого содержит исходные точки.

После нахождения угловой ориентации каждая точка второй точечной модели P_2 поворачивается в соответствии с найденными углами. В результате чего, точечные модели оказываются выровненными друг относительно друга по угловой ориентации. Затем выполняется оценка сдвига с помощью сравнения воксельных представлений каждой из точечных моделей на основе дискретного преобразования Фурье.

3.Тестирование на последовательностях изображений.

В качестве тестовых данных были рассмотрены две последовательности:

- Последовательность *Tsukuba* – последовательность кадров(изображений) внутреннего пространства комнаты. Известная последовательность, часто используемая для алгоритмов регистрации. Облака точек этой последовательности были получены при помощи алгоритмов 3vi. На рис.2 Показано изображение комнаты.



Рис.2.Кадр из последовательности Цукуба

На вход было передано три облака точек.

На рис.3 показана последовательность входных данных, а на рис.4 -то, что получился в результате применения алгоритма.

На первый взгляд различия сложно увидеть, но, присмотревшись, можно заметить, что на рис.4 облака точек совмещены лучше.

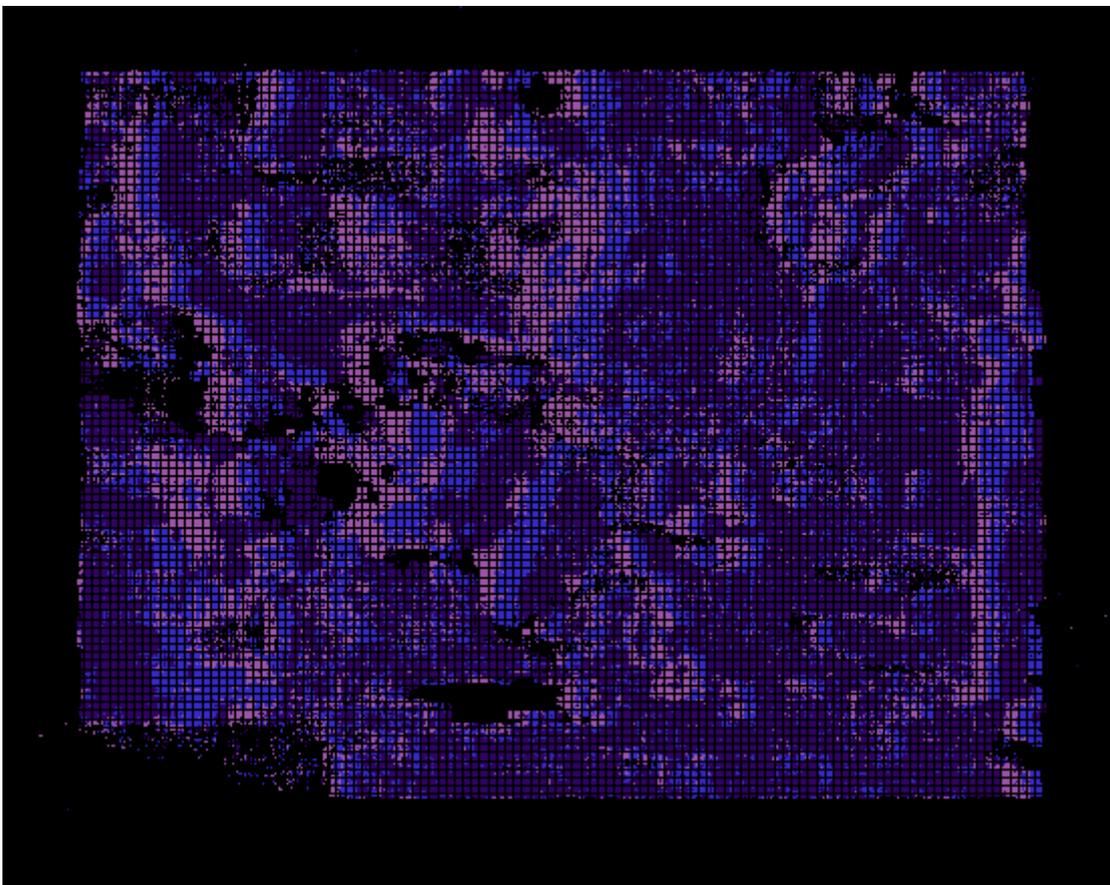


Рис.3.Входные облака точек

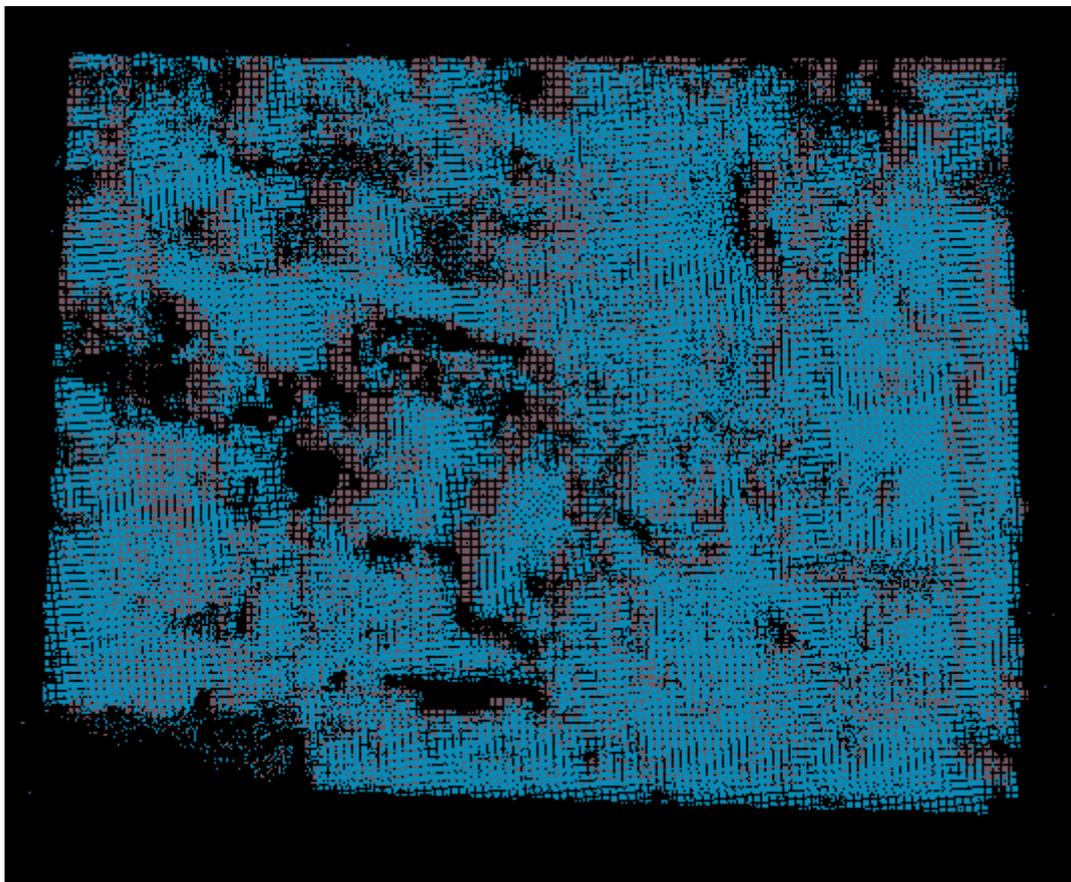


Рис.4.Облака точек, совмещённые при помощи ICP

- Последовательность *Даймлера*- последовательность изображений, полученная с помощью камеры, прикреплённой к машине. Машина равномерно с небольшой скоростью движется по дороге. Камера прикреплена к лобовому стеклу и снимает улицу, вдоль которой движется машина(Рис.5).



Рис.5.Кадр из последовательности Даймлера

При работе с данной последовательностью возникли некоторые трудности: входные данные очень зашумлены, из-за этого алгоритм плохо воспринимает различия между облаками. Для наилучшего совмещения облаков точек данной последовательности, требуется фильтрация шума.

