

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ЛАБОРАТОРИЯ КОМПЬЮТЕРНОЙ ГРАФИКИ И МУЛЬТИМЕДИЯ

Дипломная работа

Сегментация и распознавание объектов в облаках
трехмерных точек на основе схемы голосования

студент: Лебедев В. В.

группа 521

научные руководители: к.ф.-м.н., н.с. Конушин А.С.

к.т.н. Велижев А.Б.

Москва, 2012

Аннотация

В рамках этой работы рассматривается проблема автоматизированного сегментирования объектов в трехмерных сценах городского ландшафта, полученных методом лазерного сканирования. Приводится обзор существующих методов, а также предлагается новый метод, основанный на схеме голосования за центр объекта с его последующим выделением из сцены. Метод был протестирован на базе, полученной сканированием реальных городских объектов в Оттаве, Канада [21]. Сравнение с существующими методами показало сравнимую точность при отсутствии необходимости ручного подбора параметров сегментации.

Содержание

1	Введение	4
2	Постановка задачи	7
2.1	Неформальное описание задачи	7
2.2	Формальная постановка задачи	7
2.3	Критерии качества	8
2.4	Цели работы	8
3	Обзор существующих методов	10
3.1	Предобработка	10
3.1.1	Сглаживание	10
3.1.2	Фильтрация шума	10
3.1.3	Выделение объектов интереса	11
3.1.4	Введение дополнительных структур данных	11
3.2	Поиск начального приближения центра объекта	11
3.3	Сведение к задаче машинного обучения	13
3.3.1	Вычисление признаков	14
3.3.2	Обучение/классификация	15
3.4	Сегментация на основе минимального разреза графа	17
3.5	Неявные модели формы	18
4	Предлагаемый метод	20
4.1	Построение графа	20
4.2	Схема голосования	20
4.3	Задание энергии	23
4.3.1	Унарный потенциал	23
4.3.2	Парный потенциал	24
4.3.3	Минимальный радиус объекта	24
4.4	Задача оптимизации энергии	24
4.5	Поиск минимального разреза графа	25

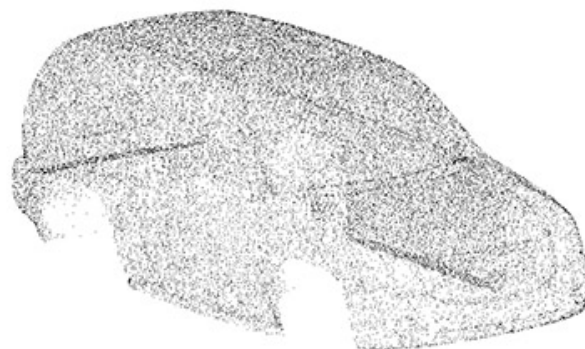
5	Описание реализация	26
6	Результаты	27
6.1	Автоматический режим сегментации	27
6.2	Точность сегментации	27
7	Заключение	33

1 Введение

Задача автоматизированного поиска и распознавания объектов в результатах лазерного сканирования возникла с появлением новых типов приборов — лидаров (от англ. **LIDAR** *Light Identification, Detection and Ranging*), позволяющих получать трехмерное изображение снимаемого объекта. Лидар имеет приемник и источник излучения. Большинство Лидаров работает на частотах 1550нм, 1064нм, а также 532нм и 355нм. Направленный луч источника отраженный от фиксируемого объекта, возвращается к источнику и улавливается приемником — светочувствительным полупроводниковым прибором [5]. Время отклика прямо пропорционально расстоянию до цели. Световые волны подвержены рассеиванию, что позволяет измерять не только расстояние до непрозрачных объектов, но и интенсивность рассеивания в прозрачных средах. Существует два основных способа проведения лазерного сканирования: наземное лазерное сканирование, когда съемка происходит с уровня земли, и воздушное лазерное сканирование, когда изображение получается с летательного аппарата. Результатом лазерного сканирования объектов является *облако точек* — множество точек в трехмерном пространстве, соответствующее точкам на реальном объекте, отраженный сигнал от которых был принят сканером. Если задать систему координат и использовать ее при проведении нескольких лазерных сканирований, то получившиеся облака точек можно преобразовать в одно прямым слиянием множеств. Подобные сканеры применяют-



Фотография лидара.



Пример облака точек

Рис. 1: Лазерное сканирование.

ся при геологоразведке [4], планировании строительных и ремонтных работах [28],

исследовании атмосферы, создание трехмерных моделей снимаемых объектов.

В 1980-х годах появились первые разработки автомобилей-роботов [29], то есть автомобилей, которые управляются без прямого вмешательства человека. В подобных разработках активно используется технология лазерного сканирования. Получаемые данные необходимо обрабатывать в автоматическом режиме. Сканеры закрепляются на автомобилях, а системы обработки производят поиск и распознавание объектов в полученных облаках точек. Кроме информации, получаемой в реальном времени, используются данные о городском ландшафте, полученные заранее — предварительной съемкой и распознаванием всех объектов. Сравнение актуальной информации с полученной ранее, позволяет увеличить точность и скорость принятия решений. В наши дни появляются первые прототипы роботизированных автомобилей. Так совсем недавно в Неваде компания Google получила лицензию на передвижение своего автомобиля-робота по дорогам общего назначения [22]. Автомобиль использует информацию о дорожном покрытии и окружающих объектах, хранящуюся в памяти, а также производит сканирование окружающего мира сканером, расположенным на крыше.



Рис. 2: Автомобиль-робот.

В рамках дипломной работы решается проблема поиска небольших объектов, таких как, автомобили, мусорные баки, дорожные знаки и т.д. в облаках точек, полученных при съемке городских сцен.

2 Постановка задачи

2.1 Неформальное описание задачи

Необходимо реализовать метод, выделяющий объекты в облаках точек, полученных при съемке городского ландшафта. Ставится задача выделять небольшие объекты, такие как: автомобили, дорожные знаки, светофоры. В область интереса не включатся здания и другие крупные объекты.

2.2 Формальная постановка задачи

Была поставлена задача реализовать алгоритм сегментации точек в облаке точек. Необходимо выделять из сцены объект определенного типа (например, светофор, автомобиль и т.д.) и отделять его от фона и других объектов. На вход алгоритму подается неразмеченное облако точек, а также дополнительная информация, полученная методами машинного обучения: для каждой точки облака ее голос — совокупность точек (их координат), а также координаты центра сегментируемого объекта. На выходе каждой точке должна быть сопоставлена метка: объект или фон (другие объекты также считаются фоном).

Алгоритм получает на вход *неразмеченное* облако точек, центр сегментируемого объекта и дополнительную информацию в виде голосов каждой точки за центр объекта (голосом будем называть четверку (x, y, z, v) , где первые три составляющие — координаты точки, за которую отдан голос, а v — вес голоса), и назначает каждой точке в соответствие метку: объект или фон. Таким образом, сигнатуру функции, реализуемую алгоритмом классификации (A) можно описать следующим образом: $A : ((x_1, y_1, z_1, votes_1), \dots, (x_n, y_n, z_n, votes_n), (c_x, c_y, c_z)) \mapsto (k_1, k_2, \dots, k_n), k_j \in \{0, 1\}$ где (x_i, y_i, z_i) — координаты точки в пространстве \mathbb{R}^3 , (c_x, c_y, c_z) — координаты центра объекта, $votes_i$ — множество голосов (координат точек) из точки (x_i, y_i, z_i) , а k — метка класса для объекта, которому принадлежит точка: $\in \{0, 1\}$.

2.3 Критерии качества

Оценивать результаты работы алгоритма стоит сравнением с эталонным результатом (тем, как разметил бы облако точек человек).

