



Московский Государственный Университет имени М.В.Ломоносова  
Факультет Вычислительной Математики и Кибернетики  
Кафедра Автоматизации Систем Вычислительных Комплексов  
Лаборатория Компьютерной Графики и Мультимедиа

Дипломная работа

**Автоматическое выделение объектов  
в данных лазерного сканирования**

Роман Викторович Шаповалов, гр.522

Научный руководитель: к.ф.-м.н., н.с. Антон Сергеевич Конушин

Москва, 2010

## **Аннотация**

Автоматическое выделение объектов  
в данных лазерного сканирования

*Роман Викторович Шаповалов*

В этой работе мы улучшили то, что нельзя было улучшить.

## **Abstract**

Automated object detection  
in laserscanning data

*Roman V. Shapovalov*

In this paper we revisit main principles and the technological basis of the Rooter algorithm. We report significant growth of both efficiency and effectiveness.

## Благодарности

I would like to thank the Academy...

# Оглавление

<b>1</b>	<b>Введение</b>	<b>3</b>
1.1	Распознавание объектов в компьютерном зрении . . . . .	3
1.2	Технология лазерного сканирования . . . . .	5
<b>2</b>	<b>Постановка задачи</b>	<b>8</b>
2.1	Формальная постановка задачи . . . . .	8
2.2	Критерии качества . . . . .	9
2.3	Цели работы . . . . .	10
<b>3</b>	<b>Обзор существующих методов</b>	<b>11</b>
3.1	Предобработка . . . . .	11
3.2	Признаки облаков точек . . . . .	12
3.2.1	Спин-изображения . . . . .	12
3.2.2	Признаки матрицы ковариаций . . . . .	14
3.2.3	Признаки, специфичные для предметной области . . . . .	15
3.3	Методы классификации . . . . .	17
3.3.1	Независимая классификация точек . . . . .	17
3.3.2	Совместная классификация точек . . . . .	18
<b>4</b>	<b>Предлагаемый метод</b>	<b>21</b>
4.1	Пространственный индекс и пересегментация . . . . .	21
4.2	Построение графа . . . . .	23
4.3	Признаки и потенциалы . . . . .	24
4.3.1	Унарные потенциалы . . . . .	24
4.3.2	Парные потенциалы . . . . .	25

Оглавление	2
4.4 Вывод в Марковской сети . . . . .	25
<b>5 Экспериментальные результаты</b>	<b>26</b>
5.1 Ошибка пересегментации . . . . .	27
5.2 Точность классификации . . . . .	28
5.3 Эффективность . . . . .	29
<b>6 Заключение</b>	<b>31</b>
<b>Литература</b>	<b>35</b>

# Глава 1

## Введение

### 1.1 Распознавание объектов в компьютерном зрении

Задача выделения объектов в данных, получаемых с помощью различного рода сенсоров, возникает очень часто. Например, в медицинских исследованиях требуется выделить определенные органы, опухоли или ткани организма на рентгеновских снимках или результатах томографии (рис. 1.1,а). В процессе биологических или фармакологических исследований часто возникают задачи, в которых требуется выделить на изображении, полученном с микроскопа, отдельные клетки или их части (рис. 1.1,в). В дефектоскопии тоже возникают подобные задачи, например, автоматизированный анализ качества продукции на конвейере или выделение дефектов дорожного покрытия [10]. Ручное выделение объектов требует кропотливой и длительной работы эксперта, и поэтому эти задачи обычно автоматизируют с использованием методов компьютерного зрения.

С начала 90-х годов XX века учёные стали активно работать над проблемой анализа данных лазерного сканирования, которые представляют собой так называемые облака точек, хотя первые работы на эту тему появились ещё в 80-х годах [15]. Интуитивно, анализ трёхмерных данных выглядит более простой задачей, чем анализ изображений, поскольку последний фактически представляет собой обратную задачу: при получении фотографии значительная часть информации о пространственной структуре сцены теряется. Высокий потенциал трёхмерного представления данных отмечал ещё один из пионеров компьютерного зрения David Marr, который считал, что даже распознавание объектов на двумерных изображениях должно выполняться посредством восстановле-

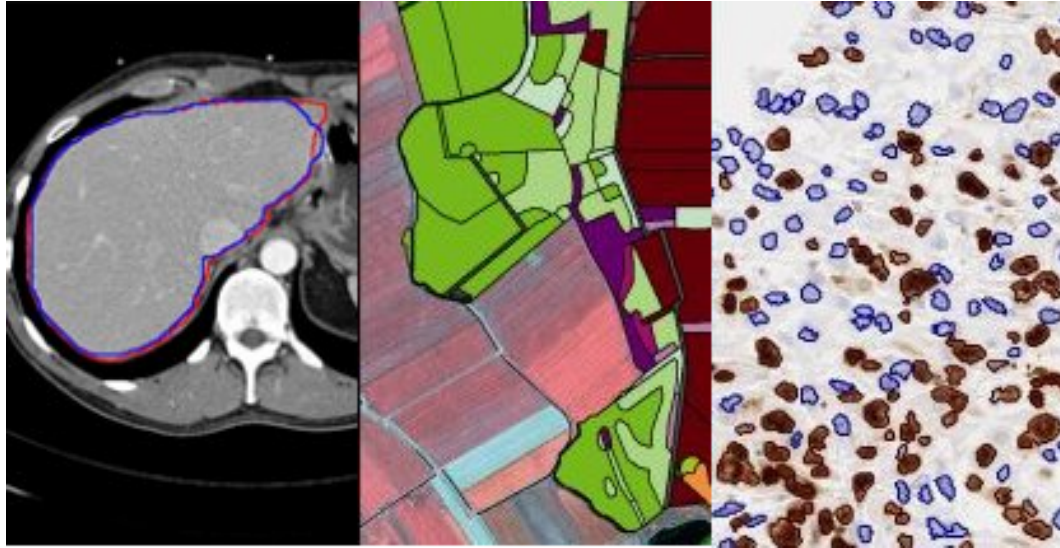


Рис. 1.1: Примеры задач выделения объектов определенных классов на изображениях: (а) – медицинские изображения, выделение печени; (б) – аэрофотоснимки, выделение полей различных культур; (в) – биологические изображения, выделение клеток различных типов

ния поверхностей трёхмерного пространства, идентифицирующих эти объекты. Однако на практике ситуация с лазерными сканами не такая оптимистичная: часто они похожи скорее на карты глубины, чем на аппроксимацию трёхмерной формы объекта. Даже если объект снят с нескольких ракурсов (разработаны методы эффективной регистрации облаков точек [2]), он может быть загорожен другими объектами или сам собой, если он невыпуклый, поэтому скан может содержать неполную поверхность объекта. Данные лазерного сканирования часто бывают зашумлены и разрежены. Они не обладают привычной цветовой информацией. К тому же, обработка облаков точек человеком с помощью существующих технических средств затруднена, так как они ориентированы прежде всего на вывод и ввод двумерных данных.

Тем не менее, анализ трёхмерных данных зачастую полезен. Например, данные с лазерных сенсоров незаменимы для навигации автономных мобильных роботов. Лазерная съёмка применяется также для создания моделей зданий в архитектуре, для обработки данных аэрофотосъёмки в геодезии, для контроля качества продукции, а также для сохранения культурного и исторического наследия. В отличие от фотографий, результат сканирования незначительно зависит от погодных условий. Данные более “объективны”

в том смысле, что охватывается большая часть пространства. Например, для обучения мобильного робота требуются фотографии с определённого ракурса, поскольку камеры, как правило, располагаются на определённой высоте. Размеченные данные для одной модели робота неэффективно использовать для другой. Высококачественная точечная модель в свою очередь задаёт положение объекта в координатной системе, связанной с землёй или даже в абсолютных координатах, поэтому такая модель универсальна для всех моделей роботов [20].

## 1.2 Технология лазерного сканирования

Технология лазерной съёмки аналогична технологии радиолокации, но, в отличие от радаров, лазерные сканеры испускают более короткие волны электромагнитного спектра: инфракрасные, видимые или ультрафиолетовые. Сканер (рис. 1.2(a)) испускает световые импульсы и замеряет время возврата или фазу отражённого от объекта сигнала (иногда также его интенсивность). Таким образом, в результате сканирования получается *облако точек*<sup>1</sup>: неупорядоченный набор точек трёхмерного пространства (рис. 1.2(b)). Имея данные о положении и ориентации сканера, можно перевести облако точек в абсолютные координаты. Если имеются несколько сканов одной сцены из разных точек обзора, они могут быть сшиты в одно облако точек (если сканы представлены в абсолютных координатах, задача становится тривиальной). Таким образом, сканируя сцену с разных позиций, можно получить достаточно полную информацию о её пространственной структуре.

