**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Прикладная математика и информатика»

**Отчет о программном проекте**

на тему **«Реконструкция геометрии 3D»**

(итоговый)

**Выполнила**:

студентка группы БПМИ187 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Андреева Д. А ф

Подпись И.О. Фамилия

Ф 20.06.2020. ф

Дата

**Принял**:

руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Имя, Отчество, Фамилия

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Должность

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Место работы

Дата \_\_\_\_\_\_\_\_\_2020 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка (по 10-тибалльной шкале) Подпись

**Москва 2020**

Аннотация

В этой работе рассматривается алгоритм перевода поинтклауда в трехмерное изображение помещений, а также создания их дальнейшего 2D плана. Итогом проекта будет программа на языке Python по рассмотренному алгоритму.

[Введение 3](#_Toc43418233)

[Основные определения и термины 4](#_Toc43418234)

[Требования к проекту 5](#_Toc43418235)

[Функциональные требования 5](#_Toc43418236)

[Нефункциональные требования 5](#_Toc43418237)

[Выбранные алгоритмы и методы 6](#_Toc43418238)

[Общий алгоритм преобразования панорамного (360о) фото в 3D модель 6](#_Toc43418239)

[Формат .ply и .pts 7](#_Toc43418240)

[RANSAC 7](#_Toc43418241)

[Описание подходов к реализации 8](#_Toc43418242)

[Драфт проекта 9](#_Toc43418243)

[Список используемой литературы 10](#_Toc43418244)

# **Введение**

Сейчас вместе с эволюцией компьютерных технологий, запросов пользователей и их интересов, когда стало возможно не только воспроизводить 8 битов, а обрабатывать невероятно сложные конструкции, немаловажной частью реальности стало 3D. Смоделировать комнату – легко! Встретиться с любимым женским персонажем (или с Наклзом из Уганды…) – еще проще. Никто даже не сможет поспорить сейчас с тем, что за 3D технологиями стоит будущее. Особенно это касается виртуальной реальности.

Понятно, что сейчас не каждый может позволить себе шлем виртуальной реальности, но на мой взгляд, когда технология будет достаточно хорошо отточена, это станет обыденностью, как микроволновка или телефон с сенсорным экраном. И вот тогда и мой проект, и 3D технологии станут популярными как никогда.

Не раз Вы сталкивались с тем, что фото на сайте по поиску квартиры просто ужасны и малоинформативны. Даже план квартиры уже был бы хорош, согласитесь? А походить по потенциальному вложению денег, не выходя при этом из дома? Еще лучше! В общем, идея на самом деле рабочая.

Хороший вариант и VR путешествия по музеям, театрам, библиотекам: уже сейчас, в условиях пандемии они показали действительно хорошие результаты. Кроме того, виртуальная реальность позволит людям попасть туда, куда попасть простым смертным невозможно даже в свободное от пандемии время, например, в Апостольскую библиотеку в Риме, куда невозможно попасть по причине ее хрупкости, или в зал конгресса в здании Венгерского парламента, который всегда используется для заседаний.

Идея превращения панорамного фото в поинтклауд, а далее в 3D модель хороша и в отрыве от VR. Можно будет отреставрировать по старым фото помещение на компьютере, как сейчас восстанавливают животных по их черепам, возможно даже хранить многомилионную базу человеческого наследия, оставляя шансы потомкам поддерживать то, что мы имеем сейчас.

Хороша еще и идея для дизайнерских разработок: достаточно удобно загрузить комнату сразу, используя только фото.

Основные определения и термины

***Сегментация изображения*** — задача поиска групп пикселей, каждая из которых характеризует один смысловой объект. В статистике эта проблема известна как кластерный анализ и является широко изученной областью с сотнями различных алгоритмов. В компьютерном зрении сегментация изображения является одной из старейших и широко изучаемых проблем.

***BIM*** (англ. ***Building Information Model*** или ***Modeling***) — информационная модель (или моделирование) зданий и сооружений, под которыми в широком смысле понимают любые объекты инфраструктуры, например инженерные сети, дороги, железные дороги, мосты, порты и тоннели и т. д.