Один из методов, применяемый для оценки качества распознавания - *F-оценка*. Метод основывается на подсчет среднего гармонического *точности* и *отклика*. Для всех точек k_i облака проверим гипотезу : k_i является точкой объекта. Тогда в зависимости от решения алгоритма и реального значения гипотезы возможны 4 случая:

	Алгоритм принял	Алгоритм отверг
верна	TP	FN
не верна	FP	TN

Таблица 1: Взаимное отношение решения алгоритма и реального значения гипотезы

Отклик — это отношение количества верно найденных точек к общему количеству точек класса. *Точность* — это отношение количества верно найденных точек к общему количеству точек, найденных алгоритмом. *F-оценка* — это среднее гармоническое между откликом и точностью.

$$\text{Точность} = \frac{TP}{TP + FP}, \text{Отклик} = \frac{TP}{TP + FN}, F\text{-оценка} = \frac{2 \cdot \text{Точность} \cdot \text{Отклик}}{\text{Точность} + \text{Отклик}}$$

Именно этот функционал будем считать оценкой качества алгоритма, и стараться его максимизировать

2.4 Цели работы

В рамках работы поставлены следующие задачи:

- провести обзор методов поиска и распознавания объектов в облаках точек с целью сравнительного анализа существующих методов;

- разработать алгоритм сегментации облаков точек на основе схемы голосования;
- реализовать этот алгоритм и провести его сравнение по F-оценке с существующими аналогами.

3 Обзор существующих методов

Все существующие алгоритмы можно рассматривать как последовательность решения некоторых подзадач: проведение предобработки (унификация данных), поиск начального приближения центра объекта и сегментация.

3.1 Предобработка

На этом этапе облако точек подготавливается к дальнейшей работе. Если имеется несколько сканов одной сцены, то облака точек объединяются. Происходит сглаживание шумов. Также возможно удаление некоторых точек, на основе соображений, связанных с конкретной задачей и введение дополнительных структур данных.

3.1.1 Сглаживание

При получении облака точек съемкой с разных позиций могут получаться неровности. Для создания более сглаженного облака применяется метод Moving Least Squares [32]. Основная идея заключается в построении непрерывной функции $p(x)$, которая будет давать наименьшее квадратичное отклонение от исходных точек, то есть минимизировать следующий функционал:

$$\sum_{i \in I} (p(x) - f_i)^2 \theta(\|x - x_i\|)$$

После нахождения такой функции исходные точки заменяются на ближайшие к ним точки, лежащие на кривой.

3.1.2 Фильтрация шума

Шум на трехмерных сканах может возникать вследствие дефектов съемки. Для избавления от подобного шума из облака удаляются все изолированные точки, то есть такие точки (или небольшие группы, состоящие из нескольких точек и находящиеся на малом расстоянии друг от друга), которые в некотором фиксированном радиусе не имеют соседей — других точек.

3.1.3 Выделение объектов интереса

В задаче поиска небольших объектов на сканах не представляют интереса крупные связанные компоненты (определим понятие связанной компоненты в облаке точек как отношение эквивалентности на множестве точек: будем считать 2 точки принадлежащими одной компоненте, если существует ломаная, соединяющая эти точки, концы отрезков которой являются точками облака, а длина каждого из отрезков не более l : V_0, V_n принадлежат одной связной компоненте, если $\exists V_0, V_1, \dots, V_n : \forall i < n, \rho(V_i, V_{i+1}) < l$).

В статье [6] для избавления от точек, принадлежащим земле удаляются все точки, которые лежат ниже некоторой фиксированной высоты. После этого проводится удаление зданий — удаляются все связанные компоненты, в которых число точек больше некоторого порогового значения.

3.1.4 Введение дополнительных структур данных

Почти все алгоритмы, связанные с сегментацией объектов, используют некоторую структуру, определенную над облаком, например, kd-деревья[30] или октодеревья[31]. Это необходимо для восстановления нормалей к точкам или построения knn-графа (knn-графом называется граф, в котором каждая точка имеет общие ребра только с ближайшими k точками: $\forall v_0 \in V \exists v_1, v_2, \dots, v_k \in V : \forall i \in (1, \dots, k) : (v_0, v_i) \in E$ и $\forall j > k \rho(v_0, v_j) \geq \rho(v_0, v_k), (v_0, v_j) \notin E$).

3.2 Поиск начального приближения центра объекта

Многие алгоритмы сегментации требуют некоторой информации об объекте, который необходимо выделить. Чаще всего, это координаты центра объекта. В обзоре приводятся некоторые методы локализации объектов:

- **Использование карты высот.** В работе [6] предлагается метод локализации объектов на основе карты высот. Карта высот — двухмерное представление облака точек, проекция на горизонтальную плоскость всех точек облака, причем интенсивность спроецированной точки определяется верхней точкой, являющейся образом проекции. Как правило, для упрощения вычислений на

горизонтальной плоскости вводится сетка и проекция на одну ячейку сетки сопоставляется с одним пикселем на двухмерном изображении. Чем выше располагалась самая верхняя точка, тем больше яркость на полученной проекции. Интенсивность I (в интервале от 0 до 255) пикселя, соответствующего клетке (i, j) вычисляется следующим образом:

$$I_{i,j} = \left\lfloor 255 * \frac{\max(z_{i,j})}{z_{\max}} \right\rfloor,$$

где z_{\max} — наивысшая точка во всём облаке.

Исходя из гипотезы о том, что все объекты находятся выше уровня земли, по карте высот можно выделять их местоположение и выбирать центры объектов внутри найденных областей. Стоит заметить, что в городских сценах уровень земли в пределах небольших областей (десятки метров) постоянен, поэтому подобный метод позволяет эффективно выделять объекты на фоне земли.

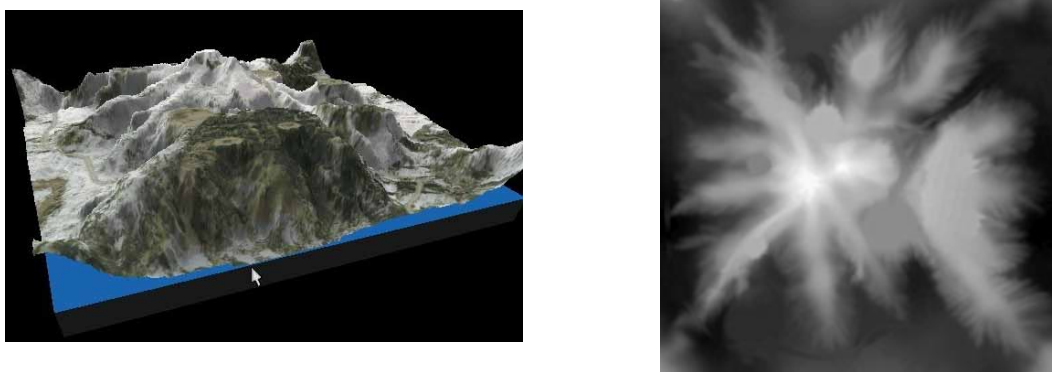


Рис. 3: трехмерная поверхность и карта высот

- **Плотность облака.** Ещё один метод, применимый для локализации объектов — поиск областей, в которых находится наибольшее число точек на единицу объема. Поиск таких областей происходит при помощи итеративного алгоритма Mean Shift [33]. Сначала выбирается точка внутри облака чуть выше уровня земли. На следующем шаге в некотором радиусе ищется область с наибольшей плотностью, центр этой области назначается новым центром. Итерации повторяются, пока не будет найден локальный максимум

плотности. Недостаток алгоритма заключается в том, что он не универсален. В связи с тем, что размеры объектов могут сильно различаться, становится невозможным подбор значения размера области поиска нового центра и единицы объема, которые одинаково хорошо работали бы и с большими, и с маленькими объектами.

- **Центр связанных компонент.** Также возможно искать центры объектов, выделяя связанные компоненты внутри облака, как предложено также в работе [6]. Связанные компоненты можно искать, строя KNN граф, соединяя каждую точку с ближайшими K точками. Центры таких компонент и будут искомыми центрами. Недостатком такого подхода является то, что близко расположенные объекты могут оказаться одной компонентой связанности, что приведет к ошибке в поиске центра объекта.