В этой работе нас прежде всего интересуют сканы естественных сцен (в противоположность сканам одного объекта). Можно выделить несколько классов таких сканов:

- сканы, снятые со стационарного сканера (рис. 1.3(a)). Если это скан городской сцены, то в них может потребоваться выделять здания, людей, транспортные средства, столбы, ЛЭП;
- сканы, снятые с движущегося по дороге сканера (рис. 1.3(b)). Классы объектов обычно такие же, как в предыдущем случае;

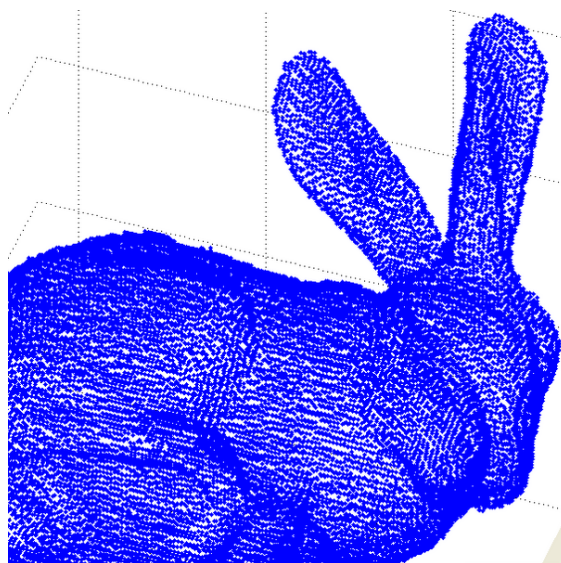
---

<sup>1</sup>В данной работе мы используем термины ‘лазерный скан’ и ‘облако точек’ как синонимы



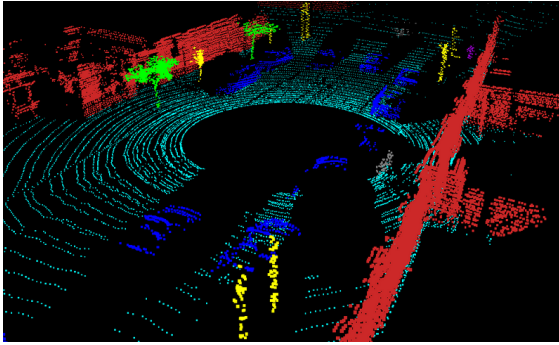


(а) Фотография стационарного лазерного сканера

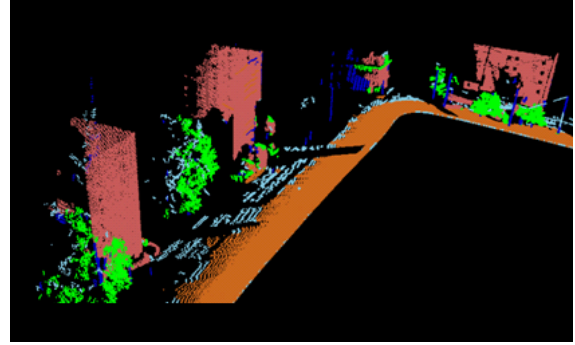


(b) Типичный пример облака точек: Stanford Bunny

Рис. 1.2: Технология лазерного сканирования



(a) Скан, полученный с помощью стационарного сканера. [20] Цвета отражают классы объектов: бирюзовый — уровень земли, синий — транспорт, жёлтый — дорожные знаки, фиолетовый — люди, красный — здания, серый — другое



(b) Скан, полученный с движущегося автомобиля [22]. Цвета отражают классы объектов: малиновый — здания, оранжевый — дорога, зелёный — деревья, синий — столбы, голубой — тротуар

Рис. 1.3: Примеры сканов наружных сцен.

- сканы, снятые с летательного аппарата. Такие сканы характеризуются более низким разрешением и отсутствием информации о вертикальных поверхностях (например, присутствуют только крыши зданий, стены на скане отсутствуют или присутствуют в очень низком разрешении). Поскольку такие сканы применяются для задач картографии и паспортизации, имеет смысл искать здания, деревья (лес), дороги, водоёмы и пр.

# Глава 2

## Постановка задачи

### 2.1 Формальная постановка задачи

Мы поставили задачу реализовать алгоритм машинного обучения классификатора точек скана. Решается задача выделения объектов в скане и их категоризация. Это значит, что конкретный класс описывает все объекты данной категории, но не конкретные их экземпляры. Например, к одному классу относятся все автомобили, к другому - все здания и т.д.

На *вход* алгоритму машинного обучения поступает размеченное облако точек, то есть множество векторов вида  $(x, y, z, c)$ , первые три компоненты которых задают координаты точек в трёхмерном метрическом пространстве  $\mathbb{R}^3$ , а последняя — метку класса для объекта, которому принадлежит точка, которая выбирается из предопределённого неупорядоченного конечного множества классов. На *выходе* алгоритм машинного обучения должен предоставить другой алгоритм — классификатор (который предполагается искать в параметрическом семействе алгоритмов), который на вход получает неразмеченное облако точек, и для каждой его точки находит метку класса. Итак, классификатор реализует функцию со следующей сигнатурой:

$$A : ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)) \mapsto (c_1, c_2, \dots, c_n), c_i \in \{1, \dots, K\}$$

Необходимо заметить, что облака точек должны быть достаточно велики ( $n$  велико) и хотя бы кусочно-непрерывны (при аппроксимации поверхностью), поскольку для каждой точки должна найтись окрестность. Это следует из того, что сама по себе точка в трёхмерном пространстве не несёт никакой информации, она важна лишь в контексте

поверхности, которую она образует.

## 2.2 Критерии качества

Как и в любой задаче машинного обучения, здесь можно определить формальные критерии качества на основе минимизации эмпирического риска. Необходимо собрать достаточно большое количество размеченных облаков точек одного типа. Разделив имеющиеся облака на обучающую и тестовую выборки, можно оценить качество алгоритма обучения. Обучившись на обучающей выборке, мы получаем алгоритм классификации. Подав ему на вход тестовую выборку, мы получим результаты классификации. Поскольку мы знаем ожидаемый результат классификации, мы можем вычислить *точность* — часть правильно классифицированных точек.

На практике глобальная точность не очень хорошо отражает качество классификации. Проблема в том, что данные часто несбалансированны по классам. Например, на естественных сканах к классу проводов обычно относится сравнительно малое число точек, а к классам деревьев и домов — большое. При этом, даже если все провода будут классифицированы неправильно, это мало повлияет на точность. Поэтому разумно рассматривать ошибки для всех классов отдельно. Популярными мерами ошибок являются точность для класса и отклик.

Рассмотрим все точки некоторого класса. Пусть мы проверяем гипотезу  $H_0$ : точка принадлежит данному классу. Тогда все точки можно разделить на четыре части в зависимости от их реального класса и ответа классификатора:

	Классификатор принял $H_0$	Классификатор отверг $H_0$
$H_0$ верна	TP	FN
$H_0$ не верна	FP	TN

Таблица 2.1: Количество точек каждой из четырёх категорий в зависимости от ответа классификатора

*Точность* — часть верно найденных точек из всех точек, отнесённых классификатором к данному классу. *Отклик* — часть верно найденных классификатором точек данного класса. *F-оценка* — среднее гармоническое между точностью и откликом. Она

характеризует общее качество поиска точек данного класса.

$$\text{Точность} = \frac{TP}{TP + FP}, \text{Отклик} = \frac{TP}{TP + FN}, F\text{-оценка} = \frac{2\sqrt{\text{Точность} \cdot \text{Отклик}}}{\text{Точность} + \text{Отклик}}. \quad (2.1)$$

Для того чтобы получить оценку качества классификации одним числом, мы считаем среднее арифметическое F-оценок по всем классам. Это и есть функционал, который мы будем стараться максимизировать.

Чтобы избежать проблемы переобучения, будем использовать технику скользящего контроля (кросс-валидации). В этом случае все доступные размеченные сканы делятся на  $n$  частей примерно равного размера, и производится  $n$  сеансов обучения и тестирования. На каждом из них из выборки выкидывается одна из частей, которая используется для теста, остальные — для обучения. На сегодняшний день техника скользящего контроля — самый надёжный способ тестирования алгоритмов машинного обучения.

## 2.3 Цели работы

Мы поставили перед собой следующие цели:

- составить обзор методов классификации облаков точек;
- собрать выборку размеченных лазерных сканов различных типов для тестирования алгоритмов;
- разработать алгоритм обучения классификатора, показывающий F-оценку, сопоставимую с F-оценкой современных методов на сканах различных типов.