***RANSAC*** (аббр. ***RANdom SAmple Consensus***) — стабильный метод оценки параметров модели на основе случайных выборок.

***Задача классификации*** — задача, в которой имеется множество объектов (ситуаций), разделённых некоторым образом на классы.

***Point clouds (облако точек)*** — набор вершин в трёхмерной системе координат. Эти вершины, как правило, определяются координатами X, Y и Z и, как правило, предназначены для представления внешней поверхности объекта.

***Октодерево*** (дерево октантов, англ. ***octree***) — тип древовидной структуры данных, в которой у каждого внутреннего узла ровно восемь «потомков».

Требования к проекту

Функциональные требования

1. Интерфейс программы - консольное приложение
2. Работа с файлами .ply (чтение и запись), хранящими облака точек
3. Сегментация облака точек на классы: пол, потолок и стены
4. Сохранение результата сегментации в удобном для дальнейшей работы формате
5. Визуализация облака точек в 3D пространстве
6. Визуализация результата сегментации облака точек
7. Получение двухмерного плана помещения (расположение стен) из облака точек
8. Сохранение плана в виде изображения
9. Выполнение перечисленных функций сразу для нескольких файлов .ply

Нефункциональные требования

1. Требования к надежности

1.1. Программа не должна завершаться некорректно. В случае, когда невозможно сегментировать/визуализировать/построить план по входному файлу, программа должна сообщать о проблеме пользователю.

1.2. Программа не должна изменять входные файлы .ply, а также другие файлы системы

1. Условия эксплуатации

Условия эксплуатации ограничены условиями эксплуатации настольных ПК

1. Требуемая квалификация и уровень подготовки пользователя

Пользователь должен уметь работать с терминалом.

1. Требования к информационной и программной совместимости

4.1. Требования к входным данным

Файлы .ply с информацией о координатах и цвете точек, количество точек не более 20 миллионов.

4.2. Требования к языкам программирования

Программа должна быть написана на языке Python не ниже версии 3.4.

Выбранные алгоритмы и методы

Общий алгоритм преобразования панорамного (360о) фото в 3D модель

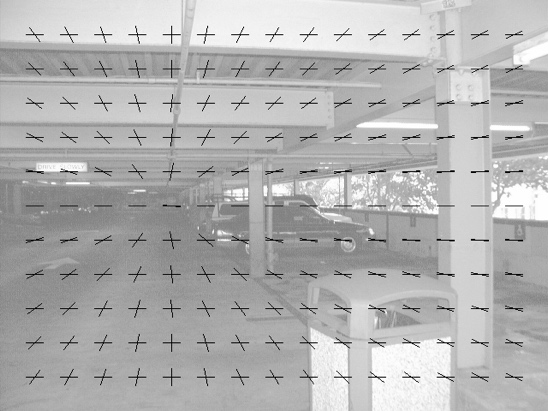
Выбранный мной вариант реализации – получение поинтклауда изображения, его сегментация на пол, стены и потолок. Будем опираться на статью [1] и модель Xiao и Furukawa (2012) о модели музея в Швеции. Предлагается следовать Manhattan World Assumption [2]: имеется три ортогональных измерения, полы и потолки по отношению к стенам перпендикулярны, параллельны плоскости OXY. Стены, соответственно, параллельно плоскости OZY или OZX.

Рисунок . Manhattan World

Недостаток модели Xiao и Furakawa был в том, что пользователю требовалось самостоятельно классифицировать объекты, наша же задача – это автоматизировать. Кроме того, по возможности, добавить объем стен, потому что они, очевидно, имеют какую-то толщину в реальной жизни – это поможет сделать план не только одной комнаты, но квартиры в целом.

Изображение выглядит как снимок экрана, дисплей, телефон

Автоматически созданное описание

Рисунок . Предположительное решение

Для начала алгоритм получает на вход данные облака точек, зарегистрированных и выровненных по главным осям. С помощью RANSAC сегментируем изображение на плоские поверхности, получаем форму поинтклауда. Таким образом, превращаем наш поинтклауд в пронумерованные плоские поверхности (они извлекаются в порядке убывания количества точек). Вычисляются нормали и положение плоскостей.