3.3 Сведение к задаче машинного обучения

Сегментация является частным случаем классификации — одной из задач машинного обучения, посвященной решению следующей проблемы: имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества. Тогда можно сформулировать задачу сегментации как задачу классификации для двух классов.

Существует большое количество различных методов классификации (байесовский классификатор, нейронная сеть, решающие деревья и т.д.) в обзоре приведены методы, наиболее часто использующиеся для решения задачи классификации в облаке точек. Важным является выбор признаков, на основе которых будет производиться классификация на объект и фон.

3.3.1 Вычисление признаков

Методы классификации могут использовать в качестве признаков свойства, посчитанные для целых сегментов (например, внутри некоторой сферы фиксированного радиуса) или для отдельных точек. В случае отдельных точек используют некоторую окрестность, поскольку одна точка не может дать никакой полезной информации. Обычно берутся ближайшие k точек, чтобы признаки не зависели от плотности сканирования. Разные признаки специфичны для конкретных задач, однако есть и универсальные признаки. Рассмотрим приведенные в литературе методы.

Спин-изображения. В статье [11] описываются спин-изображения, которые часто используются для вычисления локальных признаков. По сути, это некоторый инвариант, описывающий поверхности. Для точки O , в которой считается дескриптор и которая принадлежит поверхности, задается система координат (α, β) , где α — расстояние от точки до нормали в точке O , а β — направленное расстояние до касательной плоскости в точке O . Таким образом, все точки на окружности в плоскости, параллельной касательной в точке O будут иметь одну и ту же координату.

Признаки, специфичные для конкретных задач. В задачах поиска объектов в городском ландшафте в [6] используется признак высоты точки над ландшафтом, к примеру, если точка находится выше 3 метров над землей, то она не может принадлежать автомобилю. Также используются признаки взаимного положения объектов, к примеру, автомобили с большой вероятностью находятся на дороге, чем в кустарнике. В работе [16] для анализа городских сцен авторы используют следующие признаки:

- Вокруг точки строится вертикальный цилиндр. Вычисляются распределение точек в различных областях цилиндра (к примеру, между плоскостями $z = z_1$ и $z = z_2$).
- Сравнение высоты точки с порогом.

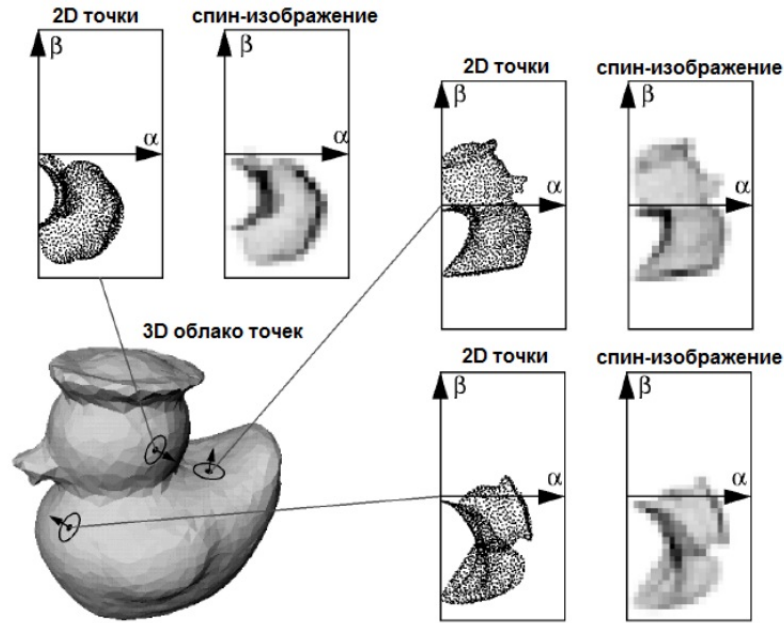


Рис. 4: Иллюстрация к определению спин-изобаржения.

3.3.2 Обучение/классификация

Мультиклассовая логистическая регрессия. Опишем сначала алгоритм для двух классов: пусть есть 2 класса: A и B . Потребуем от классификатора: пусть принадлежность объекта к классу A будет обозначаться единицей, а непринадлежность к классу A (то есть принадлежность к классу B) нулем. На выходе классификатора для каждой точки получается значение p от 0 до 1. Введя некоторый порог t , будем считать, что в случае $p > t$ классификатор отнес точку к классу A . В обратном случае - все наоборот, соответственно. Значение p можно считать вероятностью того, что точка принадлежит классу A . В основе логистической регрессии лежит линейный классификатор. Решающую функцию можно записать в следующем виде: $h = g(\theta^T X)$, где X — признаки, θ — вектор параметров обученного классификатора, а g — сигмоидная функция: $g(z) = \frac{1}{1+e^{-z}}$. Параметры классификатора обучаются из соображений минимизации оценочной функции:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \ln (h(\theta^T x^{(i)})) + (1 - y^{(i)}) \ln (1 - h(\theta^T x^{(i)})))$$

где m — это количество точек. Для построения мультиклассовой логистической регрессии применяется техника "один против всех". Среди множества всех клас-

сов по очереди выбирается по одному и для пары: этот класс и все остальные — строится логистическая регрессия. Среди всех полученных значений выбирается максимальное. К классу, на котором это значение получилось, и относится точка.

SVM. Алгоритм SVM (Support Vector Machine) применим к линейно разделимым данным. Основная идея заключается в поиске такой гиперплоскости, чтобы максимизировать "зазор" между данными из обучающей выборки и гиперплоскостью. Тогда при небольших отклонениях значения признаков на тестовой выборке от среднего значения на обучающей не будет влиять на качество классификации. Если данные линейно неразделимы, то преобразованием пространства признаков данные можно преобразовать в линейно разделимые. К примеру, посредством ядра Гаусса. В пространстве признаков для каждой точки l_i строится признак f_i на основе существующих признаков X . Значение f_i для некоторой точки l_j будет

$$f_i(l_j) = \exp\left(-\frac{\|l_j - l_i\|^2}{2\sigma^2}\right)$$

Тогда, возвращаясь к задаче для двух классов из описания логистической регрессии, значение целевой функции, описывающей принадлежность к классу A , будет $\theta^T F > 0$, где θ — параметры классификатора, а F — вектор f_i

Дерево принятия решений. При применении этого метода используются не сами признаки, а некоторые дискретные функции, зависящие от этих признаков. Этот метод имеет смысл применять в случае, когда каждому классу присущи элементы с конечным набором дискретных признаков. Например, для поиска фонарных столбов можно использовать признаки: отношение высоты к ширине больше 20 и высота более 4 метров. Суть метода заключается в построении наипростейшего алгоритма принятия решения, к какому классу отнести объект, на основе этих дискретных признаков. В дереве принятия решений на ребрах записываются атрибуты, от которых зависит целевая функция, а в листьях расположены метки класса. При построении дерева основная задача — выбор признака в каждом узле. Один из методов — выбор на каждом этапе признака, который дает наибольшую информацию, к примеру, с использованием алгоритма ID3 (Iterative Dichotomiser 3) [17].

Случайный лес. Алгоритм случайный лес является развитием идеи дерева принятия решений. Случайный лес представляет из себя ансамбль решающих деревьев. Построение происходит следующим образом.

Пусть количество признаков M , размер выборки N . Каждое дерево ансамбля строится независимо. Генерируется случайным образом подвыборка размера M , но с возможным повторением элементов основной выборки, также выбирается n признаков ($n \ll N$). На основе этих данных строятся решающие деревья. Классификация происходит на основе схемы голосования. Каждое дерево голосует за свой класс. Побеждает класс, получивший наибольшее число голосов.

Число деревьев выбирается из соображений минимизации ошибки на тестовой выборке.