## Глава 3

# Обзор существующих методов

Системы распознавания объектов на основе машинного обучения реализуют стадии обучения и классификации, на каждой из которых производятся следующие действия:

- предобработка
- подсчёт признаков
- обучение/классификация

Рассмотрим отдельно этап предобработки сканов, признаки, которые извлекаются для специфичных данных лазерного сканирования, и какие алгоритмы машинного обучения используются.

### 3.1 Предобработка

На стадии предобработки облака точек унифицируются и подготавливаются к дальнейшему анализу. Несколько сканов сцены, снятых с различных точек, могут быть зарегистрированы друг относительно друга. К нему затем может быть применён некоторый фильтр, сглаживающий шум. Для больших облаков точек также важно строить индекс, который позволяет оптимизировать процесс поиска точки в трёхмерном пространстве. Для этого могут использоваться *k*d-деревья[3], октодеревья[24], R-деревья[13] и их всевозможные модификации. Также возможно выполнение пересегментации или сэмплирования облака для ускорения анализа.

## 3.2 Признаки облаков точек

Выбор признаков является решающим фактором качества классификации. Признаки считаются либо для групп точек (сегментов), либо для отдельных точек. В последнем случае для подсчёта признаков используется некоторая окрестность точки — обычно берутся все точки в фиксированном радиусе, чтобы признаки не зависели от плотности сканирования. В статьях различных авторов используются разные наборы признаков. Как правило, они специфичны для конкретной задачи, но некоторые из них вполне универсальны. Многие авторы используют аппроксимацию окрестности точки плоскостью, что понятно, поскольку скан обычно представляет собой достаточно гладкую поверхность. Далее мы рассмотрим конкретные типы признаков, описанные в литературе.

### 3.2.1 Спин-изображения

#### Определение

В статье [17] вводится определение спин-изображений, которые часто используются для подсчёта локальных признаков, или в качестве самих признаков. Для некоторой точки, для которой задана нормаль и набор соседних точек, можно вычислить такое “изображение”, на самом деле представляющее собой матрицу. Фактически, это вид из точки на другие точки окрестности, усреднённый вращением вокруг нормали и сглаженный билинейным фильтром. Чем больше точек “видно” под конкретным углом, тем “темнее” будет в этом месте изображение. Более строго: для точки  $O$  задаётся система координат  $\alpha\beta$ , где  $\alpha$  — расстояние от точки до нормальной прямой к  $O$ , а  $\beta$  — направленное расстояние от точки до касательной плоскости в  $O$ . Таким образом, точки на окружности с центром на нормальной прямой в  $O$ , лежащей в плоскости, перпендикулярной этой прямой, проектируются в одну точку пространства  $\alpha\beta$ . (рис. 3.1) Спин-изображения часто используются для сопоставления поверхностей, так как разные сканы одной поверхности должны иметь похожие (хотя и не обязательно совпадающие точно) спин-изображения.

#### Применение

В статье [9] авторы предложили использовать спин-изображения для мелких объектов типа каркасных кукол (в противоположность городским сценам) как признаки, инва-

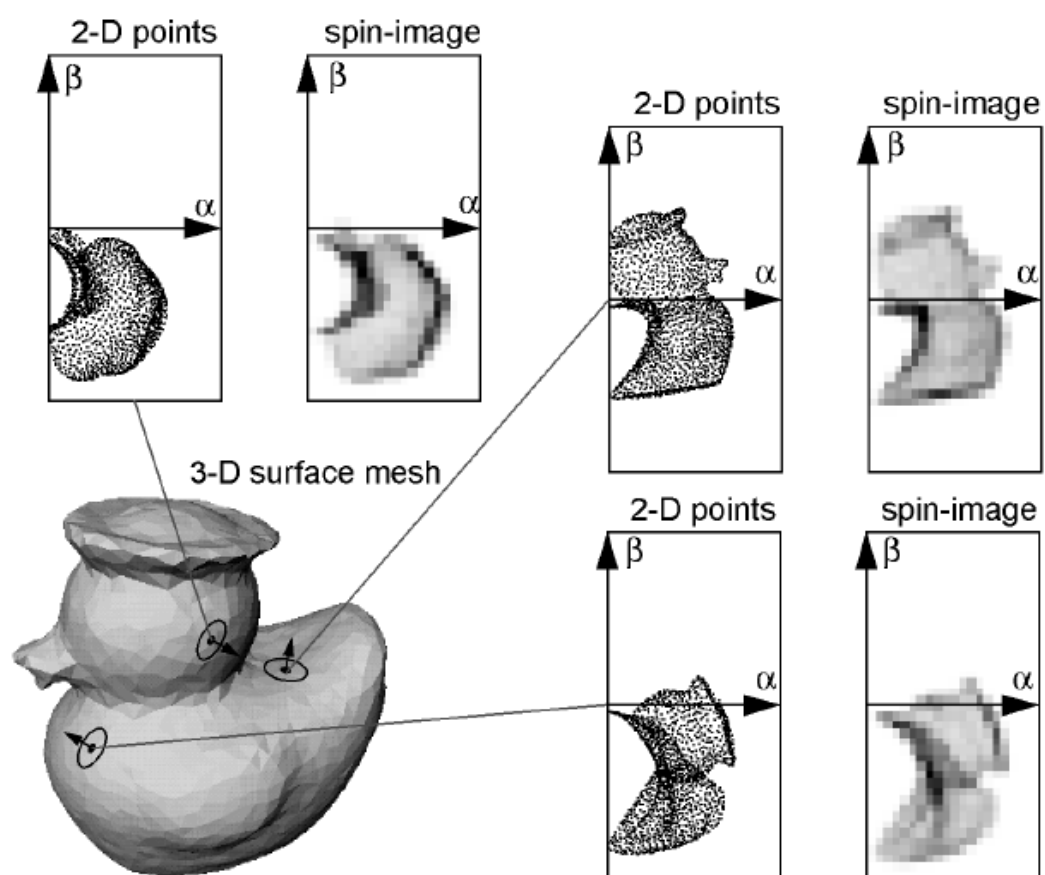


Рис. 3.1: Иллюстрация к определению спин-изображения. [17]



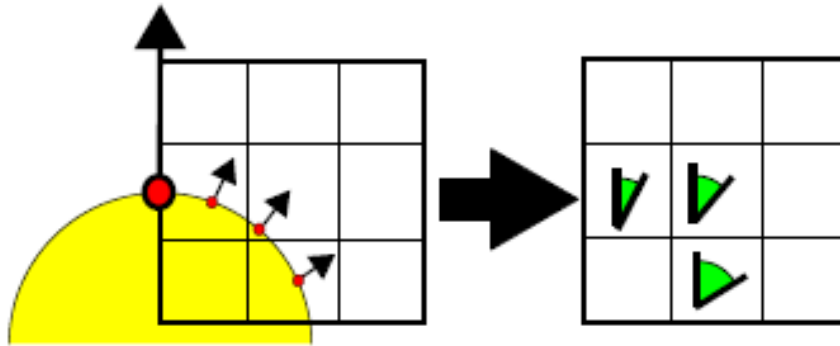


Рис. 3.2: Иллюстрация к варианту спин-изображения от Endres.[29]

риантные к любым поворотам. Предполагается для точек считать спин-изображения, квантуя описанное выше пространство  $\alpha\beta$  на  $5 \times 10$  частей. Считается несколько изображений разного масштаба, производится анализ главных компонент и берутся значения 45 главных компонент. Интересна проблема выбора размера изображений. Согласно выводам авторов статьи, он должен быть сопоставим с размером мелких деталей объектов (например, головы куклы).

В статье [16] продолжаются разработки [9]. Авторы также предлагают считать спин-изображения  $5 \times 10$  в сферах радиуса 10 и 15 сантиметров.

### Модификации

Endres и др. [29] предлагают свой вариант спин-изображений. Для точки вычисляется нормаль с помощью анализа главных компонент по окрестности с фиксированным радиусом. Как и в обычных спин-изображениях, объект вращается вокруг нормали в исходной точке, но ячейки изображения собирают не просто точки, а средние значения углового расстояния от нормалей в точках до нормали к исходной точке (рис. 3.2). Авторы показали, что их вариант даёт результаты лучше, чем классический метод при распознавании абстрактных категорий объектов (таких как “человек”, “ящик”).