Далее плоскости проецируются на плоскости OXZ и OYZ. Добавляем октодеревья, определяем коллизии и предполагаем, что горизонтальные поверхности – пол и потолок. Впоследствии мы удаляем большую часть этого хаоса, сохраняя вертикальные плоские поверхности, которые удовлетворяют определенным критериям. Остальные сегменты делятся на две категории: та, которая параллельна плоскости OYZ и плоскости OXZ; сохраняем плоские поверхности в периметре, которые находятся на минимальном расстоянии от ограничительной рамки облака точек, и объединяем плоские поверхности, которые соответствуют кубовидным стенкам. Итоговый результат — это пол, стены и потолок.

2D план по модели будет производиться путем простой проекции стен на плоскость OXY.

Выделение плоскостей

*Данный раздел почти полностью основан на работе Lin Li, Fan Yang, Haihong Zhu, Dalin Li, You Li и Lei Tang «****An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells****» - Усовершенствованный RANSAC для сегментации плоскостей трехмерного облака точек на основе ячеек трансформации к нормальному распределению.*

Главная идея той статьи – плоскостная ячейка преобразования нормального распределения выбирается как минимальный образец поверхности той же плоскости, чтобы обеспечить корректность отбора. Первичная плоскость выбирается первой, а остальные точки заново тестируются на принадлежность ей, чтобы получить итоговую плоскость. Предполагаемый метод увеличивает корректность плоскостной сегментации и устранить проблемы обычного RANSAC. Итеративный метод наименьших квадратов приближения также используется для повышения точности и надежности обнаруженной плоскости.

### **NDT метод**

Так как поинтклауд изобилует геометрической информацией, необходимо учитывать соседние точки при обработке поинтклауда. Каждая точка зачастую используется и рассчитывается с учетом ее соседей. Например, для сегментации и сжатия поинтклауда используются такие характеристики, как нормаль, гладкость и кривизна. Предложенный метод основан на ячейках с преобразованием нормального распределения (NDT), которые представляют собой дискретное представление пространства облаков точек, которое учитывает каждую ячейку, а не каждую точку.

Впервые NDT метод был предложен в 2013 году Бибером и Штрассером для 2D пространств, а позже расширен до 3D. Идея этого метода состоит в разбиении пространства на подпространства (2D на квадраты, а 3D на кубы), после чего для каждого вычисляется PDF (Probability Density Function) – функция плотности вероятности на основе распределения точек внутри ячейки. Существует два варианта разбиения: когда ячейки одного размера (регулярная дискретизация), и октодерево (нерегулярная дискретизация).

RANSAC

RANSAC был представлен Фишером и Боллесом в 1981 году. Его основная идея состоит в двух итеративно повторяющихся шагах: выдвигать гипотезу, а потом ее проверять. Выбирается минимальное количество точек, необходимое для построения той или иной поверхности (для плоскости, например, это три точки, для линии – две). Затем оставшиеся точки в поинтклауде проверяются с помощью полученных форм-кандидатов, чтобы определить, сколько точек хорошо аппроксимируется моделью. После определенного числа итераций форма, которая имеет наибольший процент элементов, извлекается, и алгоритм продолжает обрабатывать оставшиеся данные.

Предположительный план действий для выделения плоскостей:

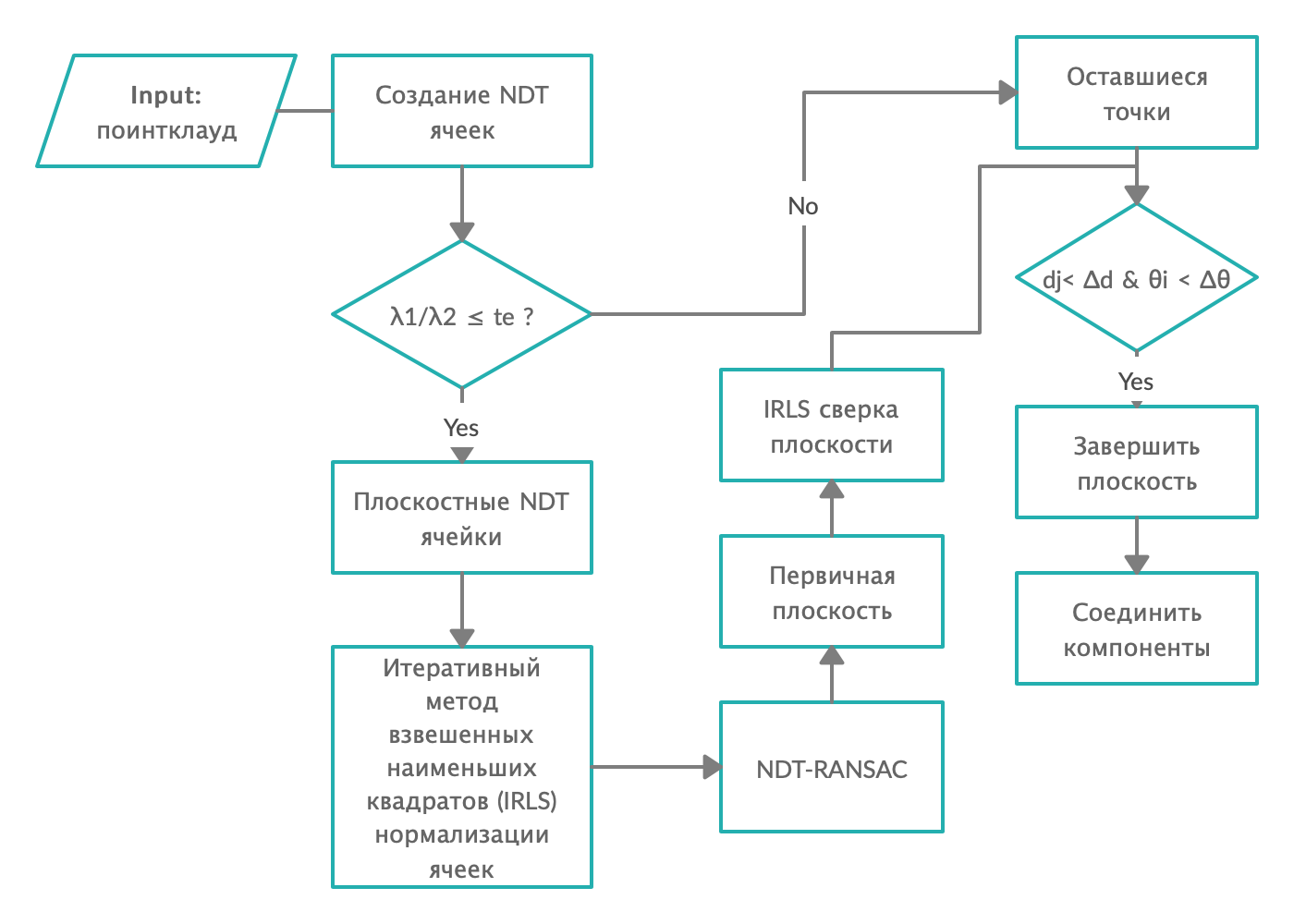


Рисунок . Основные шаги RANSAC-NDT. dj – расстояние между точкой до плоскости и θi угловая разница нормалей к плоскости. ∆d и ∆θ показывают порог расстояния точка-плоскость и порог согласованности между векторами нормалей, соответственно.

Дискретизируем 3D пространство на набор ячеек, где моделируем нормальное распределение. В каждой ячейке нормальное распределение с выборочным средним и ковариацией , параметрами многомерного нормального распределения , в каждой ячейке точек. Матрица ковариации определяет форму своей ячейки и положительно определена. Имеем собственные значение для каждой ячейки и ее матрицы ковариаций: и собственные векторы: . В зависимости от отношений собственных чисел мы можем понять, как выглядит ячейка разбиения ( – ось z, – ось y, – ось x).