3.4 Сегментация на основе минимального разреза графа

В статье [6] предложен метод сегментации, использующий поиск минимального разреза графа.

Сначала в облаке точек производится предобработка: удаляются отдельные изолированные точки, большие объекты (здания), вырезается земля. Затем производится поиск центров объектов. Предложено несколько реализаций:

- Поиск локальных максимумов на карте высот.
- Поиск областей в облаке точек с наибольшей плотностью методом Mean Shift.
- Поиск центров областей связанности.
- Построение k nn графа, задания энергии на ребрах, как гауссиан от длины ребра и проведения в графе поиска минимального разреза.

После нахождения центра сегментируемого объекта на точках облака строится граф. На этом графе задается энергия: унарный потенциал (энергия в вершине) задается как расстояние до центра объекта, а парный потенциал (энергия на ребрах) — как гауссиан от длины ребра. При вычислении потенциалов используется информация, заданная пользователем — априорный радиус объекта. Поскольку

размеры объектов могут сильно варьироваться, то требование ручной подборки оптимального радиуса является серьезным недостатком этого алгоритма.

Задача оптимизации энергии сводится к задаче поиска минимального разреза графа.

3.5 Неявные модели формы

Другой алгоритм, использующий оптимизацию энергии — алгоритм Неявные модели формы (Implicit Shape Model, дальше — ISM) [2]. До проведения сегментации проводится предварительный поиск центров объектов в облаке точек, а также вычисляется дополнительная информация, используемая для задания энергии на графе.

Алгоритм осуществляет поиск центров объектов в облаке, а также метки классов для этих объектов. На вход алгоритму подается неразмеченное облако точек, на выходе получается распределение вероятности расположения центра объекта. Для правильной работы алгоритм должен быть обучен на размеченной выборке.

При обучении на вход подается размеченное облако точек. Выбираются характерные точки. Для некоторой окрестности этих точек строятся пространственные дескрипторы 3D SURF [2]. Каждому дескриптору ставится в соответствие метка класса объекта, а также расположение центра объекта. Такое соответствие возможно, поскольку дескриптор SURF устойчив к повороту и масштабированию. Полученные дескрипторы разделяются на классы. Количество классов выбирается в соотношении 1 к 10 от общего количества дескрипторов. Построение классов происходит с помощью алгоритма K-средних.

Алгоритм K-средних является алгоритмом кластеризации, то есть разделяет данные на классы, исходя из схожести свойств данных (алгоритм машинного обучения — обучение без учителя). Метод является итерационным. На первом этапе случайным образом выбираются центры кластеров. Каждый элемент выборки, для которой происходит кластеризация, относится к кластеру, расстояние до центра которого минимально. Затем центры кластеров переносятся в центры множеств элементов, отнесенных к соответствующему кластеру на текущей итерации. Далее алгоритм повторяется до тех пор, пока кластеры не прекратят изменяться.

После разбиения множества дескрипторов в нашей задаче на отдельные кластеры каждый кластер будет иметь сразу несколько меток классов объектов и направлений на центр объекта.

На этапе поиска вероятностного распределения также выбираются точки, лежащие на краю объектов. Для них строятся дескрипторы. Каждый дескриптор относится к одному из кластеров, полученных при обучении. Таким образом, каждая из выбранных точек будет "голосовать" за центр объекта и его метку класса. Из совокупности точек получается вероятностное пространство для центров объектов и меток класса для них. Центр области наибольшей плотностью голосов объявляется центром объекта.

Для выделения объектов из облака точек применяется сегментация оптимизацией энергии в графе.

Унарный потенциал зависит от среднего расстояния до ρ ближайших правильно проголосовавших за центр объекта точек, а также от того, голосовала ли сама точка правильно за центр объекта. Парный потенциал задается исходя из 3 соображений:

- размытие: штраф за соотнесение соседних точек (точек, с общим ребром в графе) к различным классам (объект и фон);
- соседние точки должны иметь схожие нормали: штрафует за соотнесение к разным классам соседних точек с близкими значениями угла нормали;
- значение дескриптора: штрафует за соотнесение соседних точек с близким (в пространстве дескриптора SURF) значением дескриптора к разным классам;

В статье [1], как пишут авторы, основанный на ISM алгоритм показал хорошие результаты для искусственных данных (сгенерированных моделях людей и животных). К сожалению, в статье не приводится численная оценка работы алгоритма, а лишь примеры работы для различных данных. Тем не менее, оценка авторов статьи дает основания полагать, что основанный на голосовании алгоритм может быть успешно применен для поиска и распознавания объектов в трехмерных облаках, полученных при съемке городского ландшафта.

4 Предлагаемый метод

В этом разделе подробно описываются предлагаемые алгоритмы. Общая схема выглядит следующим образом: на облаке точек строится граф, вершинами которого являются точки облака. На этом графе вводится функция энергии. Энергия состоит из двух составляющих: унарный и парный потенциалы. Затем необходимо найти такой подграф этого графа, на котором эта энергия станет минимальной (подразумевается, что вершины, входящие в этот подграф, помечаются единицей, а не входящие — нулем).

4.1 Построение графа

Стандартный подход к построению графа — соединять ребрами все точки с их ближайшими k соседями, k обычно выбирают от 3 до 5 [6]. Такой подход позволяет сохранять примерно постоянную степень вершин в графе вне зависимости от плотности сканирования. В данной работе использовалось $k = 4$ и $k = 5$.

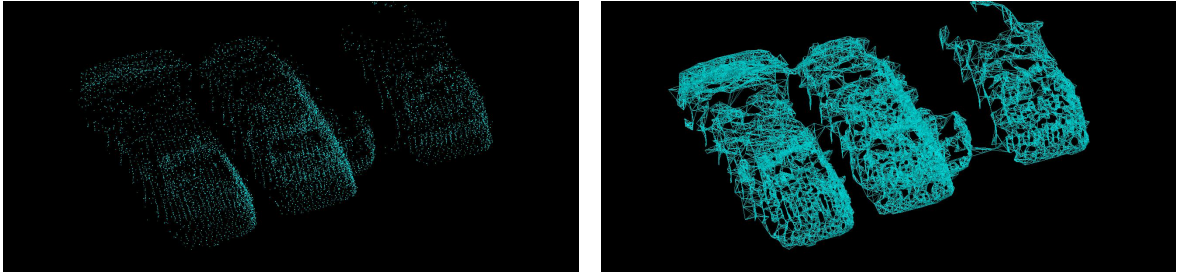


Рис. 5: облако точек и knn граф ($k=5$)

4.2 Схема голосования

Для получения на входе алгоритма таких данных, как голоса точек и центр сегментируемого объекта, необходимо предварительно их получить на основе обрабатываемого облака. Для этого применяется метод Implied Shape Model (ISM) [1].

Метод ISM является методом машинного обучения и его можно разделить на два этапа: обучение и распознавание. На входе у метода — неразмеченное облако точек, на выходе — центр объекта и метка класса, к которому относится объект.

При обучении на входе имеется облако точек с отмеченным центром объекта. Для каждой точки облака вычисляется дескриптор SURF [12], который инвариантен к повороту и масштабированию. Имея информацию об ориентации каждой точки и координаты центра объекта, можно получить вектор $(\alpha, \beta, \gamma, d)$, где (α, β, γ) — направление на центр объекта, а d — величина, равная размеру окрестности точки, на которой вычисляется дескриптор, в дальнейшем будем называть ее *масштаб*. Получается множество четырехмерных векторов.

Затем в пространстве дескрипторов строится ”мешок слов”: получившиеся вектора разделяются на кластеры, для этого применяется метод К-средних [18]. Количество кластеров выбирается в количестве 10% от общего числа векторов [2]. Полученное разбиение и называется мешком слов.