### 3.2.2 Признаки матрицы ковариаций

Munoz и др. [22][23][8] используют спектральный подход. Считаются следующие признаки:

- спектральные признаки: производится анализ главных компонент в некоторой локальной окрестности точки (авторы брали все точки в радиусе 60 см). Пусть  $\lambda_0$ ,  $\lambda_1$ ,  $\lambda_2$  – собственные значения матрицы ковариаций в порядке неубывания. Тогда  $\sigma_p = \lambda_0$ ,  $\sigma_s = \lambda_1 - \lambda_0$ ,  $\sigma_l = \lambda_2 - \lambda_1$  – признаки, отвечающие за сходство окрестности с точкой, поверхностью или линией соответственно. Подход напоминает технику выделения границ и углов на изображениях с помощью соответствующих фильтров (таких как Canny[6] и Harris[14]);
- признаки направления: ищутся касательный вектор  $\nu_t$  и вектор нормали  $\nu_n$  в точке как собственные векторы матрицы ковариаций (наименьшая и главная компоненты соответственно). Далее считаются синусы и косинусы углов, образованных этими векторами с вертикальной и горизонтальной плоскостями;
- признаки сходства для пар точек:  $\nu_t$  и  $\nu_n$  нормируются на  $\sigma_l / \max_{i \in \{p,l,s\}} \sigma_i$  и на  $\sigma_s / \max_{i \in \{p,l,s\}} \sigma_i$  соответственно. Сходство двух признаков  $f_1$  и  $f_2$  вводится как  $1/(1 + |f_1 - f_2|)$ .

### 3.2.3 Признаки, специфичные для предметной области

#### Городские сцены

Для городских сцен авторы [9] рекомендуют использовать признаки, инвариантные к повороту в плоскости  $XU$ , а также инвариантные к плотности сканирования:

- Вокруг точки описывается куб со стороной 1 метр (точка — в центре), рассматриваются все точки облака, попавшие в него. Далее выполняется анализ главных компонент, и по двум главным компонентам строится плоскость (назовём её *принципиальной*). Далее куб делится на 27 подкубов, которые ориентированы в соответствии с принципиальной плоскостью, и считается количество точек в каждом подкубе. Таким образом удобно искать “сгустки” точек и плоскости (оценивать планарность окрестности точки).
- Вокруг точки строится вертикальный цилиндр радиуса 25 см. Вычисляются проценты точек в различных сегментах цилиндра (например, лежащие между плоскостями  $z = 2.0$  и  $z = 2.5$  метров).  $z \approx 0$ , как правило, соответствует классу поверхности земли.

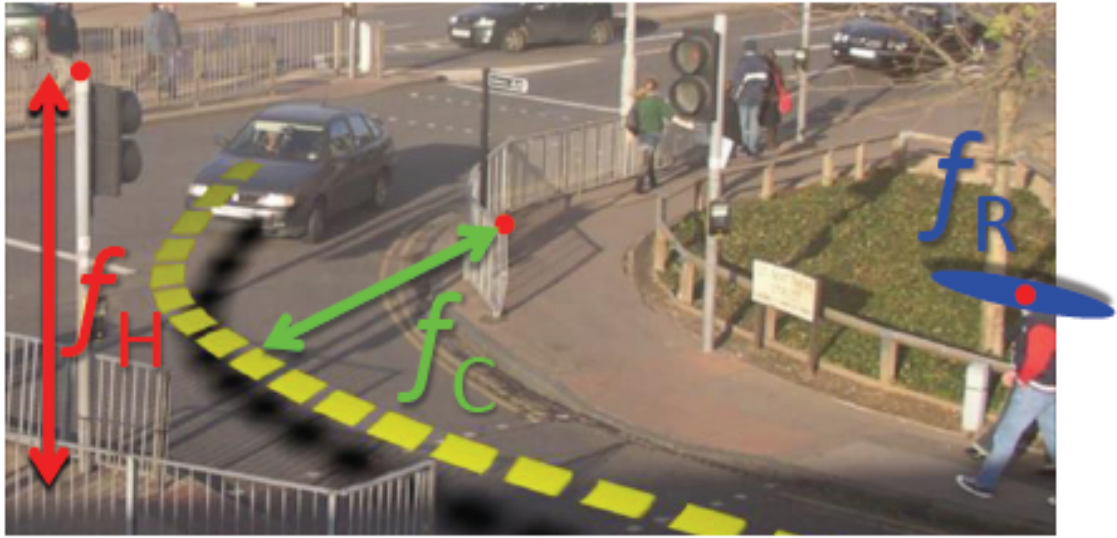


Рис. 3.3: Иллюстрация к статье [25]. Показаны три из пяти используемых типов признаков.

- Бинарный признак “ $z < 2.0$  метров” позволяет отсеивать кустарники.

### Выделение окон

Статья [28] является продолжением разработок [9]. Авторы пытались выделять окна на стене, поэтому они сначала выделяли стену как основную плоскость, а затем уже считали следующие признаки, зная её положение:

- распределение (гистограмма) косинусов углов между нормалью к стене и нормалью к точкам в окрестности;
- отклонение высоты точки от плоскости стены;
- нормализованная высота (нормировка на 15 метров).

### Навигация автотранспорта

[25] используют довольно оригинальный набор признаков, специфичных для навигации автомобилей. Хотя авторы используют для распознавания плотное стерео вместо облаков точек, такие признаки могут быть применены и в нашей задаче. Признаки выбирались так, чтобы они были робастными, интуитивными, эффективно вычислимыми

и универсальными (по отношению к категориям объектов). Авторы также считают этот набор неизбыточным:

- Высота над уровнем земли  $f_H$ : высота — единственная из координат, которая сама по себе несёт некоторый смысл. Из неё вычитают калиброванную высоту расположения камеры, чтобы данные были инвариантны относительно расположения камеры на автомобиле и положения автомобиля (рис. 3.3).
- Длина перпендикуляра к траектории  $f_C$ : хотя автомобиль может проехать по разным траекториям на одной дороге, это признак имеет смысл: деревья и здания обычно удалены от дороги, в отличие от других автомобилей. Проблема в том, что иногда точка не может быть видна (сканирована) в момент её ближайшего расстояния — например, точки на поверхности дороги загорожены самой машиной (рис. 3.3).
- Ориентация поверхности  $f_{O_x}, f_{O_y}$ : выполняется двумерная триангуляция Делоне на плоскости изображения, затем треугольники восстанавливаются обратно, считаются их нормали и берутся их проекции снова на двумерное пространство изображения. Проекция нормали задаётся в нём двумя координатами.
- Плотность траектории  $f_D$ : количество точек, проецируемых в плоскость изображения. Разные объекты могут обладать различной степенью “текстурированности”: деревья и здания более плотные, чем небо и дорога, а автомобили могут быть в разных местах плотными (кузов) и неплотными (стёкла).
- Невязка репроекции  $f_R$ : необходимо отслеживать объект на плоскости изображения и смотреть, как “смещается” облако, его представляющее. Таким образом можно отследить пешеходов и подвижный транспорт (рис. 3.3).

## 3.3 Методы классификации

### 3.3.1 Независимая классификация точек

В самом простом варианте классификации функции назначения меток класса отдельным точкам считаются независимыми одинаково-распределёнными случайными вели-

чинами. Методы восстановления зависимостей такого рода хорошо проработаны в машинном обучении: это алгоритмы обучения классификатора, такие как индукция решающих деревьев [7], бустинг [12], метод опорных векторов (SVM) [26]. Однако на практике обычно учитывают зависимость между соседними точками, поскольку это позволяет повысить робастность метода классификации. Благодаря выбору большой окрестности точки при вычислении локальных признаков происходит сглаживание выхода классификатора, но имеет смысл также использовать методы совместной классификации.

### 3.3.2 Совместная классификация точек

Для моделирования назначения меток классов часто используются Марковские сети. Марковской сетью называется графическая вероятностная модель, в которой связанные случайные величины представляют собой зависимые стохастические процессы. В контексте облаков точек в Марковскую сеть объединяются точки облака, а значением случайной величины в узле сети является метка класса, соответствующая точке [9]. Интерес представляет способ задания потенциалов Марковской сети, с помощью которых формулируется минимизируемая функция энергии.

Более строго, для задачи классификации точек вводится следующая Марковская сеть. Имеем набор дискретных случайных величин  $\mathbf{Y} = \{Y_1, \dots, Y_n\}$ , каждая из которых соответствует одной точке скана и принимает значения из  $Y_i \in \{1, \dots, K\}$ , которые представляют метки классов, назначаемые точкам. Таким образом, вероятность назначения точкам соответствующих меток  $\mathbf{y} = \{y_1, \dots, y_n\}$  задаётся совместным распределением  $\mathbf{P}(\mathbf{Y} = \mathbf{y}) = \mathbf{P}(Y_1 = y_1, \dots, Y_n = y_n)$ . Зависимость между случайными величинами задаётся неориентированным графом, в котором вершины соединяются ребром, если соответствующие точки облака находятся в близости в соответствии с некоторой метрикой (например, евклидово расстояние между точками меньше установленного порога). Задача поиска меток классов, максимизирующих эту вероятность, называется *выводом* в Марковской сети.