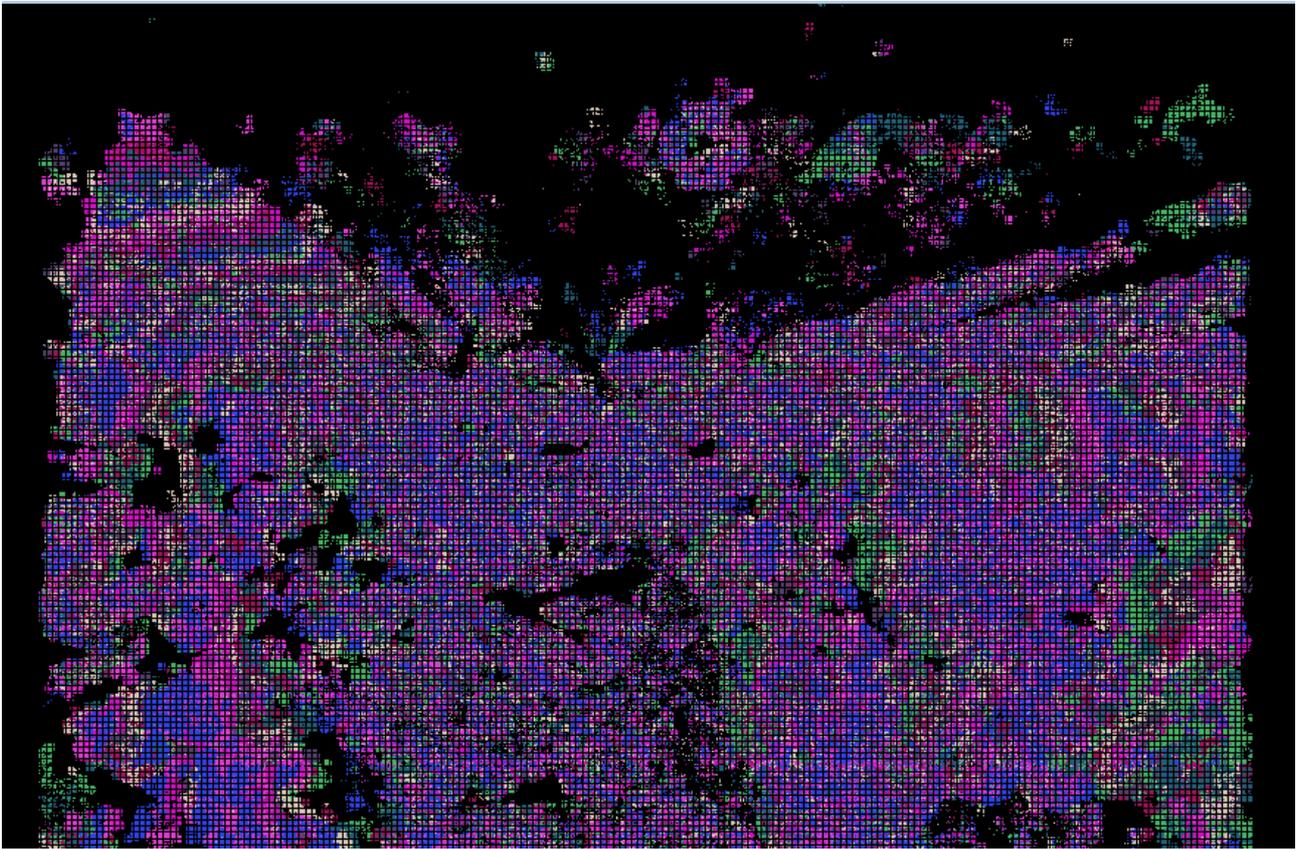


Рис.6. Входные последовательности изображений

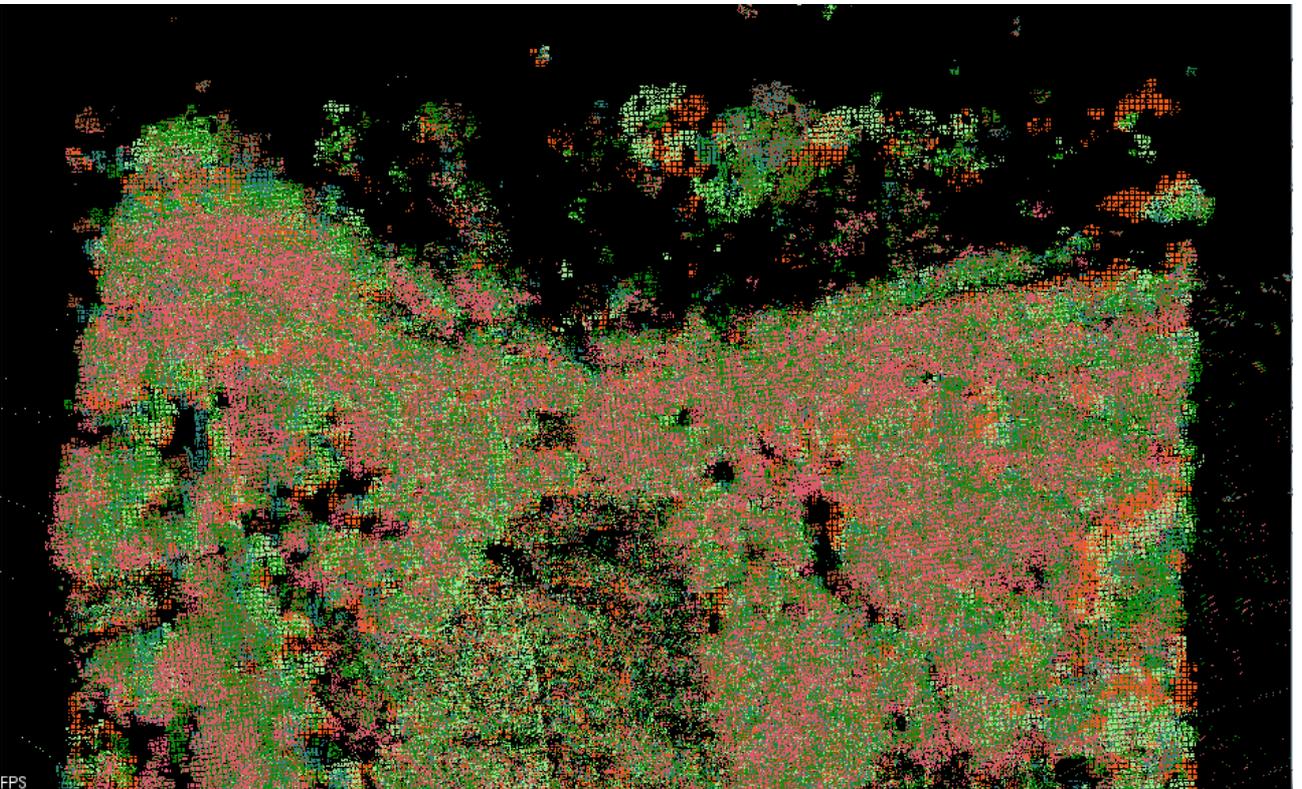


Рис.7. Полученный результат

5. Заключение

В рамках данной курсовой работы была представлена реализация алгоритма ICP, которая даёт вполне приемлемые результаты на реальных данных.

Разобраны этапы работы алгоритма ICP. Сложность предложенных методов в данном алгоритме наиболее оптимальна.

Использованный формат `pcd` не часто встречается при работе с облаками точек. В связи с этим был написан конвертер `pcd` в `ply`, реализованный на языке Python. Конвертер `ply` в `pcd` есть в PCL.

Была написана программа, которая:

- обрабатывает входные данные
- настраивает необходимые параметры для применения алгоритма к последовательности изображений в формате `pcd`
- отображает промежуточные результаты
- сохраняет конечные результаты

Данная реализация алгоритма не очень эффективна в случае с сильно зашумлёнными входными данными. Также её применение не возможно в случае сильно поворнутых сцен, так как изначальная матрица трансформации одного облака точек относительно другого не высчитывается.

6. Возможности для развития

- Использование работы Велижева[5] для *регистрации сильно поворнутых* относительно друг друга сцен
- Нахождения механизма *эффективной фильтрации* шума облаков точек
- Планируется внедрение в проект *3Vi/CVS*

6.Список литературы

1. K. S. Arun, T. S. Huang, and S. D. Blostein.

Least square fitting of two 3-d point sets.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5):698 - 700, 1987.

2. B. K. P. Horn.

Closed-form solution of absolute orientation using unit quaternions.

Journal of the Optical Society of America A, 4(4):629 - 642, April 1987.

3. B. K. P. Horn, H. M. Hilden, and Sh. Negahdaripour.

Closed-form solution of absolute orientation using orthonormal matrices.

Journal of the Optical Society of America A, 5(7):1127 - 1135, July 1988.

4. M. W. Walker, L. Shao, and R. A. Volz.

Estimating 3-d location parameters using dual number quaternions.

CVGIP: Image Understanding, 54:358 - 367, November 1991.

5. Автоматическое регистрация облаков точек на основе ориентационных гистограмм

<http://graphics.cs.msu.ru/ru/science/research/3dpoint/alignement>