Таким образом, после обучения получается набор центров кластеров, с каждым из которых можно сопоставить один или более ”голосов” (то есть голоса всех тех точек, дескрипторы которых относятся к этому кластеру). Голос — это вектор (x, y, z, w) , где (x, y, z) — точка, за которую голосуют (направление на возможный центр объекта), а w — величина голоса. Величина голоса определяется из статистических соображений и является произведением двух составляющих:

- статистический вес W_{st} , который определяет размер голоса за определенный класс
- обученный вес W_{lrn} , который нормализует величину голоса в зависимости от того, как часто каждый из кластеров верно голосует за центр объекта на обучающей выборке

Статистический вес вычисляется на обучающей выборке следующим образом:

$$W_{st}(c_i, v_j) = \frac{1}{n_{vw}(c_i)} \cdot \frac{1}{n_{vot}(v_j)} \cdot \frac{\frac{n_{vot}(c_i, v_j)}{n_{ftr}(c_i)}}{\sum_{c_k \in C} \frac{n_{vot}(c_k, v_j)}{n_{ftr}(c_k)}}$$

где $n_{vot}(v_j)$ — общее количество голосов из кластера v_j , $n_{vot}(c_i, v_j)$ — число голосов за класс c_i из кластера v_j , $n_{vw}(c_i)$ — число кластеров, голосующих за класс

c_i , а $n_{ftr}(c_i)$ — число различных вычисленных дескрипторов, на которых обучался класс c_i . C — множество всех классов объектов. Первый множитель делает каждый класс инвариантным числу кластеров, второй — нормализует число голосов, порожденное каждым кластером, и, наконец, третий множитель отражает вероятность того, что кластер v_j проголосовал за класс c_i .

Для вычисления обученного веса вводится величина λ_{ij} — один голос, случайно выбранный из множества голосов, порожденных кластером v_i за класс c_i . Затем этим голосом заменяется вхождение всех голосов за c_i в кластере v_i (для тех точек, на которых эти голоса были получены), и вычисляется расстояние уже от нового голоса до центра объекта.

Обозначим множество различных голосов в v_i за c_i как A , расстояние от нового голоса до центра как $d_a(\lambda_{ij})$, $a \in A$. Тогда обученный вес W_{lrn} вычисляется как медианное значение среди значений функций Гаусса для различных $a \in A$:

$$W_{lrn}(\lambda_{ij}) = f \left(\left\{ \exp \left(-\frac{d_a(\lambda_{ij})^2}{\sigma^2} \right) \middle| a \in A \right\} \right)$$

где σ — нормальное отклонение (10% от размера окрестности точки, на которой вычисляется дескриптор), а f — функция медианы [2].

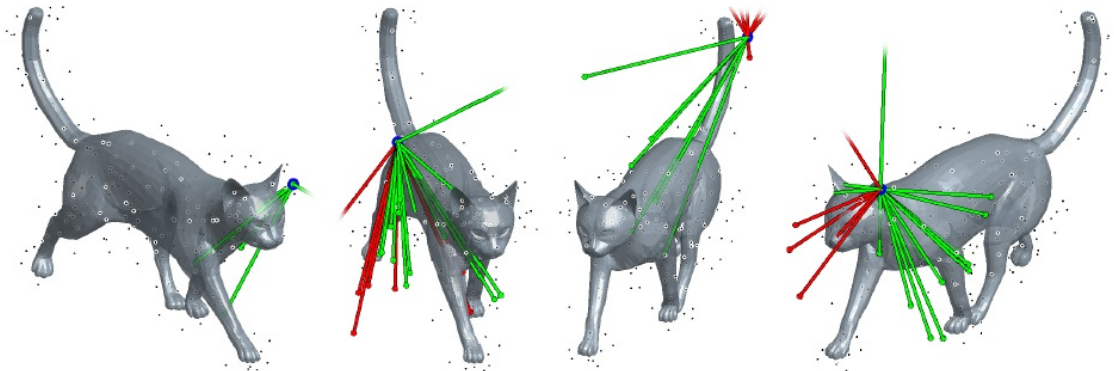


Рис. 6: Иллюстрация к описанию алгоритма ISM. Одна точка может породить несколько голосов за потенциальные центры разных классов.

При распознавании на входе имеется неразмеченное облако точек. Для каждой точки вычисляется дескриптор SURF. После вычисления дескриптора точки,

она относится к ближайшему из кластеров (по расстоянию до центра кластера в пространстве дескрипторов), полученных на этапе обучения. Каждая точка порождает все те голоса, которые имеет кластер, к которому она отнесена.

Центр объекта вычисляется следующим образом: считается, что каждый голос порождает область с потенциалом, равным величине голоса и некоторым фиксированным радиусом. После вычисления всех таких областей и сложения потенциалов в каждой точке производится поиск наибольшего потенциала. Центр области с наибольшим потенциалом объявляется центром объекта.

Для применения алгоритма сегментации необходимы координаты центра объекта, а также голоса, порожденные каждой из точек (те голоса, которые получаются на этапе распознавания алгоритма ISM).

4.3 Задание энергии

В общем виде энергия Q , заданная на графе унарными и парными потенциалами точек, выглядит следующим образом:

$$Q = \sum_{i=1}^{N_v} \left(\phi(v_i; f_i, \theta) + \sum_{j|(i,j) \in E} (\Lambda_\psi \cdot \psi(f_i, f_j, \theta)) \right)$$

где θ — вектор параметров алгоритма ($\theta = (\Lambda_\psi, \alpha, \beta, \kappa, \rho, \sigma)$), f_i — метка точки v_i , ϕ — это унарный потенциал каждой точки, а ψ — парный потенциал для точек, имеющих общее ребро.

Рассмотрим подробнее каждую составляющую энергии.

4.3.1 Унарный потенциал

В предложенном методе задание унарного потенциала происходит аналогично, описанному в статье [1].

Так как центр объекта восстановлен с использованием схемы голосования, то точки, которые голосовали за найденный центр, имеет смысл считать точками объекта. Также дополнительно имеет смысл поощрять точки, имеющие правильно проголосовавших соседей, увеличением энергии за не отнесение точки к объекту на средневзвешенное расстояние до ρ ближайших соседей (в статье [1] рекомендуется

выбирать ρ равное 10).

$$\begin{aligned}\phi(v_i; f_i, \theta) &= \delta_{f_i,0}(\varphi(v_i; f_i, \theta)) + \delta_{f_i,1}(\kappa - \varphi(v_i; f_i, \theta)), \\ \varphi(v_i; f_i, \theta) &= \frac{1}{\rho} \sum_{j=1}^{\rho} \exp\left(\frac{-\|v_i - b_i^j\|^2}{\alpha^2}\right) + f_c(v_i),\end{aligned}$$

где b_i^j — j -ая ближайшая точка, проголосовавшая за центр. $f_c(v_i)$ равно 1, если точка v_i голосовала за центр и 0 в противном случае.

4.3.2 Парный потенциал

Парный потенциал задается, исходя из двух соображений: длины ребра и сглаживания. С одной стороны, чем длиннее ребро, тем больший штраф на него накладывается, с другой стороны, точки, имеющие общее ребро штрафуются за разметку различными метками.

$$\psi(f_i, f_j, \theta) = \psi_{smooth}(f_i, f_j, \theta) + \psi_{length}(f_i, f_j, \theta)$$

Штраф за сглаживание выглядит следующим образом:

$$\psi_{smooth}(f_i, f_j, \theta) = (1 - \delta_{i,j})$$

Потенциал, зависящий от длины ребра:

$$\psi_{length}(f_i, f_j, \theta) = \left(\exp\left(\frac{-\|f_i - f_j\|^2}{\sigma^2}\right)\right)$$

4.3.3 Минимальный радиус объекта

Кроме расстановки потенциалов производится дополнительное предположение, что все точки в некоторой окрестности β центра будут принадлежать объекту. Это мотивировано тем, что существует некоторый минимальный размер объекта, поэтому все точки внутри этой области гарантированно должны относиться к объекту.