Техника факторизации позволяет представить эту вероятность в виде произведения конечного числа функций-потенциалов, заданных на подграфах исходного графа. В общем случае, для осуществления вывода в Марковских сетях нужно учитывать потенциалы всех полных подграфов, которые задают вероятность назначения на них. Несмотря на это, обычно рассматривают только унарные потенциалы (потенциалы вершин)

$\phi_i(Y_i)$  и парные потенциалы (потенциалы рёбер)  $\phi_{ij}(Y_i, Y_j)$  — заданные неотрицательные функции. В этом случае можно осуществлять эффективный вывод оценки назначения, при этом сохраняется достаточная достоверность модели. Совместное распределение случайных переменных в модели записывается следующим образом:

$$\mathbf{P}(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \prod_{i=1}^n \phi_i(y_i) \prod_{(i,j) \in E} \phi_{ij}(y_i, y_j),$$

где  $Z$  — нормировка (статистическая сумма), не зависящая от  $\mathbf{y}$ , а  $E$  — множество рёбер графа.

Необходимо с помощью выбора назначения максимизировать правдоподобие модели. Учитывая свойства монотонности логарифма, в задаче оптимизации можно перейти к минимизации суммы логарифмов, взятой со знаком минус (её часто называют функцией энергии):

$$-\sum_{i=1}^n \log(\phi_i(y_i)) - \sum_{(i,j) \in E} \log(\phi_{ij}(y_i, y_j)) + \log Z \rightarrow \min_{\mathbf{y}}$$

Последний член константный, поэтому его можно исключить из формулировки минимизируемой энергии. В задачах компьютерного зрения чаще всего используется обобщение модели Поттса выбора потенциалов, которая приводит нас к случаю *ассоциативной Марковской сети*:  $\log \phi_{ij}(k, k) = \lambda_{ij}^k \geq 0, \log \phi_{ij}(k, l) = 0, k \neq l$ . В ассоциативных Марковских сетях возможен эффективный вывод с помощью алгоритмов разрезов графа [19].

На практике потенциалы обычно назначаются как функции от признаков. Пусть мы имеем наборы признаков  $\mathbf{x}_i \in \mathbb{R}^{d_n}$  для вершин и  $\mathbf{x}_{ij} \in \mathbb{R}^{d_e}$  для рёбер. Для облаков точек признаки вершин подробно описаны в предыдущей подсекции. Признаки рёбер можно задавать как расстояние между точками, направление ребра в пространстве точек или угол между их приближёнными нормальными. На практике иногда парный потенциал определяют константным, в этом случае признаки вычислять не требуется. Для регрессии признаков к потенциалам часто используется линейная модель:  $\log(\phi_i(k)) = w_n^k \cdot \mathbf{x}_i$ ,  $\log(\phi_{ij}(k, k)) = w_e^k \cdot \mathbf{x}_{ij}$ . Учитывая ограничения, задаваемые ассоциативной Марковской сетью, удобно полагать, что веса и признаки — неотрицательные величины.

В случае классификации на два класса минимум функции энергии точно находится с помощью алгоритмов поиска минимального разреза в графе. Если же классов больше двух, то задача становится NP-трудной, но был предложен приближённый алгоритм

альфа-расширения [4], который представляет собой итерационный процесс, на каждом шаге которого вызывается процедура нахождения минимального разреза графа для двух классов. На практике алгоритм даёт хорошее приближение минимума энергии, однако он имеет ограниченную применимость. В статье [19] показано, что с помощью разрезов на графах может быть минимизирована только функция энергии, потенциалы которой представляют собой *субмодулярную функцию*, то есть

$$-\log \phi(l, l) - \log \phi(k, k) \leq -\log \phi(l, k) - \log \phi(k, l), \quad k, l \in 1, \dots, K \quad (3.1)$$

Другой подход — сформулировать эту задачу как задачу целочисленного программирования. Она решается ослаблением ограничений, то есть сведением к задаче линейного программирования. На практике решение задачи линейного программирования работает дольше, чем поиск минимального разреза на графе [27].

Ещё один метод минимизации такой энергии — алгоритм распространения доверия (belief propagation) [31]. Он даёт точный результат для минимизации энергии на Марковских сетях без циклов, но были разработаны приближённые итеративные методы для циклических сетей на основе передачи сообщений. Один из наиболее эффективных методов приближённой минимизации энергии произвольной формы приведён в [18]. Алгоритм TRW-S (sequential tree-reweighted message passing) использует как элементарную операцию распространение доверия над подграфами данного графа. Вместо решения задачи минимизации энергии, авторы рассматривают двойственную задачу линейного программирования, и формулируют оценку снизу для функции энергии, которая представляет собой вогнутую функцию двойственных переменных. Задача состоит в том, чтобы максимизировать эту оценку. TRW-S находит глобальный минимум в случае субмодулярных парных потенциалов, в противном случае на практике показывает высокую точность, хотя оптимальность не гарантирована.

## Глава 4

# Предлагаемый метод

В этой главе мы подробно рассмотрим наш подход к проблеме. Мы используем традиционные стадии. Сначала мы строим пространственный индекс над обучающим/тестовым сканом, который также служит нам сегментацией. Для сегментов находятся медоиды, и над ними строится граф — основа для Марковской сети. Затем вычисляются унарные и парные потенциалы вершин и рёбер. Для настройки унарных потенциалов используется алгоритм обучения Random Forest[5]. В случае обучения ему на вход подаётся сокращённая выборка из векторов признаков точек обучающей выборки, в случае классификации — признаки медоидов сегментов. Когда унарные и парные потенциалы вычислены, запускается алгоритм вывода в Марковской сети, который возвращает финальное назначение меток классов медоидам сегментов, которые распространяются на сегменты целиком. Теперь мы опишем каждый из этапов подробнее в соответствующей секции.

### 4.1 Пространственный индекс и пересегментация

Пространственный индекс над облаком точек позволяет быстро выполнять пространственные поисковые запросы (такие как “найти  $k$  ближайших соседей” данной точки). Для этой цели мы используем модифицированную структуру R-дерева [1]. R-дерево [13] представляет собой иерархию вложенных параллелепипедов. Листья дерева содержат близкие точки пространства. Дерево полностью сбалансировано, то есть высота поддеревьев на каждом уровне постоянна. Это даёт определённые гарантии по времени



поиска.

Дерево строится в потоковом режиме, то есть точки подаются последовательно. Для каждой точки находится лист, в который её необходимо вставить. Если в листе нет свободных мест, он делится на два новых, которые становятся потомками родителя старого листа. Если при этом родитель переполняется, то он тоже делится на две новых вершины. Таким образом, разделение распространяется вверх, к корню дерева. Наши главные изменения касаются алгоритма выбора листа для вставки точки и алгоритма разделения вершины.

Оригинальный алгоритм предлагает вставлять вершину в лист с кратчайшим минимальным расстоянием до вершины. Поскольку авторы Р-дерева тестировали свой алгоритм на двумерных данных (с оглядкой на приложение в геоинформационных системах), это работало неплохо. В трёхмерном случае поиск такого листа может занимать значительное время, кроме того, это довольно общая ситуация, когда минимальное расстояние от точки до нескольких листьев дерева равно нулю, то есть точка лежит в области пространства, образованной пересечением огибающих параллелепипедов этих листьев, и оригинальный алгоритм оказывается неоднозначным. Мы вводим понятие псевдоцентроида вершины: это средняя точка среди геометрических центров огибающих прямоугольников потомков данной вершины (или точек, если вершина — лист). Наш способ вставки учитывает компромисс между скоростью работы и формой листьев, определяя приоритет листьев для вставки, вычисляя линейную комбинацию расстояния до псевдоцентроида и минимального расстояния до вершины с заранее заданными коэффициентами.

Мы изменили также авторский алгоритм разделения переполняющихся вершин. Гуттман стремился минимизировать суммарный объём огибающих параллелепипедов вершин, получающихся при разделении, и предложил несколько методов, которые минимизируют этот функционал с разной степенью точности и временем работы. Для нас главным критерием является “компактная” форма сегментов: точки, попавшие в один лист должны лежать максимально близко друг к другу. Математически, нужно минимизировать среднее расстояние от точки листа до центра листа (среднее арифметическое его точек). Мы выяснили, что более компактные сегменты получаются, если для разделения вершин используется алгоритм кластеризации на два кластера. В частности, мы использовали алгоритм  $k$ -средних [21].