* Если , распределение NDT ячейки линейное,
* Если нелинейное, но , распределение NDT плоскостное
* Иначе, если ни одно собственное значение не больше в раз, чем другое, имеем сферу

Значение *te* ограничивает отношение между третьей осью и второй осью, например, если *te* = 1/25 = 0,04, то отношение между третьей осью и второй осью равно 1/5; если *te* = 1/100 = 0,01, то отношение между третьей осью и второй осью равно 1/10. Эмпирически рекомендуется *te* < 0,04. Чем меньше значение *te*, тем более плоской является ячейка NDT. Изображение выглядит как часы

Автоматически созданное описание

Рисунок . Вид ячейки в зависимости от собственных значений матрицы ковариаций

Вычислив собственные значения и классифицировав точки, можно приступать к следующему шагу. Принимаем два решения: во-первых, большинство NDT ячеек, расположенных на плоских поверхностях, имеют плоский вид. Во-вторых, NDT-ячейки, расположенные на одной и той же поверхности, имеют сходные плоскостные параметры.

RANSAC, основанный на NDT-ячейках немного быстрее обычного – потому что нет необходимости высчитывать нормаль каждой точки, нужно высчитывать только нормаль ячейки.

Максимальное количество итераций зависит от априорной вероятности выборки инлаеров над данными. Имеем поинтклауд P, Q – массив плоскостных ячеек размера N, R – плоскость, содержащая n ячеек. Так как ранее мы договаривались, что ячейка состоит минимум из трех точек, то уже из них мы можем выбрать минимальную подплоскость. Тогда P(n) данных являются инлаерами, что означает, что вероятность случайного выбора каждой ячейки на плоскости R в эксперименте равна P(n) = , а следовательно вероятность *не* выбора ничего за k итераций – , вероятность нахождения хоть чего-то за k итераций – .

Чтобы с уверенностью убедиться, что ячейка, свободная от мусора, нужно обработать k кандидатов, где . Параметр обычно где-то между 0.90 и 0.99

Реализация

Данная задача может быть достигнута как путем нейронных сетей, так и путем сегментации изображения.

Рассматривался вариант обучения нейросети FloorNet, однако она требует данные для обучения, плохо работает с разреженным облаком данных и нестабильно на плотном поинтклауде.

Еще один хороший вариант – статья A Whole-room 3D Context Model for Panoramic Scene Understanding» Y. Zhang Sh. Song P. Tan† J. Xiao.

Другой вариант – получение поинтклауда изображения, его сегментация на пол, стены и потолок. Он и был мною выбран.

Для написания кода было решено использовать Jupiter notebook / Google colab – он удобен тем, что нет необходимости устанавливать тысячи библиотек на свой компьютер и легко их всех будет подключать, да и в целом, руководитель это одобрил. По его же рекомендации было решено использовать библиотеку Open3D для работы с поинтклаудами, сами данные для тестов были предоставлены ментором.

В качестве второго шага требовался RANSAC для трехмерных случаев. Я решила использовать улучшенный RANSAC на основе NDT. Очевидно, что если есть что-то улучшенное, то лучше воспользоваться этим. Свои собственные эксперименты касательно эффективности данного метода относительно предшественника я не проводила, но в статье, на которой был основан код, они приведены:

Изображение выглядит как снимок экрана

Автоматически созданное описание

## Драфт проекта

<https://gitlab.com/labsallday/interns/hse_floorplan_reconstruction> - основной репозиторий проекта на GitLab.

Список используемой литературы

1. «**Detection of Walls, Floors, and Ceilings in Point Cloud Data**» I. Anagnostopoulos; V. Pătrăucean; I. Brilakis; и P. Vela. Construction Research Congress 2016
2. **«PanoContext: A Whole-room 3D Context Model for Panoramic Scene Understanding»** Y. Zhang S. Song P. Tan J. Xiao
3. «**Manhattan-World**» Coughlan & Yuille (1999)
4. «**An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells**» Lin Li, Fan Yang, Haihong Zhu, Dalin Li, You Li и Lei Tang 2017
5. «**Навигация с применением RANSAC фильтрации точечных особенностей**» Машенцев Владимир Юрьевич 2013
6. «**Анализ изображений и видео. Часть 1**», Наталья Васильева, Антон Конушин