4.4 Задача оптимизации энергии

После задания энергии необходимо выбрать такой подграф, на котором энергия, заданная указанным способом, будет минимальной. Считается что вершины,

входящие в подграф, имеют метку 1, а не входящие — метку 0. Причем в подграфе V' графа V должно выполняться условие: если $v_i, v_j \in V, (v_i, v_j) \in E$ и $v_i, v_j \in V'$, то $(v_i, v_j) \in E'$.

4.5 Поиск минимального разреза графа

Как показано в [24] задачу оптимизации энергии, можно свести к задаче поиска минимального разреза графа. Задача поиска минимального разреза графа формулируется следующим образом:

Пусть $G = \langle V, E \rangle$ — взвешенный граф с двумя выделенными вершинами, назовем их терминальными. Разрезом графа $C \subset E$ называется множество ребер графа G , такое, что при удалении этих ребер из графа G , терминальные вершины оказываются в разных компонентах связности, и ни для какого подмножества C это свойство не выполняется. Ценой разреза называется сумма веса ребер, входящих в разрез. Задача поиска минимального разреза заключается в поиске разреза графа с минимальной ценой. Именно в такой классической формулировке ставится задача оптимизации¹.

¹В некоторых публикациях задача поиска минимального разреза формулируется по-другому. Так, например, в [19] вычисляется *глобальный минимальный разрез*. Ищется минимальный по стоимости разрез графа из всех разрезов, разделяющих граф на две непустые части. В [20] используется *нормализованный разрез*, в котором другим способом определяется значение стоимости.

5 Описание реализация

Алгоритм был разработан на базе существующей реализации алгоритма ISM[2]. На вход реализованный алгоритм получает выход алгоритма ISM (координаты центра объекта и метку класса, которая определяет параметры при задании энергии), а также данные, используемые при работе ISM (голоса точек).

Реализация была выполнена на языке C++. Выбор языка программирования был обусловлен высокой скоростью работы и существующими библиотеками для работы со структурами данных, описывающими трехмерные облака точек. Библиотека Point Cloud Library [3] предоставляет структуры для работы с облаками точек и методы их преобразований. Для реализации поиска минимального разреза графа использовалась библиотека gco-v3.0 [24].

В статьях *Scene cut: Class-specific object detection and segmentation in 3D scenes* [1] и *Min-cut based segmentation of point clouds* [7] описаны схемы проведения сегментации объектов в облаках точек. В этих алгоритмах используется схожая техника проведения сегментации (поиск минимального разреза графа), поэтому является интересным сравнение результатов работы разработанного алгоритма и алгоритмов в статьях. Однако результаты, приведенные в статьях, сильно зависят от данных, к которым применяются алгоритмы. В связи с этим было принято решение также реализовать и эти методы. Алгоритмы были реализованы, сравнение результатов работы предложенного метода и методов, описанных в статьях, изложены в главе 6 "Результаты".

Реализация включает в себя:

- модуль построения графа по облаку точек;
- реализация подсчета унарных потенциалов;
- реализация подсчета парных потенциалов;
- реализация поиска минимального разреза графа с использованием библиотеки gco-v3.0;
- модуль оценки качества работы алгоритма;

6 Результаты

Качество работы разработанного алгоритма оценивалось на данных, снятых лидаром в Оттаве, Канада [21]. Алгоритм применялся к классу объектов "Автомобиль". Данные проходят предобработку, после чего они поступают на вход ISM. Выход ISM поступает на вход разработанному алгоритму.

Сравнение результатов производилось с алгоритмами, описанными в статьях [1] и [7], которые также были реализованы. Контрольная выборка была получена ручной разметкой.

6.1 Автоматический режим сегментации

Метод, предложенный в статье Головинского [7], реализует сегментацию с использованием дополнительной информации, получаемой от пользователя. Для проведения успешной сегментации требуется задать радиус сегментируемого объекта (см. Рис. 7). Так как класс сегментируемого объекта в алгоритме не учитывается, то выбор оптимального радиуса представляется задачей нетривиальной. Кроме того, внутрикласовая изменчивость может приводить к тому, что даже в пределах одного класса объектов значения радиуса будет необходимо выбирать разные.

Разработанный же алгоритм не требует дополнительных данных от пользователя, что является его важным преимуществом.

6.2 Точность сегментации

Было поставлено несколько экспериментов для сравнения предложенного метода с существующими подходами. Ниже приводится таблица, описывающая результаты сравнения.

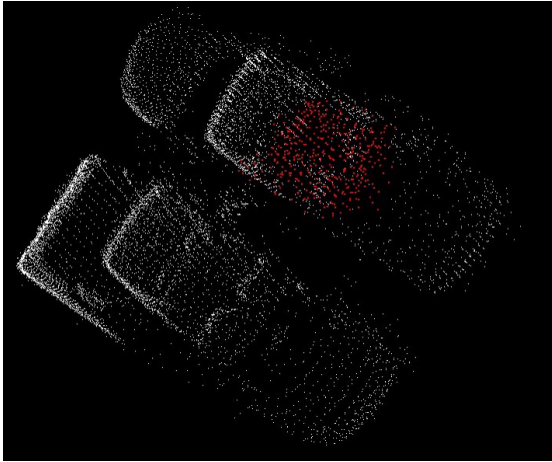
В первой колонке представлены результаты работы алгоритма из статьи Головинского [7] с вручную подобранным значением радиуса. Во второй колонке представлены результаты работы алгоритма из статьи Кноппа. И, наконец, в третьей колонке — результаты предложенного метода.

	Головинский [7]	Кноп [1]	Предложенный метод
Отклик	97.26%	92.51%	97.14%
Точность	99.18%	99.89%	99.65%
F-меры	98.1%	95.73%	98.27%

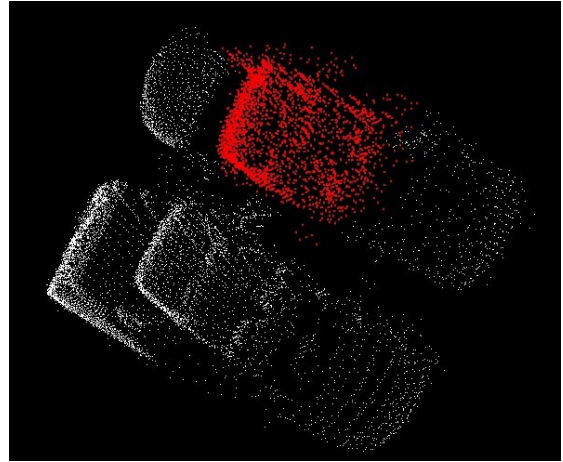
Таблица 2: Результаты работы алгоритмов на тестовых данных. Выбраны средние значения отклика, точности и F-меры среди все проведенных экспериментов.

Как видно из таблицы предложенный алгоритм повторяет результаты алгоритма из статьи Головинского, но не требует дополнительных данных от пользователя.

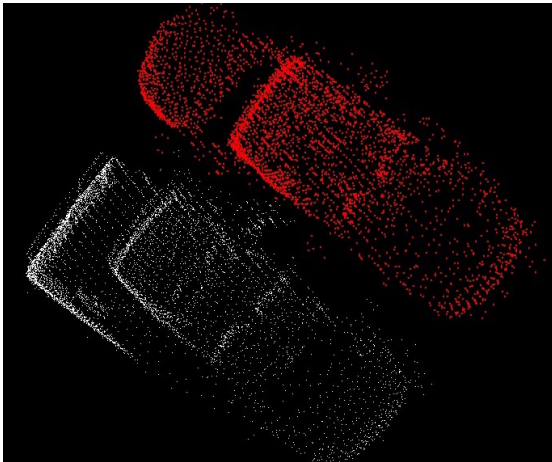
Сравнение с алгоритмом, описанным в статье Кнопа, показывает, что разработанный алгоритм имеет лучший отклик (то есть имеет меньшую ошибку второго рода). Ниже приводится иллюстрация работы алгоритмов.