Таким образом, компактная форма точек в листьях позволяет нам использовать структуру листьев в качестве пересегментации. См. рисунок 5.2(е) для примера результата пересегментации. При классификации, признаки вычисляются только для медоидов сегментов, затем только на медоидах строится Марковская сеть. Результат классификации распространяется с медоидов на соответствующие сегменты. Это несколько огрубляет результаты классификации, но позволяет бороться с шумом сканирования и разметки. При этом объекты, размер которых меньше размера сегмента, не могут найтись даже теоретически. Мы оцениваем максимальную ошибку, вызванную пересегментацией, в секции 5.1. Важно, что пересегментация позволяет ускорить классификацию на порядок, делая возможным эффективную классификацию сканов размером в миллионы точек с помощью одной графической модели.

## 4.2 Построение графа

На сканах как обучающей, так и тестовой выборки мы должны построить граф над медоидами сегментов. В случае обучающей выборки мы собираем статистику для назначения парных потенциалов, в случае тестовой выборки — мы назначаем парные потенциалы и производим вывод в Марковской сети, основанной на построенном графе. Стандартный подход к построению графа - соединять рёбрами все точки с их  $k$  ближайшими соседями, где  $k$  типично принимает значения от 3 до 5 [22]. Это позволяет сохранять примерно постоянную степень вершин в графе даже при меняющейся плотности сканирования. В случае ассоциативных Марковских сетей задаются “усредняющие” потенциалы. Мы же используем более гибкую форму парных потенциалов (подробнее см. в следующей секции), что делает разумным использовать большие значения  $k$ . Для экспериментов мы использовали  $k = 5$ .

Данные аэросъёмки имеют свои особенности. Типичный такой скан представляет собой карту высоты, снятую с отвесно вертикальной позиции. Это ведёт к тому, что на карте отсутствуют вертикальные поверхности, соответствующие, например, заборам или стенам домов. Отсканированная поверхность оказывается в этом случае разрывной, а таким объектам, как крыша дома или крона дерева, соответствует отдельная компонента связности в графе, построенном по описанной выше методике. Поэтому при классификации данных аэросъёмки мы добавляли в граф также рёбра, соответ-

ствующие  $k$  соседним точкам в проекции на горизонтальную плоскость. Это позволило связать рёбрами крыши домов с соседними участками земли. Это важно для нашего метода, поскольку отношения между классами, такие как “крыша лежит выше земли”, обрабатываются специальным образом, чего не делается в модели Поттса.

## 4.3 Признаки и потенциалы

### 4.3.1 Унарные потенциалы

Для назначения унарных потенциалов мы используем выход классификатора Random Forest [5]. Классификатор представляет собой набор из решающих деревьев, каждое из которых обучено на рандомизированном подмножестве признаков. Мы использовали вероятностный выход классификатора, считая части деревьев, проголосовавших за отдельные метки классов. Эти числа используются для назначения унарных потенциалов в Марковской сети. На рисунке 5.2(f) показан типичный выход классификатора для набора точек.

Для обучения классификатора нет необходимости использовать все точки обучающей выборки. Мы прореживаем обучающую выборку, чтобы ускорить обучение, избежать эффекта переобучения и сбалансировать классы. Мы прореживаем выборку так, чтобы в итоге в ней осталось равное число представителей всех классов. Согласно [11], классификатор, обученный на сбалансированной выборке, возвращает несмещённый результат. В противном случае, классификатор игнорирует редкие классы, такие как “провод”, “столбы”.

Мы использовали следующие признаки для настройки Random Forest, подробнее описанные в секции 3.2:

- спектральные признаки и признаки направления, как описано в [22];
- спин-изображения [17] с пониженной размерностью: точки аккумулировались не по клеткам сетки, а отдельно по строкам и столбцам, эксперименты показали, что такая редукция не приводит к потере различающей способности;
- угловые спин-изображения [29], у которых была сокращена размерность таким же образом.

### 4.3.2 Парные потенциалы

TBD.

## 4.4 Вывод в Марковской сети

Для вывода максимальной апостериорной оценки назначения в Марковской сети часто используются алгоритмы на основе разрезов в графах. В нашем случае они неприменимы, так как потенциалы не удовлетворяют субмодулярным ограничениям. Поэтому мы попытались применить алгоритмы циклического распространения доверия (loopy belief propagation, [31]) и передачи сообщений с помощью перевзвешивания деревьев (TRW; tree-reweighted message passing, [30]). Последний показал лучшую производительность и эффективность.

Мы использовали модификацию TRW-S (последовательная TRW), разработанную Владимиром Колмогоровым [18]. Алгоритм находит глобальный минимум вогнутой функции нижней грани для функции энергии. Для этого на графе задаётся порядок вершин, затем его рёбра покрываются монотонными цепями-подграфами (каждое ребро должно оказаться покрытым хотя бы одной цепью). Для эффективности вывода имеет смысл максимизировать длину таких цепей. Поскольку каждая цепь не содержит циклов, на ней возможен вывод с помощью передачи сообщений. Алгоритм выполняет итеративно две стадии: репараметризации и усреднения. На первой передаются сообщения с учётом порядка вершин и структуры монотонных цепей, на втором результат усредняется по всем деревьям. Колмогоров также доказал сходимость алгоритма и дал критерий останова, однако на практике удобно просто ограничивать число итераций или следить за поведением неубывающей функции — нижней грани энергии.

## Глава 5

# Экспериментальные результаты

Для проведения экспериментов была подготовлена программная реализация на языке C++, включающая в себя:

- реализацию модифицированного Р-дерева, утилит для работы с ним и поиска ближайших соседей<sup>1</sup>;
- реализацию подсистемы подсчёта признаков, позволяющую легко конфигурировать набор признаков для классификации;
- подсистему для подсчёта унарных потенциалов, обеспечивающую интеграцию с реализацией Random Forest из библиотеки OpenCV<sup>2</sup>;
- подсистему для подсчёта потенциалов Марковской сети и интеграцию с реализациями алгоритма вывода на основе разрезов графов<sup>3</sup> и TRW-S<sup>4</sup>;
- подсистему для настройки весов при признаках для назначения парных потенциалов с помощью структурного метода опорных векторов, интеграция с библиотекой SVM<sup>struct</sup> <sup>5</sup>.

---

<sup>1</sup>Код оформлен в виде библиотеки с открытым исходным GML LidarK, <http://graphics.cs.msu.ru/en/science/research/3dpoint/lidark>

<sup>2</sup><http://opencv.willowgarage.com/wiki/>

<sup>3</sup><http://www.csd.uwo.ca/~olga/code.html>

<sup>4</sup><http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/TRW-S.html>

<sup>5</sup>[http://www.cs.cornell.edu/People/tj/svm\\_light/svm\\_struct.html](http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html)

Мы оцениваем производительность нашего метода на двух наборах данных, полученных с помощью сканирующей системы ALTM 2050 (Optech Inc.), которые мы обозначим как наборы данных А и В. Мы выделяем следующие классы объектов: “земля”, “здание”, “дерево”, “куст”; на первом скане мы также выделяем класс “автомобиль”. Данные распределены между классами на первом скане в следующей пропорции: 43.8% земли, 2.0% зданий, 0.3% автомобилей, 53.1% деревьев и 0.8% кустов. На втором скане распределение похожее, но отсутствуют автомобили. Мы разделили каждый из двух сканов на обучающую и тестовую выборки размеров 1.1 и 1.0 миллионов точек для набора А и 1.5 и 1.2 миллионов точек для набора В соответственно. Для обучения классификатора, назначающего унарные потенциалы, используются признаки не всех точек обучающего скана. Признаки считаются по исходному скану, но при формировании обучающей выборки векторы признаков прореживаются в соответствии с популярностью классов так, чтобы в выборке оказалось примерно равное число объектов каждого класса. Классификатор, обученный на такой выборке, выдаёт несмещённый результат. Таким образом, обучающая выборка для набора А состоит из 9 тысяч векторов признаков.

## 5.1 Ошибка пересегментации

Очевидно, использование пересегментации делает результаты классификации грубее. В этой секции мы описываем постановку эксперимента для оценки ошибки, которая может быть вызвана пересегментацией. Мы вычисляем минимальную ошибку, которую можно получить, назначая всем точкам каждого сегмента общую метку класса. Для обоих наборов данных результаты представлены на рисунке 5.1. Меньшая ошибка на наборе В объясняется отсутствием в нём объектов класса автомобиль. Автомобили не могут быть сегментированы качественно, так как они обычно маленького размера (число точек, соответствующих автомобилю, совпадает по порядку с размером сегмента), также они часто сливаются с землёй. Уменьшение количества точек в сегменте уменьшает также нижнюю грань ошибки, но увеличивает число сегментов, что ведёт к повышению вычислительной сложности. Принимая во внимание общий результат классификации, мы остановились на размере сегмента в 64 точки, что соответствует теоретическим ошибкам 2.2% и 1.6% для наборов А и В соответственно, что соответствовало 30 тысячам сегментов в каждом из сканов.