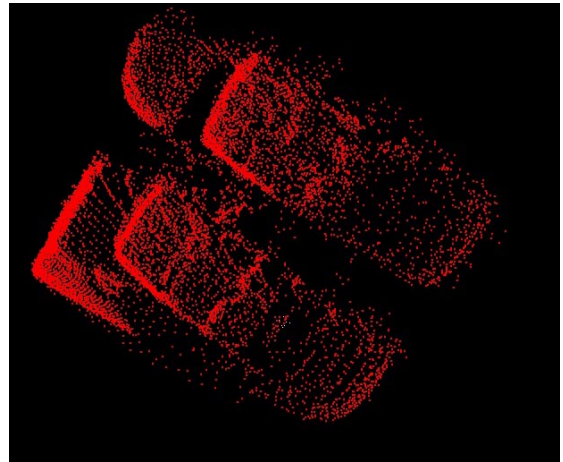
Радиус 1 метр



Радиус 2 метра



Радиус 4 метра



Радиус 7 метров

Рис. 7: Результат применения алгоритма из статьи Головинского для разных значений радиуса объекта.

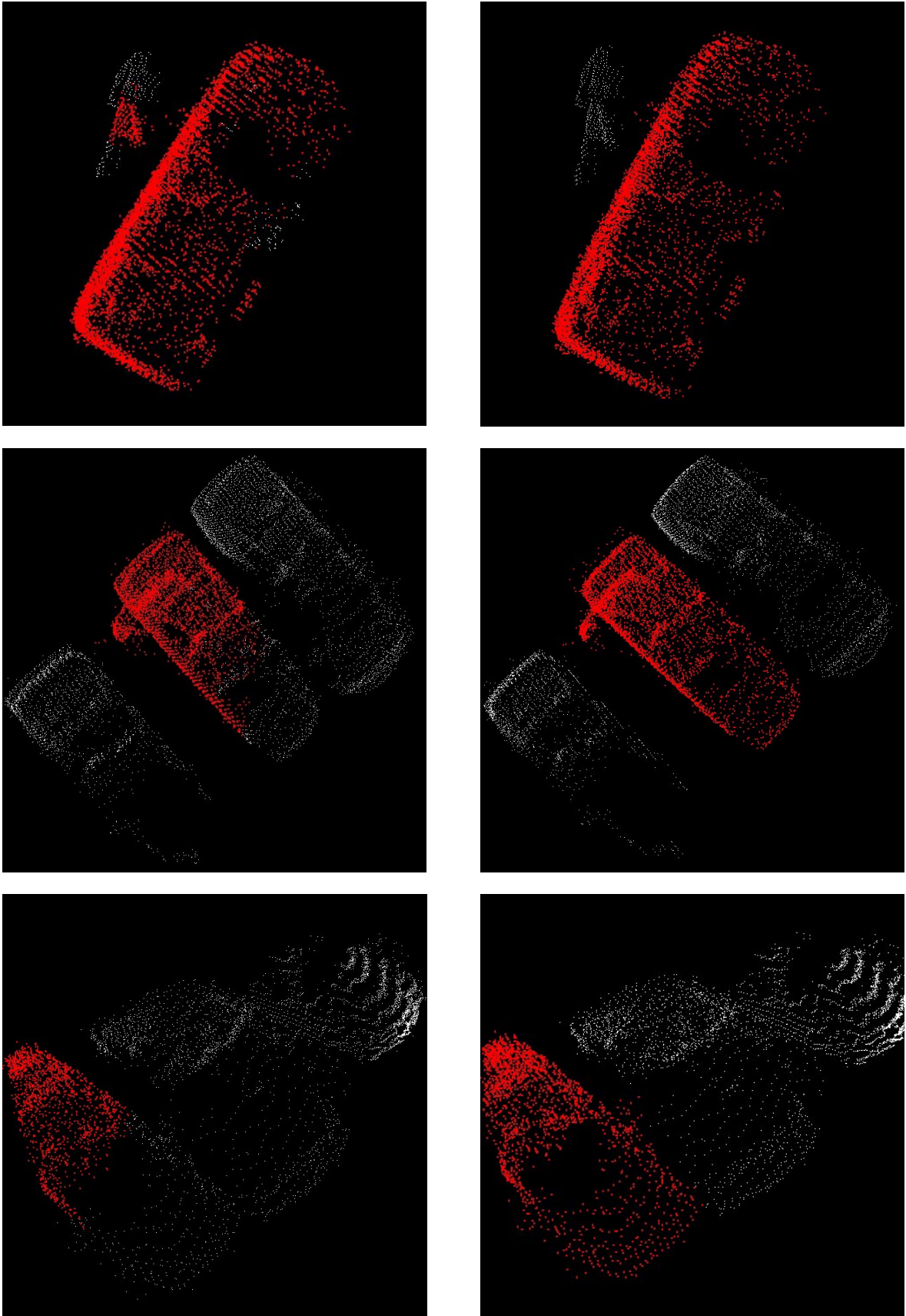


Рис. 8: Результат применения алгоритма из статьи Кноппа (слева) и разработанного алгоритма (справа).

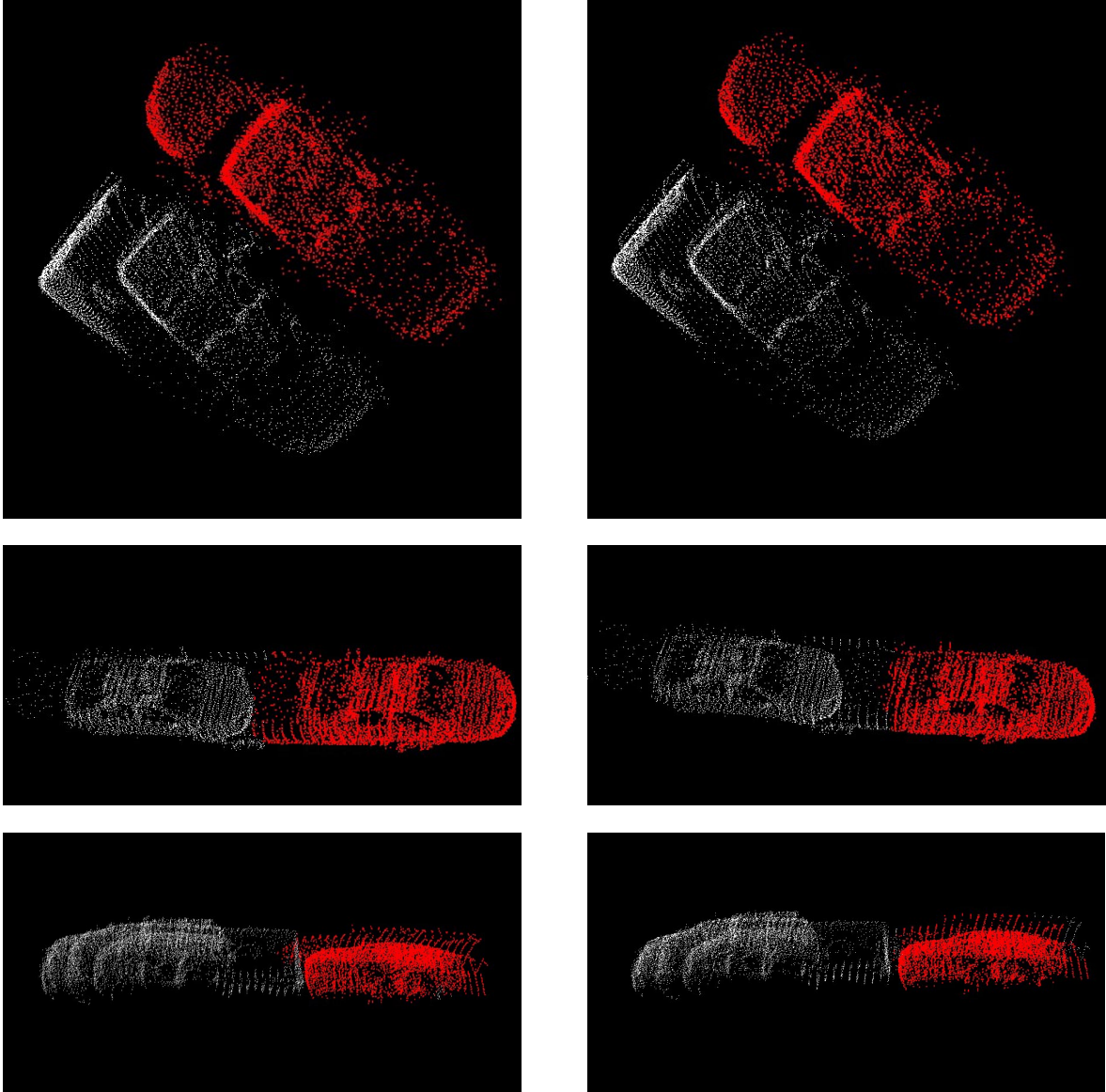


Рис. 9: Результат применения алгоритма из статьи Головинского (слева) с заданным вручную радиусом и разработанного алгоритма (справа).

Разработанный алгоритм превосходит результаты работы существующего автоматического метода и повторяет результат полуавтоматического метода, использующего дополнительную информацию от пользователя.

7 Заключение

В рамках дипломной работы были решены следующие задачи:

- проведен обзор методов сегментации объектов в трехмерных сканах;
- разработан метод сегментации объектов в трехмерных сканах на основе метода ISM с использованием схемы голосования и показано его превосходство над существующими методами;
- на основе разработанного метода реализована система сегментации объектов в трехмерных лазерных сканах.

Использование схемы голосования предоставляет дополнительную информацию о взаимном расположении точек объекта. Использование этой информации для задания энергии на облаке точек с последующей ее минимизацией позволяет получать более точные результаты при сегментации объектов.

Разработанный метод показывает результаты, сопоставимые с результатами полуавтоматического метода, но, в отличие от него, не требует ручного ввода дополнительных данных от пользователя.

Дальнейшее развитие темы данной дипломной работы возможно в следующих направлениях:

- поиск лучшего функционала энергии в графе для сегментации объектов в зашумленных данных;
- улучшение математической или программной части алгоритма для сокращения времени работы (в текущей реализации алгоритм работает 25 секунд на облаке из 15 тысяч точек);

Список литературы

- [1] J. Knopp, M. Prasad, and L. Van Gool. Scene cut: Class-specific object detection and segmentation in 3D scenes. // 3DIMPVT 2011. P. 180-187
- [2] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, Hough transform and 3D SURF for robust three dimensional classification. // ECCV (6) 2010: P. 589-602
- [3] Point Cloud Library. [HTTP] <http://pointclouds.org/>
- [4] Середович А. В. «Методика создания цифровых моделей объектов нефтегазовых предприятий средствами наземного лазерного сканирования»
Новосибирск, 2007 165 с. РГБ ОД, 61:07-5/3352
- [5] Инфракрасные лидары для экологического мониторинга атмосферы :
Учебное пособие для вузов / Б. И. Васильев, У. М. Маннун ; Министерство образования и науки Российской Федерации, Московский физико-технический институт (государственный университет). - М. : МФТИ, 2005
- [6] A. Golovinskiy.; VG. Kim and T. Funkhouser: Shape-based recognition of 3d point clouds in urban environments // In Proc. ICCV, 2009, P.2154-2161.
- [7] A. Golovinskiy and T. Funkhouser: Min-cut based segmentation of point clouds. // ICCV, 2009, P. 39-46
- [8] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert. Scale selection for classification of point sampled 3-d surfaces. // Tech. report CMU-RI-TR-05-01, Robotics Institute, Carnegie Mellon University, January, 2005
- [9] J. Chen and B. Chen. Architectural modeling from sparsely scanned range data. // IJCV, 2008, P. 223-236
- [10] H. Xu, N. Gossett, and B. Chen. Knowledge and heuristic based modeling of laser-scanned trees. // ACM Trans. Graph., 26(4):19, 2007.

- [11] Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes.// IEEE PAMI (1999) vol.21, no.5. — P.433-449
- [12] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). // In Proceedings of Computer Vision and Image Understanding. 2008, P. 346-359
- [13] F. Endres, C. Plagemann, C. Stachniss, W. Burgard. Unsupervised Discovery of Object Classes from Range Data using Latent Dirichlet Allocation. Robotics: Science and Systems. // Robotics: Science and Systems. — Seattle, USA: 2009
- [14] R. Shapovalov, A. Velizhev, O. Barinova. "Non-associative Markov networks for 3D point cloud classification."Photogrammetric Computer Vision and Image Analysis // PCV 2010. IntArchPhRS, vol.38(A), P. 103-108, Paris, 2010.
- [15] D. Munoz, J. Bagnell, N. Vandapel, M. Hebert. Contextual classification with functional Max-Margin Markov Networks // IEEE Conference on Computer Vision and Pattern Recognition. — IEEE, 2009, P. 975-982.
- [16] D. Anguelov, B. Taskar, V. Chatalbashev. Discriminative Learning of Markov Random Fields for Segmentation of 3D // IEEE Conference on Computer Vision and Pattern Recognition, 2005, P. 169-176. [PDF] <http://www.seas.upenn.edu/~taskar/pubs/cvpr05.pdf>
- [17] Andrew Colin, "Building Decision Trees with the ID3 Algorithm Dr. Dobbs Journal, ИЮНЬ 1996
- [18] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations // Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, P. 281-297
- [19] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11):1101–1113, November 1993.
- [20] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, August 2000.

- [21] Neptec. Wright state 100, 2009. www.daytaohio.com/WrightState100.php.
- [22] Google's Driverless Cars Now Officially Licensed in Nevada // [HTTP] <http://techland.time.com/2012/05/08/googles-driverless-cars-now-officially-licensed-in-nevada/>
- [23] Zoltan Csaba Marton, Radu Bogdan Rusu, Michael Beetz. On Fast Surface Reconstruction Methods for Large and Noisy Datasets // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2009.
- [24] Y. Boykov, O. Veksler, R.Zabih. Efficient Approximate Energy Minimization via Graph Cuts // IEEE TPAMI, 20(12):1222-1239, Nov 2001.
- [25] V. Kolmogorov, R.Zabih. What Energy Functions can be Minimized via Graph Cuts? // IEEE TPAMI, 26(2):147-159, Feb 2004.
- [26] Y. Boykov, V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision // IEEE TPAMI, 26(9):1124-1137, Sep 2004.
- [27] A. Delong, A. Osokin, H. N. Isack, Y. Boykov. Fast Approximate Energy Minimization with Label Costs // CVPR, June 2010.
- [28] Послеаварийное лазерное сканирование на Саяно-Шушенской ГЭС [HTTP] <http://www.navgeocom.ru/projects/671/4565/>
- [29] When Will You Have a Car That Drives Itself? [HTTP] <http://www.good.is/post/when-will-you-have-a-car-that-drives-itself/>
- [30] J. L. Bentley, Multidimensional binary search trees used for associative searching // Communications of the ACM., 1975., Vol. 18, no. 9. [HTTP] <http://portal.acm.org/citation.cfm?id=361007>.
- [31] D.R. Reddy and S. Rubin, Representation of three-dimensional objects, CMU-CS.-78-I13, ComputerScienceDepartment, Carnegie-MellonUniversity, Pittsburgh, April 1978.

- [32] P. Breitkopf, H. Naceur, A. Rassineux, P. Villon, Moving least squares response surface approximation: Formulation and metal forming applications // Computers Structures (2005), 83(17-18):1411-1428
- [33] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition // Information Theory, IEEE Transactions, 21(1):32-40, 1975.