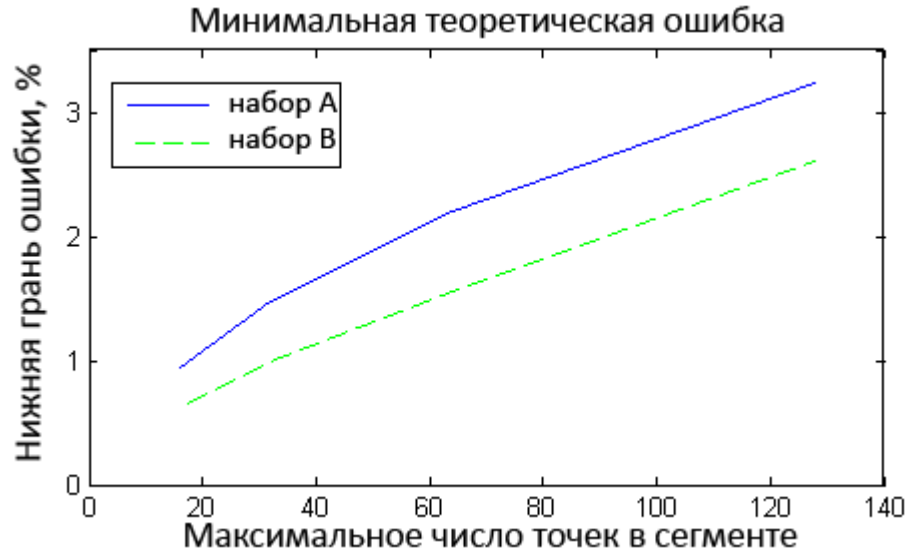


Рис. 5.1: Минимальная теоретическая ошибка, вызванная пересегментацией, линейна по отношению к максимальному размеру сегмента

## 5.2 Точность классификации

Мы поставили несколько экспериментов, чтобы сравнить наш метод с существующими подходами: выходом Random Forest без Марковской сети и с результатом Марковской сети с постоянными потенциалами ( $\phi_{ij}(Y_i, Y_j) = [Y_i = Y_j]$ ), в которой рёбрами соединены 3 ближайших соседа каждой точки.

Ассоциативная Марковская сеть с постоянными потенциалами сглаживает результаты, удаляя как шум, так и мелкие классы. (5.2(c)). На первом скане такое сглаживание помогло значительно улучшить результат, в то время как на втором улучшения не было. Неассоциативная Марковская сеть содержит больше рёбер, потенциалы которых зависят от расстояния между точками и других признаков. Это ведёт к более разборчивому сглаживанию (5.2(d)). Такие потенциалы позволяют выражать отношения типа “крыша находится выше земли”, что приводит к более качественной сегментации.

Как видно из рисунка, после применения неассоциативной Марковской сети всё равно остаются ошибки. Но это ошибки другого рода. Ассоциативная Марковская сеть часто классифицирует неправильно целые мелкие объекты, в то время как наш алгоритм ошибается в частях одного объекта. Такие ошибки могут быть исправлены на этапе

постобработки с помощью фильтрации, либо с помощью более аккуратной настройки потенциалов.

### 5.3 Эффективность

Поскольку мы используем пересегментацию, классификация работает относительно быстро. Временные характеристики собраны в 5.1. Тесты запускались на машине с процессором Intel Pentium с рабочей частотой 2.80 ГГц и 1 Гб оперативной памяти. Заметим, что мы строили Марковскую сеть над целым сканом, содержащим миллион точек, в отличие от, скажем, авторов [23], которые явно делили сканы на более мелкие части. Таким образом, для классификации всего набора нам достаточно одной графической модели. Мы строим Р-дерево над классифицируемым сканом и используем структуру его листьев в качестве пересегментации. На каждом из наборов данных получалось около 30 тысяч сегментов. Затем мы строим новое дерево над медоидами полученных сегментов. Такая структура обеспечивает масштабируемость алгоритма построения графа связей.

	набор А	набор В
Построение индекса	278	315
Вычисление признаков	100	146
Построение графа	351	498
Классификация Random Forest	8	8
Вывод в Марковской сети	86	60
<b>Итого</b>	<b>823</b>	<b>1027</b>

Таблица 5.1: Время вычислений, с



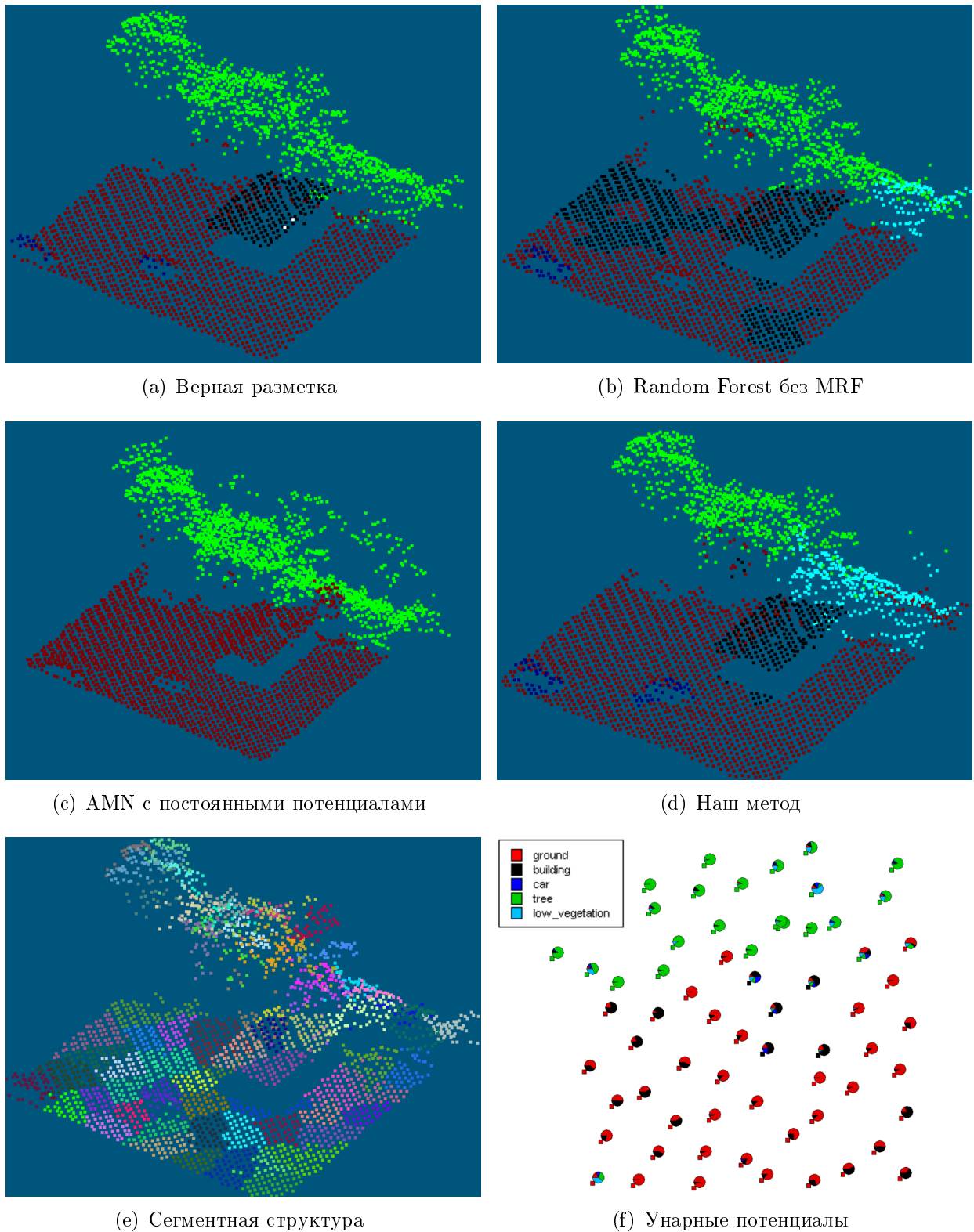


Рис. 5.2: (а) - (d) Результаты, полученные с помощью различных алгоритмов. Цветами обозначены: красный – земля, чёрный – здание, синий – автомобиль, зелёный – дерево, бирюзовый – куст. (е) Выход алгоритма сегментации. (f) Выход алгоритма Random Forest. Для каждого сегмента квадрат отражает верную метку класса, круговая гистограмма – выход мультиклассового классификатора. *Лучше рассматривать в цвете и увеличенным!*

## Глава 6

# Заключение

В рамках дипломной работы были решены следующие задачи:

- написан обзор методов классификации лазерных сканов;
- разработан новый подход к классификации лазерных сканов на основе неассоциативных Марковских сетей, проведено его сравнение с существующими методами;
- на основе разработанного алгоритма реализована система классификации лазерных сканов с произвольными классами;
- разработана система эффективного выделения земли на данных, снятых с транспортного средства.

Планируется продолжить разработки в этом направлении. Одно из возможных усовершенствований алгоритма заключается в использовании Марковских сетей с кликами высокого порядка, как это делается в [8].

# Литература

- [1] Автоматическая сегментация облаков точек на основе элементов поверхности / A. Velizhev, R. Shapovalov, D. Potapov et al. // GraphiCon. — Moscow: 2009. [http://www.graphicon.ru/proceedings/2009/conference/se10/114/114\\_Paper.pdf](http://www.graphicon.ru/proceedings/2009/conference/se10/114/114_Paper.pdf).
- [2] Велижеев, . Разработка и исследование алгоритмов взаимного ориентирования трёхмерных дискретных моделей объектов, полученных в результате лазерного сканирования: Диссертация на соискание учёной степени кандидата наук / Московский государственный университет геодезии и картографии. — 2008.
- [3] Bentley, J. L. Multidimensional binary search trees used for associative searching / J. L. Bentley // *Communications of the ACM*. — 1975. — Vol. 18, no. 9. <http://portal.acm.org/citation.cfm?id=361007>.
- [4] Boykov, Y. Fast approximate energy minimization via graph cuts / Y. Boykov, O. Veksler, R. Zabih // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 2001. — Vol. 23, no. 11. — P. 1222 – 1239. <http://doi.ieeecomputersociety.org/10.1109/10.1109/34.969114>.
- [5] Breiman, L. Random forests / L. Breiman // *Machine Learning*. — 2001. — Vol. 45, no. 1. — P. 5–32. <http://www.springerlink.com/index/U0P06167N6173512.pdf>.
- [6] Canny, J. A computational approach to edge detection / J. Canny // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 1986. — Vol. 8, no. 6. <http://portal.acm.org/citation.cfm?id=11275>.
- [7] Classification and regression trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. — Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

- [8] Contextual classification with functional Max-Margin Markov Networks / D. Munoz, J. Bagnell, N. Vandapel, M. Hebert // IEEE Conference on Computer Vision and Pattern Recognition. — IEEE, 2009. — Июнь. — Pp. 975–982. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206590>.
- [9] Discriminative Learning of Markov Random Fields for Segmentation of 3D / D. Anguelov, B. Taskar, V. Chatalbashev et al. // IEEE Conference on Computer Vision and Pattern Recognition. — 2005. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.3211&rep=rep1&type=pdf>.
- [10] Efficient road mapping via interactive image segmentation / O. Barinova, R. Shapovalov, S. Sudakov et al. // 3D City Models, Road Databases and Traffic Monitoring (CMRT). Vol XXXVIII / Ed. by U. Stilla, F. Rottensteiner, N. Paparoditis. — Paris: IAPRS, 2009. — Pp. 1–8. [http://www.isprs.org/proceedings/XXXVIII/3-W4/Pub/CMRT09\\_Barinova\\_et\\_al.pdf](http://www.isprs.org/proceedings/XXXVIII/3-W4/Pub/CMRT09_Barinova_et_al.pdf).
- [11] *Elkan, C.* The foundations of cost-sensitive learning / C. Elkan // International Joint Conference on Artificial Intelligence. — Vol. 17. — Citeseer, 2001. — P. 973–978. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.514&rep=rep1&type=pdf>.
- [12] *Freund, Y.* A Short Introduction to Boosting / Y. Freund, R. Schapire // International Joint Conference on Artificial Intelligence. — 1999. — Pp. 1401–1406.
- [13] *Guttman, A.* R-trees: A dynamic index structure for spatial searching / A. Guttman // ACM SIGMOD International Conference on Management of Data. — ACM New York, NY, USA, 1984. — P. 47–57. <http://portal.acm.org/citation.cfm?id=602259.602266>.
- [14] *Harris, C.* A Combined Corner and Edge Detection / C. Harris, M. Stephens // Alvey Vision Conference. — 1988. — Pp. 147 – 151. [www.bmva.org/bmvc/1988/avc-88-023.pdf](http://www.bmva.org/bmvc/1988/avc-88-023.pdf).
- [15] *Horn, B.* Extended Gaussian images / B. Horn // *Proceedings of the IEEE*. — 1984. — Vol. 72, no. 12. — Pp. 1671–1686. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1457341>.

- [16] Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data / R. Triebel, R. Shmidt, O. Martinez, W. Burgard // International Joint Conference on Artificial Intelligence. — 2007. — Pp. 2225–2230.
- [17] *Johnson, A.* Using spin images for efficient object recognition in cluttered 3 d scenes / A. Johnson, M. Hebert // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 1999. — Vol. 21, no. 5. — P. 433–449. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.8816&rep=rep1&type=pdf>.
- [18] *Kolmogorov, V.* Convergent tree-reweighted message passing for energy minimization / V. Kolmogorov // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 2006. — Vol. 28, no. 10. — P. 1568. <http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/TRW-S-PAMI.pdf>.
- [19] *Kolmogorov, V.* What energy functions can be minimized via graph cuts? / V. Kolmogorov, R. Zabih // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 2004. — Февраль. — Vol. 26, no. 2. — Pp. 147–59. <http://www.ncbi.nlm.nih.gov/pubmed/15376891>.
- [20] *Lai, K.* 3D Laser Scan Classification Using Web Data and Domain Adaptation / K. Lai, D. Fox // Robotics: Science and Systems. — 2009. <http://www.roboticsproceedings.org/rss05/p22.pdf>.
- [21] *Macqueen, J.* Some methods for classification and analysis of multivariate observations / J. Macqueen // Berkeley Symposium on Mathematical Statistics and Probability. — Vol. 233. — University of California Press, 1967. — Pp. 281–297.
- [22] *Munoz, D.* Directional associative markov network for 3-d point cloud classification / D. Munoz, N. Vandapel, M. Hebert // International Symposium on 3D Data Processing, Visualization and Transmission. — Atlanta, GA: 2008. <http://www.cc.gatech.edu/conferences/3DPVT08/Program/Papers/paper200.pdf>.
- [23] *Munoz, D.* Onboard contextual classification of 3-D point clouds with learned high-order Markov Random Fields / D. Munoz, N. Vandapel, M. Hebert // *IEEE International Conference on Robotics and Automation*. — 2009. — Май. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152856>.

- [24] *Reddy, D.* Representation of Three-Dimensional Objects. — 1978.
- [25] Segmentation and recognition using structure from motion point clouds / G. Brostow, J. Shotton, J. Fauqueur, R. Cipolla // European Conference on Computer Vision. — Marseille, France: Springer, 2008. — P. 44.
- [26] Support Vector Regression Machines / H. Drucker, C. J. C. Burges, L. Kaufman et al. // NIPS. — Vol. 9. — MIT Press, 1996. — Pp. 155–161.
- [27] *Taskar, B.* Learning associative Markov networks / B. Taskar, V. Chatalbashev, D. Koller // International Conference on Machine Learning. — ACM New York, NY, USA, 2004. <http://portal.acm.org/citation.cfm?id=1015444>.
- [28] *Triebel, R.* Robust 3d scan point classification using associative markov networks / R. Triebel, K. Kersting, W. Burgard // IEEE International Conference on Robotics and Automation. — 2006.
- [29] Unsupervised Discovery of Object Classes from Range Data using Latent Dirichlet Allocation / F. Endres, C. Plagemann, C. Stachniss, W. Burgard // Robotics: Science and Systems. — Seattle, USA: 2009.
- [30] *Wainwright, M. J.* MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches / M. J. Wainwright, T. S. Jaakkola, A. S. Willsky // *IEEE Transactions on Information Theory*. — 2005. — Vol. 51(11). — P. 3697 – 3717.
- [31] *Yedidia, J.* Generalized belief propagation / J. Yedidia, W. Freeman, Y. Weiss // NIPS. — 2001. <http://eprints.kfupm.edu.sa/42528>.